

# **Metodi di Monte Carlo Applicati Alla Computer Grafica**

Tesi in Calcolo Numerico

**Presentata da:  
Tanzi Alessio**

**Relatore:  
Prof.ssa Marina Popolizio**

Anno Accademico 2022/2023



**LIBERATORIA ALLA CONSULTAZIONE DELLA TESI DI LAUREA DI CUI  
ALL'ART.4 DEL REGOLAMENTO DI ATENEO PER LA CONSULTAZIONE DELLE  
TESI DI LAUREA (D.R. n. 479 del 14/11/2016).**

Il sottoscritto TANZI ALESSIO matricola 581991

Corso di Laurea

INGEGNERIA INFORMATICA E DELL'AUTOMAZIONE

autore della presente tesi di Laurea dal titolo

METODI DI MONTE CARLO APPLICATI ALLA COMPUTER GRAFICA

Parola chiave METODI DI MONTE CARLO

Abstract La sintesi delle immagini computerizzate fotorealistiche presenta complicazioni. In particolare, la sua formalizzazione culmina in una equazione integrale ricorsiva ad alta dimensionalita', dunque risolvibile solamente con metodi numerici stocastici quali l'integrazione secondo i Metodi di Monte Carlo, approfonditi per il Light Transport.

Autorizza

Non autorizza

la consultazione della presente tesi, fatto divieto a chiunque di riprodurre in tutto o in parte quanto in essa contenuto.

Bari,

Firma Alessio Tanzì

*La/il sottoscritta/o,*

Cognome: TANZI

Nome: ALESSIO

Matricola: 581991

Data Seduta Laurea: 26/07/2023.

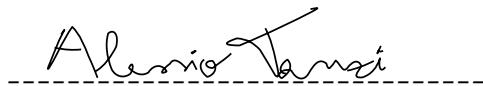
*dichiara*

*che la tesi cartacea consegnata in data odierna è conforme a quella caricata su esse3, quindi, approvata  
dal Relatore Prof. POPOLIZIO MARINA*

(in stampatello)

Bari, 14/07/2023

*Firma*

  
Alessio Tanzì

---

## Abstract

Nel seguente documento si presentano tutti i fondamenti matematici e scientifici al problema della sintesi di un'immagine basata su fondamenti fisici. A tal scopo, l'algoritmo proposto per la renderizzazione di scene tridimensionali è il *Path Tracing*, il quale risolve il problema della *Global Illumination*, cioè colorazione delle superfici contando illuminazione diretta ed indiretta.

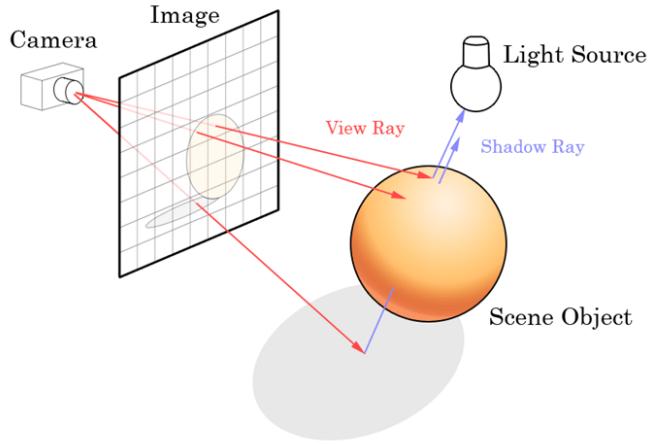


Figure 1: Illustrazione dell'algoritmo Path Tracing. L'obiettivo di questo documento è spiegare come i Metodi di Monte Carlo rendono possibile la stima della funzione immagine  $r_f(x, y)$

Immediatamente si presenta un problema per il calcolo di tale immagine (mostrata in Figura 1): al fine di poter costituire il risultato finale, l'*Image Plane*<sup>1</sup> è suddiviso a formare una griglia uniforme. In seguito, supponendo  $r(x, y)$  funzione immagine, una data cella di area  $A_{px}$  avrà colorazione pari all'integrale

$$r_f(x, y) = \int_{A_{px}} r(x', y') dA \quad (1)$$

Nelle applicazioni pratiche tale funzione non solo non è analiticamente integrabile, ma non è nota analiticamente. Infatti,  $r(x, y)$  può soltanto essere campionata, generando dei raggi con origine  $\vec{o}$  pari alla posizione dell'osservatore, e direzione  $\hat{\omega}$  pari a  $\frac{\vec{p}_i - \vec{o}}{\|\vec{p}_i - \vec{o}\|}$ . Una volta campionato tale segnale bidimensionale, un filtro di ricostruzione  $f(x, y)$  è utilizzato, mediante l'operazione di convoluzione  $*$ , per ottenere un'approssimazione della funzione immagine, filtrata,  $r_f(x, y)$ .

$$r_f(x, y) = \int_{A_{px}} f(x - x', y - y') r(x', y') dA = (f * r)(x, y) \quad (2)$$

Emerge quindi il problema di tale integrale. Tale espressione si presta alla risoluzione mediante i Metodi di Monte Carlo, del quale si riporta qui lo stimatore più semplice

$$r_f(x, y) \approx \frac{\|A_{px}\|}{n} \sum_{i=1}^n f(x - x_i, y - y_i) r(x_i, y_i) \quad (3)$$

<sup>1</sup>Successivamente chiamato anche *Film Plane*, seguendo la nomenclatura [PJH23]

---

Si sposta l'attenzione allora al calcolo di ciascun campione  $r(x_i, y_i)$ : il suddetto raggio è propagato per la scena, finchè non interseca una superficie. Dunque, il compito adesso è quello di calcolare il colore che tale superficie emette. In quanto il concetto di colore è strettamente legato alla percezione umana, tale computazione vede come grandezza utilizzata la *Radianza Spettrale*, potenza uscente o incidente in un punto di una superficie, da/a una data direzione. Tale grandezza è opportunamente campionata in punti equispaziati dello spettro visibile e da luogo ad un integrale più complicato, che verrà descritto nel seguito (Capitolo 6).

Nelle pagine a seguire si descrivono i Metodi di Monte Carlo, dimostrando la loro grande potenzialità nella risoluzione del problema presentato, fino a formulare la strategia completa dell'algoritmo di Path Tracing.

## Acknowledgements

Vorrei ringraziare la Prof.ssa Marina Popolizio che ha acceso in me la passione per il calcolo numerico e permesso di dedicarmici spostando anche l'attenzione sulla grafica computerizzata, oggetto della mia curiosità negli ultimi tempi. Inoltre, ringrazio la mia famiglia per avermi fornito coraggio e supporto per tutto il mio percorso.

Spendo le ultime parole di questa introduzione per sottolineare quanto affascinante sia la Computer Grafica, quanto complicata. Di seguito, è indicato il link per il codice relativo alla implementazione dei concetti qui spiegati:  
<https://github.com/alexoz12v2/monteCarlo>

# Contents

<b>I Introduzione ai Metodi di Monte Carlo</b>	<b>1</b>
<b>1 Integrazione di Monte Carlo</b>	<b>2</b>
1.1 Preliminari . . . . .	2
1.2 Monte Carlo Integration . . . . .	3
1.3 Esempio: Risoluzione di Integrali Semplici . . . . .	5
1.4 Schemi di Campionamento . . . . .	6
1.4.1 Inverse Transform Sampling . . . . .	7
1.4.2 Acceptance-Rejection Method . . . . .	9
1.4.3 Metropolis-Hastings Sampling . . . . .	10
1.5 Metodi di Riduzione della Varianza . . . . .	12
1.5.1 Importance Sampling . . . . .	13
<b>2 Campionamento e Ricostruzione con Quasirandom Number Generation</b>	<b>14</b>
2.1 Campionamento . . . . .	15
2.1.1 Richiami Fourier Analysis . . . . .	15
2.1.2 Quasirandom Number Generation . . . . .	17
2.1.3 Halton Sampler . . . . .	17
2.2 Ricostruzione . . . . .	19
2.2.1 Box Filter . . . . .	20
2.2.2 Gaussian Filter . . . . .	21
<b>3 Light Transport Modello e Surface Reflection</b>	<b>25</b>
3.1 Path Integral della Rendering Equation . . . . .	26
3.2 Path Tracing . . . . .	28
3.3 Applicazione del Metodo di Monte Carlo: Esempio Path Tracing Semplificato . .	29
<b>II Fondamenti Fisici</b>	<b>39</b>
<b>4 Richiami sull'elettromagnetismo e Radiometria</b>	<b>40</b>
4.1 Radiometria . . . . .	40
4.1.1 Radianza incidente ed uscente . . . . .	43
4.2 Radiazione di Corpo Nero . . . . .	43
<b>5 Fotometria, Colore e Camera</b>	<b>45</b>
5.1 Fotometria . . . . .	45
5.1.1 fotometria e sistemi di rendering . . . . .	46

5.2	Colorimetria . . . . .	46
5.2.1	sRGB color space . . . . .	49
5.2.2	Conversione da XYZ o RGB a SPD . . . . .	50
5.3	Camera . . . . .	52
5.3.1	pinhole camera model . . . . .	52
5.3.2	filtrare image samples . . . . .	55
<b>6</b>	<b>Modelli di Fonti Luminose e Superfici</b>	<b>57</b>
6.1	Modellazione di Superfici . . . . .	57
6.1.1	Proprietà ottiche delle interfacce . . . . .	60
6.1.2	BRDF . . . . .	63
6.1.3	Modelli di BRDF . . . . .	66
6.1.4	Modelli di superficie secondo Microfacet Theory . . . . .	69
6.2	Sorgenti Luminose . . . . .	78
6.2.1	Modelli di sorgenti luminose . . . . .	79
6.2.2	Light Sampling . . . . .	83
<b>III</b>	<b>Appendici</b>	<b>85</b>
<b>A</b>	<b>Numeri Pseudocasuali e Algoritmi Vari</b>	<b>86</b>
A.1	Generazione di numeri Pseudorandom . . . . .	86
<b>B</b>	<b>Codice</b>	<b>88</b>
B.1	Owen Scrambling di punti della Sequenza di Halton . . . . .	88
B.2	Acceptance-Rejection Sampling di un disco unitario . . . . .	89
B.3	Metropolis-Hastings Sampling per campionare $\mathcal{N}(0, 1)$ . . . . .	90
B.4	Monte Carlo Integration per un integrale semplice . . . . .	90
B.5	Semplice Path Tracer . . . . .	91
B.6	Calcolo della varianza di un'immagine . . . . .	93
	<b>Bibliography</b>	<b>95</b>
	<b>Glossary</b>	<b>96</b>

## **Part I**

# **Introduzione ai Metodi di Monte Carlo**

# Chapter 1

## Integrazione di Monte Carlo

Il Rendering è fondato su equazioni integrali, le quali, non possiedono una forma chiusa generale, per metodi numerici come la regola del trapezio possiedono convergenza lenta per integrali multidimensionali e discontinui, oltre a richiedere un gran numero di sample points, crescente esponenzialmente con la dimensionalità del problema (*Curse of Dimensionality*).

Dunque, per risolvere tali problemi, si opta ad un approccio non deterministico, Monte Carlo Integration, il quale ci permette di stimare il valore di un integrale  $\int f(x)dx$  arbitrario con il solo prerequisito di poter calcolare la funzione integranda in determinati punti.

Si noti che gli algoritmi di Monte Carlo, in quanto basano la scelta dei punti del dominio in cui valutare la funzione integranda casualmente, *forniscono diversi risultati* in ogni esecuzione, dato lo stesso input, ma forniscono un risultato che *statisticamente, in media, è vicino alla soluzione corretta*<sup>1</sup>.

### 1.1 Preliminari

Introduciamo il concetto statistico di *Stimatore*

**Def. 1.1.1.** — Si supponga  $S$  sia un parametro (di un campione o di una distribuzione) da calcolare, detto *Stimando*. Uno *Stimatore*  $\tilde{S}_n$  è una funzione di una collezione di variabili aleatorie  $X_i$  che mappa lo spazio campionario a una *Stima*. Tale Stimatore è dunque una variabile casuale del tipo

$$\tilde{S}_n = f(X_1, X_2, \dots, X_n) \quad (1.1)$$

La convergenza di uno Stimatore è qualificata a seconda di quanto "forte e stringente" tale convergenza è. In particolare, data una sequenza di variabili aleatorie  $X_n$ , ci sono tre modi in cui essa può convergere ad una variabile aleatoria  $X$

Convergenza in Distribuzione: le CDF di  $X_n$ ,  $F_{X_n}$  convergono puntualmente alla CDF di  $X$ ,  $F_X$  in tutti i punti in cui essa è definita e continua.

$$X_n \xrightarrow{d} X \text{ iff } \lim_{n \rightarrow \infty} F_{X_n}(x) = F_X(x) \quad (1.2)$$

---

<sup>1</sup>Per essere più precisi, si dovrebbe affermare che il risultato fornito converge in probabilità alla soluzione

Si noti che è la tipologia di convergenza più debole in quanto non restringe in alcun modo le osservazioni estratte da ciascuna delle variabili aleatorie

Convergenza in Probabilità: La probabilità di vicinanza arbitraria tra le osservazioni di  $X_n$  e  $X$  cresce man mano che si prosegue nella sequenza

$$X_n \xrightarrow{P} X \text{ iff } \lim_{n \rightarrow \infty} \Pr(|X_n - X| \geq \varepsilon) = 0 \quad \forall \varepsilon > 0 \quad (1.3)$$

Si noti che ciò non assicura la correttezza di alcuna osservazione della sequenza, soltanto che è molto improbabile discostarsi di molto dal valore esatto per  $n$  sufficientemente grande

Convergenza Quasi Certa: Il sottoinsieme  $\bar{\Omega}$  nel quale le osservazioni della sequenza si discostano dal valore vero è finito. Ciò vuol dire che, al limite, la convergenza è assicurata, con esattezza quasi certa, per via dell'esistenza di finiti valori errati nelle osservazioni della sequenza

$$X_n \xrightarrow{a.s.} X \text{ iff } \Pr\left(\left\{\omega \in \Omega : \lim_{n \rightarrow \infty} X_n(\omega) = X(\omega)\right\}\right) = 1 \quad (1.4)$$

Maggiore è il "grado di convergenza", e maggiore è la frequenza in cui lo stimatore ci restituisce un risultato quasi esatto. Dunque si dice *Debolmente Consistente* uno Stimatore che converge in probabilità allo stimando, e *Fortemente Consistente* uno Stimatore che converge quasi certamente allo stimando.

Tale parametro può essere stimato con lo stimatore *Media Campionaria*, il quale è definito come

$$\tilde{E}_n[X] = \frac{1}{n} \sum_{i=1}^n x_i \quad (1.5)$$

dove  $\{x_1, \dots, x_n\}$  campione estratto casualmente da un insieme di  $n$  variabili aleatorie indipendenti e identicamente distribuite (i.i.d.)  $X_i$ . Tale stima può essere incrementalmente raffinata come

$$\tilde{E}_i[X] = \frac{(i-1)\tilde{E}_{i-1}[X] + x_i}{i} \quad (1.6)$$

Si dimostra, rispettivamente, tramite la WLLN (Legge dei grandi numeri debole) che lo stimatore media campionaria è debolmente consistente, e tramite la SLLN (Legge dei grandi numeri forte) che lo stimatore *media campionaria* è *fortemente consistente*. Ciò, assieme al fatto che tale stimatore altro non fa che stimare un integrale, rende la media campionaria un ottimo candidato per formulare lo stimatore di Monte Carlo.

## 1.2 Monte Carlo Integration

Si ricordi che la aspettazione di una variabile aleatoria  $X$ , la cui PDF  $p_X$  definita in un dominio  $D$  è pari a  $E[X] = \int_D x p_X(x) dx$ , per la quale vale la seguente proprietà: sia  $y = f(x)$ , allora

$$E[f(X)] = \int_{\Omega_y} y p_Y(y) dy = \int_{\Omega_x} f(x) p_y(f(x)) |\det(J_f(x))| dx = \int_{\Omega_x} f(x) p_X(x) dx$$



Figure 1.1: Illustrazione della varianza in una immagine renderizzata. Immagine da [PJH23]

È possibile ricavare uno stimatore per l'integrale di una funzione integranda arbitraria  $f(x)$

$$E\left[\frac{f(X)}{p_X(X)}\right] = \int_{\mathcal{D}} \frac{f(x)}{p_X(x)} p_X(x) dx = \int_{\mathcal{D}} f(x) dx \quad (1.7)$$

Assumendo densità di probabilità uniforme nel dominio  $\mathcal{D}$ , si ottiene

$$\|\mathcal{D}\| E[f(X)] = \|\mathcal{D}\| \int_{\mathcal{D}} f(x) p_X(x) dx = \int_{\mathcal{D}} f(x) dx \quad (1.8)$$

Sostituendo l'aspettazione con la media campionaria, otteniamo uno stimatore *unbiased*<sup>2</sup> per l'integrale

$$\tilde{F}_n = \|\mathcal{D}\| \tilde{E}_n[f(X)] = \frac{\|\mathcal{D}\|}{n} \sum_{i=1}^n f(X_i) \quad (1.9)$$

Rilasseremo in seguito, all'introduzione di Importance Sampling, l'assunzione di campionamento con PDF uniforme, giungendo a

$$\tilde{F}_n = \|\mathcal{D}\| \tilde{E}_n[f(X)] = \frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{p(X_i)} \quad (1.10)$$

Una misura di accuratezza di tale stimatore è la sua varianza, la quale si traduce, nell'immagine renderizzata, in variazioni brusche tra pixel adiacenti, come mostrato in Figura 1.1. Tale varianza<sup>3</sup>, per lo stimatore in Equazione 1.9

$$\begin{aligned} V[\tilde{F}_n] &= V[\|\mathcal{D}\| \tilde{E}_n[f(X)]] = \|\mathcal{D}\|^2 V[\tilde{E}_n[f(X)]] = \frac{\|\mathcal{D}\|^2}{n} V[f(X)] \\ &= \frac{\|\mathcal{D}\|^2}{n} \frac{1}{\|\mathcal{D}\|} \int_{\mathcal{D}} (f(x) - \bar{f})^2 dx = \frac{\|\mathcal{D}\|}{n} \int_{\mathcal{D}} f(x)^2 dx - \bar{f}^2 \end{aligned} \quad (1.11)$$

Da cui la deviazione standard, la quale rappresenta l'errore dello stimatore di monte carlo, è pari a

$$\sigma[\tilde{F}_n] = \sqrt{V[\tilde{F}_n]} = \sqrt{\frac{\|\mathcal{D}\|^2}{n} V[f(X)]} = \frac{\|\mathcal{D}\|}{n^{\frac{1}{2}}} \sigma[f(X)] \quad (1.12)$$

<sup>2</sup>vedi sezioni seguenti

Essa diminuisce come  $\mathcal{O}(n^{-\frac{1}{2}})$ , il che dimostra che l'ordine di convergenza dello stimatore è  $1/2$ . Esso è chiamato "diminishing return" dal fatto che per abbattere l'errore atteso di un fattore  $1/n$  bisogna campionare  $n^2$  volte la funzione in esame. Si noti che tecniche di quadratura convergono più velocemente per integrali 1D, ma appena si passa a integrali multidimensionali il loro costo e convergenza peggiorano considerevolmente, mentre la accuratezza di Monte Carlo non dipende dalla dimensionalità, rendendolo l'unica alternativa. Altre caratteristiche dello stimatore sono *efficienza, bias, mse*

Bias: Differenza tra aspettazione dello stimatore e lo stimando

$$\beta = E[\tilde{F}_n(X)] - \int_{\mathcal{D}} f(x)dx \quad (1.13)$$

Nonostante sembrino sconvenienti, in quanto non tendono al valore desiderato, possono comunque risultare vantaggiosi nel caso si desideri uno stimatore con varianza minore. Per esempio, lo stimatore biased per la aspettazione

$$\frac{1}{2} \max\{X_1, X_2, \dots, X_n\} \quad (1.14)$$

ha deviazione standard che segue  $\mathcal{O}(n^{-1})$

Efficienza: Parametro che permette di stimare la bontà di uno stimatore in relazione alla varianza ottenuta  $V[X]$  e al suo running time  $T[X]$ , pari a

$$\epsilon[\tilde{F}_n] = \frac{1}{V[\tilde{F}_n]T[\tilde{F}_n]} \quad (1.15)$$

MSE: Errore quadratico medio, strettamente collegato alla varianza, in quanto è definito come l'aspettazione del quadrato della differenza tra uno stimatore ed il valore vero

$$MSE[\tilde{F}_n] = E\left[\left(\tilde{F}_n - \int_{\mathcal{D}} f(x)dx\right)^2\right] = V[\tilde{F}_n] + \beta^2 \quad (1.16)$$

Tale metrica risulta utile quando si è a conoscenza di una previa stima, magari con un numero elevato di campioni, di un integrale  $F$ , per la quale possiamo applicare lo stimatore media campionaria per calcolarne il suo valore approssimato.

### 1.3 Esempio: Risoluzione di Integrali Semplici

Al fine di rendere chiara l'applicazione della Equazione 1.9 al calcolo degli integrali, applichiamo il metodo di Monte Carlo all'integrale definito in  $[0, 10]$  della seguente funzione:

$$f(x) = \frac{\sin\left(\sqrt{\frac{\pi^2}{10}}x\right)}{\sqrt{\frac{\pi^2}{10}}x} e^{\sqrt{\frac{\pi^2}{10}}x} \quad (1.17)$$

---

<sup>3</sup>Ricordiamo che  $V[aX + b] = a^2V[X]$  e che, se le variabili  $\{X_i\}_{i=1}^n$  sono incorrelate,  $V[\tilde{E}_n[f(X)]] = V[X]/n$

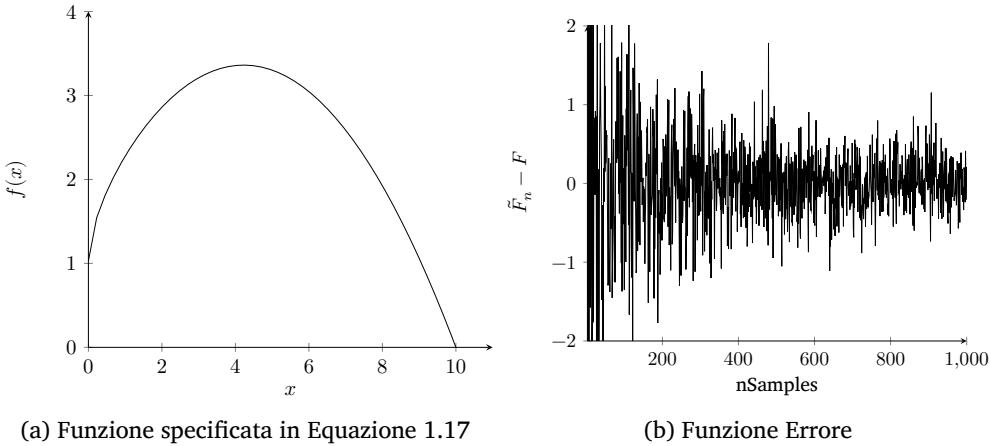


Figure 1.2: Grafico della funzione nell’Esempio di Sezione 1.3, con errore della stima ottenuta da Equazione 1.9

per la quale è possibile giungere alla soluzione analitica

$$\hat{f}(x) = \frac{10 \left( \sin\left(\pi\sqrt{\frac{x}{10}}\right) - \cos\left(\pi\sqrt{\frac{x}{10}}\right) \right) e^{\pi\sqrt{\frac{x}{10}}}}{\pi^2} + C \quad (1.18)$$

Per cui il valore di tale integrale è noto e pari a

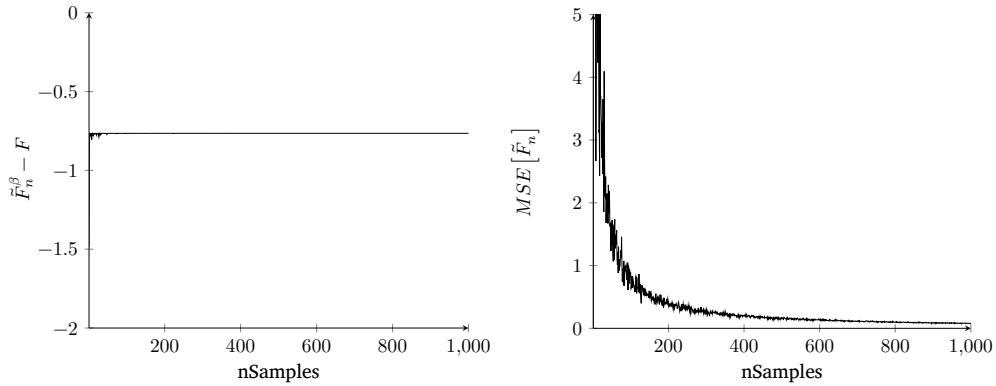
$$F = \int_0^{10} f(x) dx = \hat{f}(10) - \hat{f}(0) = \frac{10e^\pi + 10}{\pi^2} \approx 24.45963551499059 \quad (1.19)$$

Al fine di poter stimare numericamente tale integrale, è stata applicata Equazione 1.9, dove i campioni  $x_i$  sono stati estratti con distribuzione uniforme nel dominio  $[0, 10]$ , la cui generazione è stata implementata mediante la libreria numpy. In particolare, dei generatori supportati, è stato utilizzato un Generatore Congruenziale Permutato<sup>4</sup> I risultati ottenuti dalla simulazione di Monte Carlo con codice mostrato in Appendice B.4 sono mostrati in Figura 1.2b e Figura 1.3

## 1.4 Schemi di Campionamento

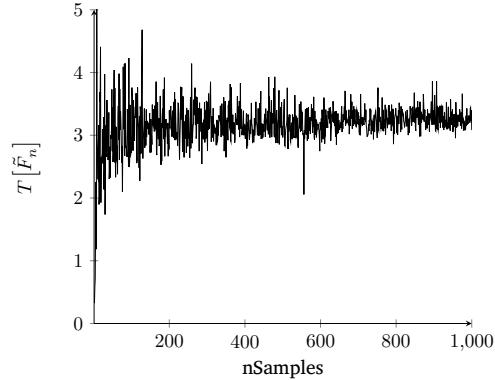
Supponendo si abbia a disposizione un algoritmo per estrarre un numero casuale uniformemente distribuito nell’intervallo  $[0, 1]$  (Appendice A.1) oppure un algoritmo che permetta di estrarre numeri quasirandom da una sequenza con low discrepancy (Sezione 2.1.2), analizziamo metodi per poter campionare samples da una distribuzione arbitraria, come quelle analizzate in Capitolo 6.

Le tre tecniche che citiamo a tal scopo sono *Inverse Transform Sampling*, *Acceptance-Rejection Sampling*, *Metropolis Hastings Sampling*



(a) Funzione errore derivata dall'uso dello stimatore biased  $\frac{1}{2} \max\{f(X_1), \dots\}$ . Si osserva come la varianza sia visibilmente diminuita, a costo della convergenza al valore corretto.

(b)  $MSE$  dello stimatore unbiased



(c) Funzione efficienza  $T[\tilde{F}_n]$  dello stimatore unbiased. Si noti come esso presenta varie fluttuazioni, derivanti dalle condizioni continuamente variabili del calcolatore.

Figure 1.3: Ulteriori risultati dell'Esempio presentato in Sezione 1.3

#### 1.4.1 Inverse Transform Sampling

Date due variabili casuali  $X$  e  $Y$ , di CDFs<sup>5</sup>  $P_x$  e  $P_y$ , già calcolate analiticamente, e la possibilità di campionare  $X$  ed ottenere osservazioni  $x$ , possiamo trasformare tali campioni in campioni  $y$  secondo la seguente procedura:

- Calcolare la *funzione quantile* della variabile aleatoria  $Y$ ,  $P_y^{-1}$
- La variabile aleatoria  $Y$  è dunque posta a  $P_y^{-1}(P_x(x))$

<sup>4</sup>Brevemente introdotti in Appendice A, per più informazioni si rimanda a [https://en.wikipedia.org/wiki/Permuted\\_congruential\\_generator](https://en.wikipedia.org/wiki/Permuted_congruential_generator)

<sup>5</sup>Questo metodo funziona anche nel discreto, sostituendo PDF con PMF e integrali con somme

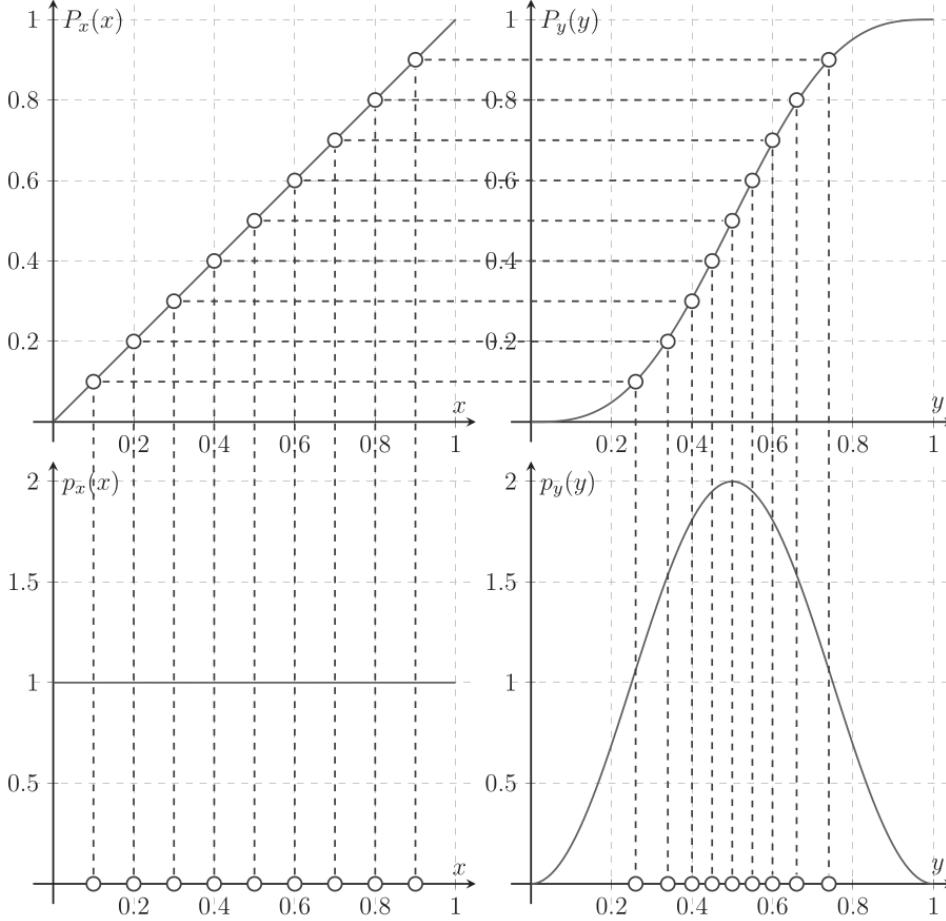


Figure 1.4: Illustrazione della trasformazione di samples tra due PDFs (riga inferiore) usando le loro CDFs (riga superiore). Immagine da [Peg16]

Dove la funzione quantile altro non è che una funzione di distribuzione inversa generalizzata, la quale rende possibile invertire CDFs con intervalli costanti

$$P_y^{-1}(p) = \inf \{y \in \mathbb{R}^d : P_y(y) \geq p\}, \quad p \in [0, 1] \quad (1.20)$$

Il procedimento è illustrato in Figura 1.4.

Nel caso di una distribuzione multivariata, l' inverse transform sampling è un processo iterativo nel quale, scelta una prima dimensione in cui campionare, si calcola la sua CDF marginale per poi invertirla e trasformare la prima dimensione del campione iniziale. In seguito, tutte le altre dimensioni sono calcolate mantenendo fisse le dimensioni già campionate mediante il calcolo di CDFs condizionali della dimensione corrente, dato il valore fissato di tutte le dimensioni già campionate, e la loro inversione permette di portare a termine la computazione.

Ad esempio, data una PDF multivariata bidimensionale  $p(x, y)$ , una coordinata casuale  $x_i$  può essere campionata applicando la procedura di inverse transform sampling alla sua PDF marginale  $p_x(x) = \int_{\Omega_y} p(x, y) dy$ , mentre una coordinata casuale  $y_i$  può essere campionata

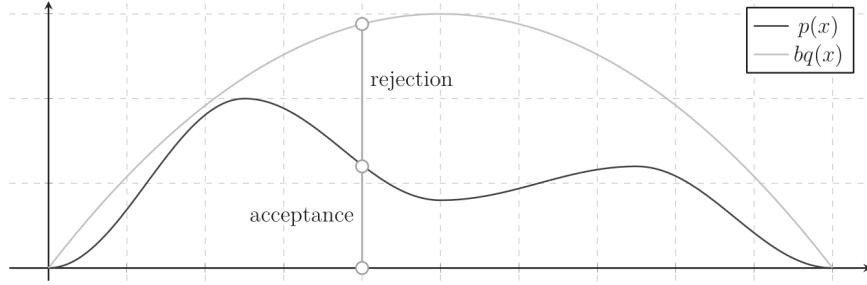


Figure 1.5: Illustrazione del rejection sampling. Immagine da [Peg16]

applicando la procedura di inverse transform sampling alla PDF condizionale  $p_{y|x}(y|x_i) = p(x_i, y)/p_x(x_i)$ .

Nel caso in cui la PDF congiunta risulta separabile, cioè le variabili sono indipendenti  $\int \int p(x, y) dx dy = \int p_x(x) dx \int p_y(y) dy$  allora siamo in grado di applicare inverse transform sampling a ciascuna coordinata utilizzando le rispettive PDFs marginali.

#### 1.4.2 Acceptance-Rejection Method

Alcune funzioni, una volta normalizzate<sup>6</sup>, possiedono CDFs non invertibili o non calcolabili. Il metodo *Acceptance-Rejection Sampling* permette di campionare samples secondo una data PDF  $p(x)$  supponendo di avere la possibilità di campionare, mediante altri metodi, una PDF  $q(x)$ , tale che esista un upper bound  $b > p(x)/q(x)$ ,  $\forall x \in \Omega$  cioè

$$0 \leq \frac{p(x)}{bq(x)} \leq 1 \quad (1.21)$$

L'algoritmo consiste nel

- Estrarre un campione  $x_i$  dalla variabile aleatoria  $X$ , distribuito secondo la PDF  $q(x)$
- Generare un numero casuale  $\xi \sim \mathcal{U}(0, 1)$
- Se  $\xi b q(x_i) \leq p(x_i)$ , allora il campione  $x_i$  è accettato come realizzazione della PDF target  $p(x)$ , ed è rifiutato in caso contrario

Dato un campione  $x$ , la *acceptance probability* è data da

$$\Pr\left(\xi \leq \frac{p(x)}{bq(x)}\right) = \int_0^{\frac{p(x)}{bq(x)}} dx' = \frac{p(x)}{bq(x)} \quad (1.22)$$

La cui aspettazione rappresenta la proporzione di campioni accettati

$$E\left[\frac{p(X)}{bq(X)}\right] = \int_{\Omega} \frac{p(x)}{bq(x)} q(x) dx = \frac{1}{b} \int_{\Omega} p(x) dx = \frac{1}{b} \quad (1.23)$$

in media, dunque,  $b$  campioni sono estratti in media prima di accettarne uno. Ciò nonostante, Rejection-Sampling può non essere efficace nel campionamento di distribuzioni altamente concentrate in una piccola regione del loro dominio. Inoltre, man mano che la dimensionalità del

<sup>6</sup>Come spiegato in seguito, una tecnica che può ridurre la varianza nei metodi di Monte Carlo, *Importance Sampling*, consiste nel campionare punti in cui valutare la funzione integranda con probabilità proporzionale al valore assunto dalla funzione stessa, dunque spiegata la necessità di ricavare PDFs da funzioni arbitrarie

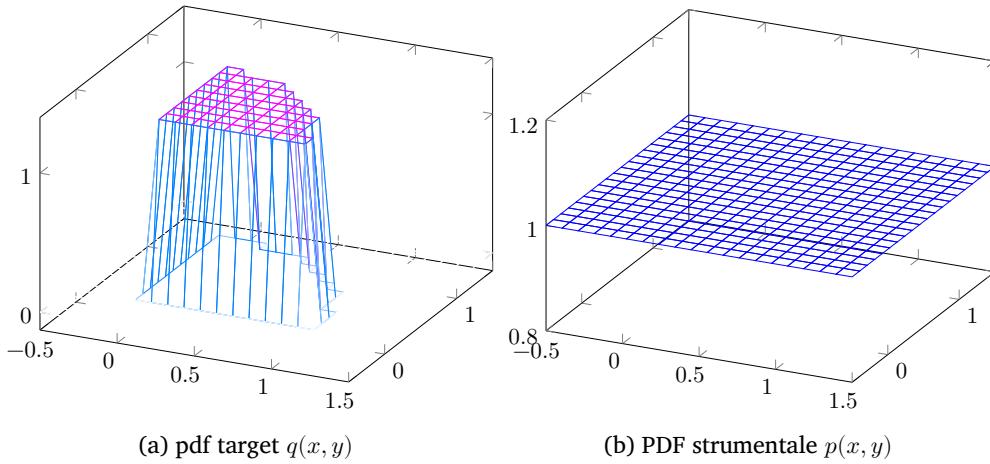


Figure 1.6: Illustrazione delle due PDF coinvolte nell'esempio proposto per Rejection rampling

problema aumenta, il volume "in eccesso" della distribuzione strumentale aumenta, rendendo il metodo soggetto alla curse of dimensionality.

Per *Curse of Dimensionality* si riferisce all'insieme di fenomeni che penalizzano o rallentano determinati algoritmi numerici, i quali diventano più evidenti man mano che la dimensionalità aumenta. Nel caso del campionamento, tale problema deriva dall'aumento esponenziale del volume all'aggiunta di una dimensione extra in uno spazio campionario.

Come esempio di applicazione, si consideri il campionamento con distribuzione uniforme di un dominio pari ad un quarto di disco unitario, utilizzando come proposal distribution una distribuzione uniforme definita in un quadrato unitario<sup>7</sup>.

La PDF target è dunque

$$q(x,y) = \begin{cases} \frac{4}{\pi} & \text{se } x^2 + y^2 \leq 1 \text{ e } x, y \geq 0 \\ 0 & \text{altrimenti} \end{cases} \quad (1.24)$$

con proposed distribution

$$p(x, y) = \begin{cases} 1 & \text{se } (x, y) \in [0, 1]^2 \\ 0 & \text{altrimenti} \end{cases} \quad (1.25)$$

da cui si evince che  $b \geq 4/\pi$ , il che fa in modo che un campione sia accettato dopo aver generato  $\approx 1.27$  samples. Il codice Python che implementa tale procedura è in Appendice B.2

### 1.4.3 Metropolis-Hastings Sampling

Piuttosto che generare campioni in maniera indipendente, costruiamo una *Markov Chain* costituente un Random Walk nello spazio campionario, la cui distribuzione, al limite, converge alla distribuzione desiderata, detta *distribuzione stazionaria* o distribuzione all'equilibrio. I vantaggi rispetto agli altri metodi qui presentati sono

<sup>7</sup>È interessante notare come, riformulando tale procedura come mostrato in `alt_rejection_sampling`, si possa stimare il valore di  $\pi$

- Ogni iterazione produce un campione
- Non richiede la normalizzazione della funzione target  $f(x)$ . Vedi Equazione 1.31
- Non richiede una CDF calcolabile ed invertibile

Il suo difetto è quello di generare campioni correlati tra loro, e ciò rende inefficaci tecniche di riduzione della varianza come Stratified Sampling.

In quanto si tratta di una catena di Markov, la probabilità condizionale di ottenere un dato campione  $x_i$  dipende solamente dal campione precedente (*Markov Property*)

$$p(x_i|x_0, \dots, x_{i-1}) = p(x_i|x_{i-1}) \quad (1.26)$$

Inoltre tale catena si dice *reversibile* in quanto gode della proprietà (*detailed balance property*)

$$p(x_{i-1})p(x_i|x_{i-1}) = p(x_i)p(x_{i-1}|x_i) \quad (1.27)$$

Si parte da un campione iniziale, ottenuto arbitrariamente,  $x_0$  detto *seed*, per poi pescare samples successivi applicando perturbazioni tramite una funzione di transizione<sup>8</sup>  $p_t(x_{i-1} \rightarrow x_i)$ , per cui ogni campione è pescato con densità di probabilità

$$p_i(x_i) = \int_{\Omega} p_{i-1}(x_{i-1})p_t(x_{i-1} \rightarrow x_i)dx_{i-1} \quad (1.28)$$

Si può dimostrare che la condizione per la convergenza è data dalla proprietà di reversibilità per la PDF  $p_{\infty}(x)$ , e che una volta soddisfatta, la densità di probabilità dei campioni  $x_i, x_{i+1}, \dots$  rimane fissa a  $p_{\infty}(x)$ .

Affinchè tale densità di transizione ci conduca all'equilibrio alla PDF target  $p_{\infty}(x) = p(x) = f(x)/ \int f(x)dx$ , l'approccio utilizzato è quello di dividere la densità di distribuzione di transizione in

- *Proposal Distribution Density*  $p_p(x_i|x_{i-1})$ , normalizzata, PDF condizionale di proporre lo stato  $x_i$  dato  $x_{i-1}$
- *Acceptance Probability*  $a(x_{i-1} \rightarrow x_i)$ , probabilità di accettare il nuovo stato  $x_i$  considerato lo stato di partenza  $x_{i-1}$ . Se il nuovo stato è rifiutato, allora si pone il campione  $i$ -esimo pari al campione  $(i-1)$ -esimo

definite in modo tale che

$$p_t(x_{i-1} \rightarrow x_i) \geq p_p(x_i|x_{i-1})a(x_{i-1} \rightarrow x_i) \quad (1.29)$$

Riformulando la condizione di detailed balance, in caso di ugualanza, in

$$\frac{a(x_{i-1} \rightarrow x_i)}{a(x_i \rightarrow x_{i-1})} = \frac{p_{\infty}(x_i)p_p(x_{i-1}|x_i)}{p_{\infty}(x_{i-1})p_p(x_i|x_{i-1})} = \frac{p_{\infty}(x_i)p_p(x_{i-1}|x_i)}{p_{\infty}(x_{i-1})p_p(x_i|x_{i-1})} \quad (1.30)$$

Dove la proposal distribution è scelta arbitrariamente, mentre il rateo a cui la distribuzione converge a  $p(x)$  può essere massimizzato scegliendo acceptance probability [Peg16]

$$a(x_{i-1} \rightarrow x_i) = \min \left\{ \frac{p(x_i)p_p(x_{i-1}|x_i)}{p(x_{i-1})p_p(x_i|x_{i-1})}, 1 \right\} = \min \left\{ \frac{f(x_i)p_p(x_{i-1}|x_i)}{f(x_{i-1})p_p(x_i|x_{i-1})}, 1 \right\} \quad (1.31)$$

Inoltre, nel caso in cui la proposal density sia simmetrica  $p_p(x_{i-1}|x_i) = p_p(x_i|x_{i-1})$ , allora si riduce in

$$a(x_{i-1} \rightarrow x_i) = \min \left\{ \frac{f(x_i)}{f(x_{i-1})}, 1 \right\} \quad (1.32)$$

---

<sup>8</sup> $p_t(x_{i-1} \rightarrow x_i) = p_t(x_i|x_{i-1})$

cioè il cammino casuale non dipende dalla direzione ma solo dalla distanza, e nella pratica (incluso il codice in appendice) questo è l'approccio seguito.

Si può delineare dunque la procedura di Metropolis-Hastings Sampling

- Data target PDF  $f(x)/\int f(x)dx$  e sample iniziale  $x_0$ , ripeti per  $n$  volte
  - campiona  $x^*$  con densità  $p_p(x|x_{i-1})$  e  $\xi$  con densità  $\mathcal{U}(0, 1)$
  - calcola la probabilità di accettazione  $\alpha = a(x_{i-1} \rightarrow x_i)$
  - se  $\xi < \alpha$ , allora  $x_i = x^*$ , altrimenti  $x_i = x^*$

Esempio: Applico l'algoritmo per campionare una distribuzione gaussiana standard. Scelgo come  $p_p$  una gaussiana centrata in  $x_i$  con deviazione standard 0.05. In quanto tale PDF simmetrica,  $p_p(x_i|x^*) = p_p(x^*|x_i)$ , tale PDF si semplifica nel calcolo della probabilità di accettazione. Il codice Python dedicato è trovato in Appendice B.3

Dato che la PDF di ogni campione  $x_i$  approssima la densità di distribuzione stazionaria solamente al limite per  $x \rightarrow \infty$ , i primi campioni generati non sono sufficientemente accurati. L'algoritmo soffre del cosiddetto *Startup Bias*, il quale può essere mitigato scartando dei campioni iniziali, pur non sapendo a priori quanti scartare. Tale strategia è nominata *Burn-In Phase*.

Per assicurare una buona copertura dello spazio campionario, la Proposal Distribution Density

- Deve frequentemente restituire piccole mutazioni, ma
- Deve avere probabilità non nulla di produrre grosse mutazioni<sup>9</sup>

Un approccio simmetrico che ci permette di eseguire tale Random Walk nel dominio è una proposal distribution density uniforme attorno al punto  $x_{i-1}$ , la quale ci fornisce la seguente regola di mutazione

$$x_i^{*(k)} = (x_{i-1}^{(k)} \pm s\xi) \bmod 1 \in [0, 1] \quad (1.33)$$

dove  $\cdot^{(k)}$  indica la coordinata  $k$ -esima del punto,  $s$  seed definito a priori.

Si noti che tale approccio non è universale, perché per assicurare una copertura ottimale la Proposal Distribution Density deve approssimare la Target Distribution Density, magari utilizzando una PDF opportunamente scelta e campionata mediante Inverse Transform Sampling.

## 1.5 Metodi di Riduzione della Varianza

Dato uno stimatore di Monte Carlo Unbiased, abbiamo a disposizione diverse tecniche per ridurne la varianza o running time, in modo tale da aumentarne l'efficienza (Equazione 1.15). Le tecniche di seguito illustrate sono

Stratified Sampling: Suddivisione del dominio di integrazione in  $n$  regioni chiamate *strata*

Importance Sampling: Concentra i punti campionati per la stima dell'integrale utilizzando una densità di distribuzione proporzionale alla funzione integranda, per catturarne più rapidamente i contributi più significativi

Russian Roulette: salta la valutazione di campioni che contribuiscono poco

Splitting: Al contrario di Russian Roulette, aumenta il numero di campioni in alcune dimensioni di un integrale multidimensionale

Ulteriori tecniche di campionamento sono esplorate in Capitolo 2

<sup>9</sup>Tale proprietà è chiamata *Ergodicità*,  $\forall x_{i-1}, x_i \exists' f(x_{i-1}) > 0 \wedge f(x_i) > 0, p_p(x_i, x_{i-1})$

### 1.5.1 Importance Sampling

Approfondiamo la tecnica dell'*Importance Sampling*, la quale è indubbiamente l'approccio più diffuso nel campo della computer grafica, motivo per il quale, nei capitoli successivi, citeremo tecniche per campionare ciascun fattore delle funzioni integrande presentate in seguito.

L'obiettivo dell'Importance sampling è concentrare i samples nelle regioni del dominio di integrazione in cui  $|f(x)|$  sia relativamente elevato, dunque campionando con PDF proporzionale alla funzione integranda  $f(x)$ . Per dimostrare la sua efficacia, consideriamo lo stimatore in Equazione 1.10. Assumendo variabili incorrelate, la sua varianza è pari a

$$V[\tilde{F}] = V\left[\frac{1}{n} \sum_{i=1}^n \frac{f(X_i)}{p(X_i)}\right] = \frac{1}{n^2} \sum_{i=1}^n V\left[\frac{f(X_i)}{p(X_i)}\right] \quad (1.34)$$

Ciò vuol dire che, se il rapporto  $f(X)/p(X)$  è approssimativamente costante, la varianza diminuisce. Nel caso ideale, in cui il rapporto è costante, si ottiene uno stimatore perfetto. Al contrario, se la varianza di tale rapporto è alta, il che consegue dalla scelta di una PDF che non segue l'andamento della funzione integranda, la varianza aumenta.

Molto spesso ci troviamo di fronte ad integrali la cui funzione integranda è prodotto di più funzioni,  $\int f_a(x)f_b(x)dx$ . Spesso è possibile individuare strategie di campionamento per ciascuna presa singolarmente, ma non insieme. È questo il caso dell'integrale della Rendering Equation (Equazione 6.30), nella quale un singolo contributo è rappresentato come prodotto tra BSDF, radianza incidente, fattore coseno. Trovati metodi di campionamento per ciascuno di essi singolarmente  $p_a$  e  $p_b$ , introduciamo la strategia *Multiple Importance Sampling*. Essa consiste nel combinare due strategie di campionamento utilizzanti Importance Sampling e sommarne i contributi mediante una combinazione convessa con pesi opportunamente scelti. Siano  $n_j$  numero di campioni scelti con PDF  $p_j$ , tali che la loro somma sia pari a  $n$ . Allora, il MIS Monte Carlo Estimator

$$\tilde{F} = \sum_{j=1}^n \frac{1}{n_j} \sum_{i=1}^{n_j} w_j(X_{ij}) \frac{f(X_{ij})}{p_j(X_{ij})} \quad (1.35)$$

Dove i pesi sono ricavati applicando minimizzazione utilizzando moltiplicatori di Lagrange [Peg16] per ottenere una formula nota come *balance heuristic*

$$W_j(x) = \frac{n_j p_j(x)}{\sum_{k=1}^{n_j} n_k p_k(x)} \quad (1.36)$$

Da quest'ultima formula si può anche ricavare una controparte manovrabile, che permette di dare più rilevanza a dei campioni piuttosto che altri tramite esponenziazione, detta *power heuristic*

$$w_j(x) = \frac{(n_j p_j(x))^\beta}{\sum_{k=1}^{n_j} (n_k p_k(x))^\beta} \quad (1.37)$$

## Chapter 2

# Campionamento e Ricostruzione con Quasirandom Number Generation

La funzione immagine definita sull'Image plane è una funzione continua, ma l'output del sistema di rendering è una griglia 2D di punti colorati. Scegliere accuratamente la strategia di campionamento dal film plane è fondamentale per migliorare la qualità del risultato finale.

Al fine di poter formalizzare il processo di scelta di campioni dal film plane<sup>1</sup> e valutazione della funzione continua dai campioni (Ricostruzione), si richiamano di seguito due distinte teorie:

Analisi di Fourier del Segnale: Possiamo analizzare in frequenza la varianza di una strategia di campionamento. In particolare, analizzandone il suo spettro di potenza, è possibile analizzare quali frequenze contribuiscono maggiormente alla varianza stessa. Affinchè l'immagine finale risulti di qualità, è desiderabile per una strategia di campionamento presentare varianza quasi nulla o nulla per le basse frequenze.

Un esempio di tale analisi è mostrato in Figura 2.1

Quasirandom Number Generation: Al fine di migliorare la copertura dello spazio campionario dei campioni generati, ci si può affidare a sequenze deterministiche minimizzanti una metrica chiamata *Discrepancy*, definita nelle sezioni successive

Si precisa infine che la parola "pixel" è usata in due contesti differenti

Un *Display Pixel* è un elemento fisico di un display che emette luce di un determinato colore

Un *Image Pixel* è un *campione puntiforme privo di area* della funzione immagine filtrata  $r_f(x, y)$ , i quali definiscono il valore del display pixel integrando una cella definita da quest'ultimo

---

<sup>1</sup>Sinonimo di Image plane, nomenclatura introdotta da [PJH23]

## 2.1 Campionamento

### 2.1.1 Richiami Fourier Analysis

L'analisi di Fourier è usata per valutare la qualità del segnale ricostruito rispetto all'originale. La *Trasformazione di Fourier* consiste nella scomposizione di una funzione continua in un integrale di sinusoidi pesate, le cui ampiezze e fasi costituiscono una funzione complessa a valori reali (frequenza), la trasformata secondo Fourier. Possiamo trasformare tale trasformata dal dominio delle frequenze al dominio di partenza applicando la *Trasformazione inversa di Fourier*. Tale coppia di trasformazioni è definita come

$$F(\nu) = \mathcal{F}\{f(x)\} = \int_{-\infty}^{+\infty} f(x)e^{-\nu 2\pi x} dx \quad (2.1)$$

$$f(x) = \mathcal{F}^{-1}\{F(\nu)\} = \int_{-\infty}^{+\infty} F(\nu)e^{\nu 2\pi x} d\nu \quad (2.2)$$

Le quali possono essere estese a domini multidimensionali applicando la trasformazione a ciascuna coordinata singolarmente

$$\begin{aligned} F(\nu_1, \dots, \nu_d) &= \mathcal{F}\{f(x_1, \dots, x_d)\} \\ &= \int_{-\infty}^{+\infty} \dots \left( \int_{-\infty}^{+\infty} f(x_1, \dots, x_d) e^{-\nu_1 2\pi x_1} dx_1 \right) \dots e^{-\nu_d 2\pi x_d} dx_d \end{aligned} \quad (2.3)$$

$$\begin{aligned} f(x_1, \dots, x_d) &= \mathcal{F}^{-1}\{F(\nu_1, \dots, \nu_d)\} \\ &= \int_{-\infty}^{+\infty} \dots \left( \int_{-\infty}^{+\infty} F(\nu_1, \dots, \nu_d) e^{\nu_1 2\pi x_1} d\nu_1 \right) \dots e^{\nu_d 2\pi x_d} d\nu_d \end{aligned} \quad (2.4)$$

Le seguenti generalizzazioni dunque valgono anche per segnali multidimensionali. La formalizzazione del processo di campionamento (ideale) con periodo  $T$  è data dalla moltiplicazione di una funzione continua per un treno di impulsi  $\mathbf{III}_T(x) = T \sum_{i=-\infty}^{\infty} \delta(x - iT)$ ,

$$f(x)\mathbf{III}_T(x) = Tf(x) \sum_{i=-\infty}^{\infty} \delta(x - iT) = \sum_{i=-\infty}^{\infty} f[i]\delta(x - iT) \quad (2.5)$$

dove  $f[i] = Tf(iT)$  segnale tempo discreto campionato con periodo  $T$

Tale espressione nel dominio della frequenza consiste in una somma di repliche del segnale originale

$$\begin{aligned} \mathcal{F}\{f(x)\mathbf{III}_T(x)\} &= F(\nu) * T\mathbf{III}_{1/T}(\nu) \\ &= \int_{-\infty}^{+\infty} F(u)T\mathbf{III}_{1/T}(\nu - u) du \\ &= \int_{-\infty}^{+\infty} F(u) \sum_{k=-\infty}^{+\infty} \delta\left(\nu - u + \frac{k}{T}\right) du \\ &= \sum_{k=-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u)\delta\left(\nu - u + \frac{k}{T}\right) du \\ &= \sum_{k=-\infty}^{+\infty} F\left(\nu - u + \frac{k}{T}\right) \end{aligned} \quad (2.6)$$

Da cui si enuncia il fondamentale *Teorema di Shannon* del campionamento [Peg16]

**Th. 2.1.1.** — Sia  $F(\nu)$  segnale a banda limitata,  $\exists \nu_b \in \mathbb{R}^+ \ni F(\nu) = 0 \forall |\nu| > \nu_b$ , il *Teorema del campionamento di Nyquist-Shannon* afferma che tale segnale è perfettamente costruibile dai suoi samples, quando è soddisfatto il *Nyquist Criterion*  $\frac{\nu_s}{\nu_b} > 2$ , cioè se la frequenza di campionamento  $\nu_s$  è più grande del *Nyquist Rate*  $2\nu_b$ , o equivalentemente quando la frequenza limite di banda  $\nu_b$  è inferiore della *Nyquist Frequency*  $\frac{\nu_s}{2}$

La distanza tra bande successive è incrementata aumentando la frequenza di campionamento  $\nu_s$ , processo detto *oversampling* di un fattore  $N > 1$  se  $\frac{\nu_s}{\nu_b} > 2N$ , il che permette ai filtri (dopo) di avere una banda di transizione più variabile. Al contrario, *undersampling* consiste nel campionare con frequenza  $\nu_s$  tale che  $N \leq 0$ , il che significa che il teorema del campionamento non è soddisfatto, generando la sovrapposizione delle repliche scalate degli spettri del segnale originale in frequenza, non permettendo la ricostruzione (dopo), il che provoca il fenomeno noto come *Aliasing*, il che si può manifestare in immagini statiche come *pattern di Moiré* ed in animazioni come *wagon-wheel effect*. Con più precisione, ci si riferisce con *Pre-Aliasing* agli artefatti di aliasing introdotti dal campionamento e con *Post-Aliasing* agli artefatti di aliasing introdotti dalla ricostruzione. Non sempre si può aumentare la frequenza di campionamento per migliorare la qualità dell'immagine, in quanto si rischia di incorrere in costi elevati di computazione.

Si noti che segnali reali mai sono banda limitata, per la presenza di discontinuità.

Data una sequenza di campioni, campionata con frequenza di campionamento  $\nu_s = T^{-1}$ , che soddisfa il teorema di Shannon  $f[i]$ , essa può essere utilizzata per ricostruire la funzione originare con l'*interpolazione ideale di Shannon-Whittaker*

$$f(x) = \sum_{j=-\infty}^{+\infty} \text{sinc}(x - kT) f[k] \quad (2.7)$$

Il che è equivalente ad applicare un filtro passa basso di estensione pari alla banda limite del segnale  $\nu_b$ . Tale funzione non è utilizzata nella pratica per lo svantaggio di dover valutare tutti i campioni per la ricostruzione in ogni singolo punto della funzione. Preferiamo metodi che permettono di ricostruire la funzione originale valutandola in un intorno del punto di valutazione, ed è per questo che applichiamo un filtro di ricostruzione.

Ricordiamo la definizione di *Power Spectral Density*, per un segnale con energia

$$\begin{aligned} S_f(\nu) &= \lim_{T \rightarrow \infty} \frac{1}{T} |\mathcal{F}\{f_T(x)\}|^2 \\ f_T(x) &= \begin{cases} f(x) & \text{se } x \in [-T, T] \\ 0 & \text{altrimenti} \end{cases} \end{aligned} \quad (2.8)$$

Tipicamente, per segnali che sono campionati e processati nel dominio tempo-discreto, utilizziamo una seconda definizione, che può essere ottenuta da questa campionando il segnale e prendendo il limite  $T_s \rightarrow 0$

$$S_f(\nu) = F(\nu) \overline{F(\nu)} \quad (2.9)$$

Tale concetto è utile per analizzare statisticamente un Sampling pattern. Talvolta un PSD per un sampling pattern può essere derivata analiticamente, come per il campionamento uniforme [PJH23], altre va calcolato numericamente, come per sampling stocastico.

In generale, L'*aliasing* è ridotto se la PSD di una strategia di campionamento è minima alle basse frequenze

### 2.1.2 Quasirandom Number Generation

Un altro concetto al di fuori della Fourier Analysis per valutare la qualità di samples è il concetto di *Discrepancy*. La generazione di sequenze di numeri a bassa discrepanza ha lo scopo di costruire delle formule matematiche (deterministiche) per campionare punti all'apparenza casuali con migliore uniformità e copertura del dominio che si ottiene con un campionatore uniforme. La Discrepanza infatti è una metrica che misura quanto uniformemente una data successione copre un dato spazio campionario.

([PJH23]) L'idea è quella di suddividere<sup>2</sup> un dominio d-dimensionale  $[0, 1]^d$  in tante regioni, contare il numero di punti in ciascuna regione, e comparare tale numero  $\|\{x_i \in b\}\|$  alla frazione di volume della regione  $V(b)$ . La frazione di punti rispetto al numero di campioni totali di una data regione dovrebbe essere proporzionale alla percentuale di volume totale della regione.

Sceglieremo come suddivisione di  $[0, 1]^d$  una famiglia di parallelepipedi con un vertice sull'origine

$$B = \{[0, v_1] \times [0, v_2] \times \cdots \times [0, v_d]\} \quad (2.10)$$

Da cui, dati  $n$  samples  $P = \{x_1, \dots, x_n\}$ , la *discrepancy* di  $P$  rispetto a  $B$  è definita come

$$D_n(B, P) = \sup_{b \in B} \left| \frac{\|\{x_i \in b\}\|}{n} - V(b) \right| \quad (2.11)$$

In particolare, quando la famiglia di volumi scelta per calcolare la *discrepancy* è l'insieme di parallelepipedi con centro l'origine, il valore che è calcolato è anche chiamato *Star Discrepancy*.

Tale valore può essere calcolato in modo analitico per alcuni particolari insiemi di punti. Consideriamo infatti  $n$  punti equispaziati,  $x_i = \frac{i}{n}$ . Scegliendo come famiglia di volumi

$$B_u = \left\{ \left[0, \frac{1}{n}\right), \left[0, \frac{2}{n}\right), \dots, [0, 1) \right\}$$

Si ottiene star discrepancy

$$D_n^*(\{x_i\}_{i=1}^n, B_u) = \frac{1}{n} \quad (2.12)$$

Una sequenza di punti d-dimensionale è a bassa discrepanza se la sua *star discrepancy* è dell'ordine  $\mathcal{O}\left(\frac{(\log n)^d}{n}\right)$

### 2.1.3 Halton Sampler

Tra i vari campionatori esistenti ([PJH23], [Peg16], [AHH18]), analizziamo con i due framework brevemente spiegati in precedenza, il campionatore basato sulla quasirandom *Sequenza di Halton*. Tale sequenza è costruita utilizzando l'*Inverso Radicale*:

Un numero positivo è esprimibile in base  $b$  come somma di cifre

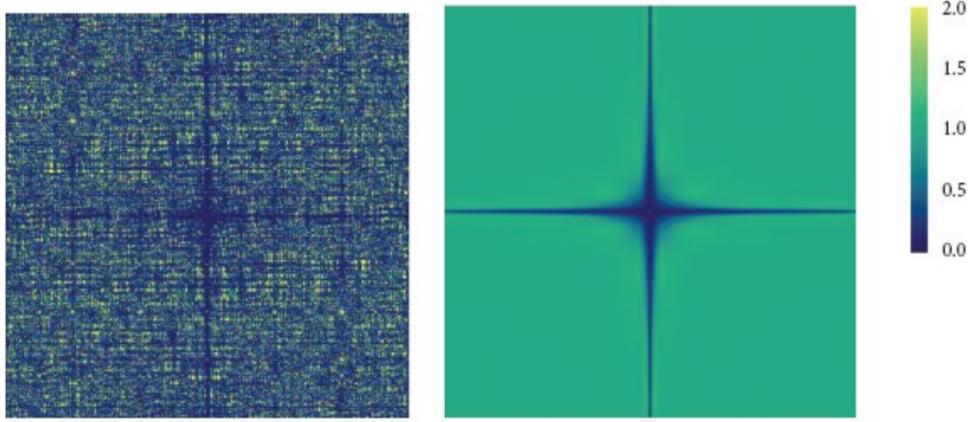
$$a = \sum_{i=1}^m d_i(a) b^{i-1}, \quad d_i(a) \in [0, 1) \quad (2.13)$$

La Funzione Inverso Radicale  $\Phi_b$  in base  $b$  converte un intero nonnegativo in una frazione  $\in [0, 1)$  riflettendo le cifre del numero di partenza rispetto al punto

$$\Phi_b(a) = 0.d_1(a)d_2(a)\dots d_m(a) = \sum_{i=1}^m d_i(a) b^{-i} \quad (2.14)$$

---

<sup>2</sup>Per suddivisione non si intende una partizione del volume totale, ma sottoinsiemi arbitrari di esso la cui unione è il volume totale



(a) Senza Owen Scrambling. Prestazioni (b) Con Owen Scrambling. PSD quasi unitaria nella migliori rispetto al campionamento uni- maggior parte del dominio alle alte frequenze. forme, in quanto presenta contributi min- imi alle basse frequenze, ma contiene ec- cessiva variazione nelle alte frequenze

Figure 2.1: Illustrazione delle Power Spectral Densities del campionatore basato sulla sequenza di Halton con basi 2 e 3, con e senza Owen Scrambling. Immagini da [PJH23]

Un campionatore d-dimensionale può essere costruito scegliendo per ogni dimensione una funzione inverso radicale con base pari ad un numero primo. La generazione di  $n$  campioni consiste nel valutare tali funzioni per  $0, 1, \dots, n - 1$

$$x_a = (\Phi_2(a), \Phi_3(a), \dots, \Phi_{p_d}(a)) \quad (2.15)$$

La star discrepancy della sequenza ottenuta è pari a  $D_n^*(x_a, B_u) = \mathcal{O}\left(\frac{(\log n)^d}{n}\right)$ , cioè il valore ottimale. Si noti che l'implementazione del calcolo di tali radicali è mantenuto con aritmetica intera per evitare di accumulare errore di round-off.

Uno svantaggio di tale sequenza ([PJH23]) è che è deterministica, dunque non ne possiamo valutare la varianza di un integrale stimato con essa. Inoltre, man mano che la base cresce, la sequenza diventa sempre più regolare, rendendo basi alte inutilizzabili. Per risolvere tali problemi si può attuare una permutazione casuale, diversa per ogni cifra, affinchè tali permutazioni generino una sequenza che approcci una distribuzione uniforme. Tale processamento dei punti della sequenza di Halton è chiamata *Scrambling*.

In particolare, una strategia di Scrambling efficace per la sequenza di Halton è l'*Owen Scrambling* [PJH23], per il quale si può dimostrare diminuire l'andamento asintotico dell'errore di integrazione a

$$\mathcal{O}\left(n^{-\frac{3}{2}}(\log n)^{\frac{d-1}{2}}\right) \quad (2.16)$$

Il che è superiore all' $\mathcal{O}(n^{-1/2})$  per Monte Carlo con punti campionati uniformemente.

Lo Scrambling per l'n  $i$ -esima coordinata del  $k$ -esimo punto della sequenza  $x_k^{(i)} = 0.d_1d_2\dots d_m$  è mostrato in B.1 con codice C++ ripreso da [PJH23] e commentato.

## 2.2 Ricostruzione

Un segnale continuo può essere ricostruito dai suoi campioni (perfettamente o meno) con una moltiplicazione nel dominio della frequenza con la funzione di trasferimento di un *filtro di ricostruzione*  $F_r$ , o equivalentemente con la convoluzione del segnale a dominio discreto rappresentato dai campioni con la risposta all'impulso del filtro,  $f_r$ . Ricordiamo che le immagini che andiamo a campionare non sono a banda limitata, in quanto presentano discontinuità. L'obiettivo è dunque trovare una strategia che permetta di minimizzare l'errore tra la funzione ricostruita e la funzione originale.

Il filtro ideale risulta non essere una scelta appropriata ([PJH23]), in quanto esso è affetto dal problema del *ringing* per funzioni oltre la *nyquist frequency*, ciò vuol dire che i bordi degli oggetti risultano replicati nei pixel vicini (più formalmente noto come Fenomeno di Gibbs). Inoltre, sinc ha estensione infinita, in quanto non va a zero per un valore finito della distanza.

Si noti che i filtri di seguito descritti saranno 1D, ma da essi possiamo costruire un filtro d-dimensionale in due modi:

Costruendo un *filtro separabile* (si veda 2.2.1)

$$f(x, y) = f(x)f(y) \quad (2.17)$$

Il cui campionamento può essere compiuto integrando in ogni quadrante, supponendo  $f$  simmetrica

$$4 \int_0^{y_s} \int_0^{x_s} f(x, y) dx dy = \left( 2 \int_0^{x_s} f(x) dx \right) \left( 2 \int_0^{y_s} f(y) dy \right) \quad (2.18)$$

da cui basta applicare Inverse Transform Sampling alla CDF definita da ciascuno degli integrali monodimensionali. Inoltre, per scalare tale filtro per  $t = Tx$ , basta sostituire

$$2 \int_0^{t_s} \frac{1}{|T|} f\left(\frac{t}{T}\right) dt = 2 \int_0^{\frac{t_s}{T}} f(t) dt \quad (2.19)$$

La seconda scelta per costruire un filtro bidimensionale da uno monodimensionale è un *filtro radiale*

$$\begin{aligned} f(x, y) &= \frac{f(\rho)}{2\pi c}, \\ \rho &= \sqrt{x^2 + y^2} \\ c &= \int_0^\infty \rho f(\rho) d\rho \end{aligned} \quad (2.20)$$

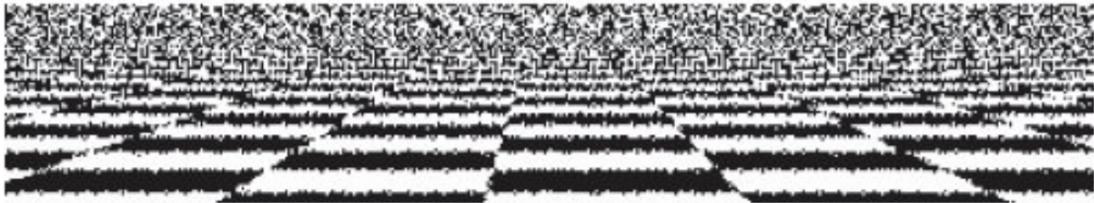


Figure 2.2: Risultato dell'applicazione del campionamento con Scrambled Halton Sequence. Immagine da [PJH23]

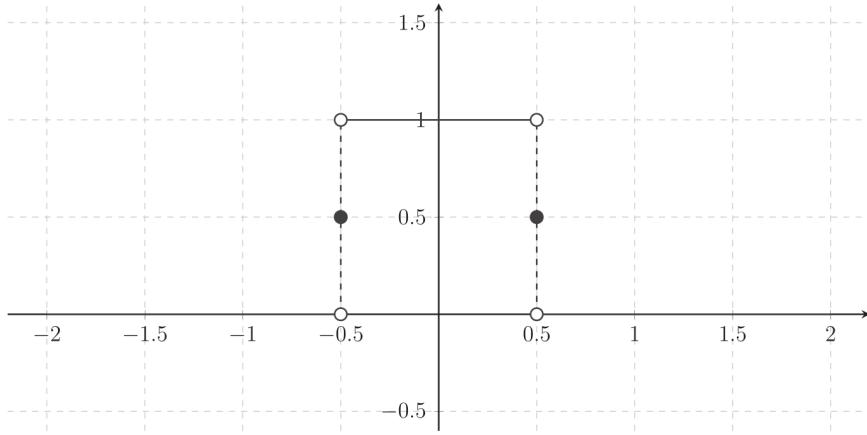


Figure 2.3: Illustrazione della risposta all'impulso del box filter. Immagine da [Peg16]

Per il quale, seguendo ragionamento analogo al precedente [Peg16], possiamo applicare inverse transform sampling

$$\int_0^{\rho_s} \frac{1}{T|T|} \frac{1}{c} f\left(\frac{\rho}{T}\right) \rho d\rho = \int_0^{\frac{\rho_s}{T}} \frac{1}{c} f(\rho) \rho d\rho \quad (2.21)$$

### 2.2.1 Box Filter

Uno dei filtri più comunemente usati è il box filter, il quale rappresenta effettivamente il risultato "standard", utilizzato ad esempio per comparare campionatori. Essendo il più semplice dei filtri, il quale ricostruisce una funzione piecewise constant, introduce post aliasing, anche se il campionamento è effettuato con un elevata sampling frequency.

$$f(x) = \Pi(x) = \begin{cases} 1 & \text{se } |x| < \frac{1}{2} \\ \frac{1}{2} & \text{se } |x| = \frac{1}{2} \\ 0 & \text{se } |x| > \frac{1}{2} \end{cases} \quad (2.22)$$

Il quale ha funzione di trasferimento

$$F(\nu) = \text{sinc}(\pi\nu) \quad (2.23)$$

Utilizzandolo in due dimensioni come un filtro separabile esso ha espressione  $f(x, y) = \Pi(x) \Pi(y)$  con integrale pari a  $4x_b y_b$ , dove  $x_b$  e  $y_b$  semiestensioni del filtro, e funzione di trasferimento  $F(\nu_x) * F(\nu_y)$

Nel caso in cui si utilizzi il box filter come filtro radiale,

$$f(x, y) = f\left(\frac{r}{c}\right) \quad (2.24)$$

dove  $c$  semiestensione del filtro. Esso ha funzione di trasferimento

$$\begin{aligned}
F(\nu_x, \nu_y) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-\iota 2\pi(\nu_x x + \nu_y y)} dx dy \\
&= \int_0^{2\pi} \int_0^{\infty} f\left(\frac{r}{c}\right) e^{-\iota 2\pi r \rho \cos(\varphi - \phi)} r dr d\varphi \\
&\stackrel{\theta = \varphi - \phi}{=} \int_0^{\infty} f\left(\frac{r}{c}\right) \int_{-\phi}^{2\pi - \phi} e^{-\iota 2\pi r \rho \cos \theta} d\theta r dr \\
&= 2\pi \int_0^{\infty} f\left(\frac{r}{c}\right) J_0(2\pi r \rho) r dr \\
&= 2\pi \int_0^{\frac{c}{2}} J_0(2\pi r \rho) r dr \stackrel{\tau = 2\pi r \rho}{=} \frac{1}{2\pi \rho^2} \int_0^{c\pi\rho} J_0(\tau) \tau d\tau \\
&= \frac{c}{2\rho} J_1(c\pi\rho)
\end{aligned} \tag{2.25}$$

Per campionare il filtro separabile (vedi Equazione 5.35), la CDF della box function è pari a

$$2 \int_0^{x_s} \square(x) dx = 2 \int_0^{x_s} = 2x_s \tag{2.26}$$

Dunque possiamo applicare Inverse transform sampling, dato un  $\xi \sim \mathcal{U}(0, 1)$ ,

$$\xi = 2x_s \longrightarrow x_s = \frac{\xi}{2} \tag{2.27}$$

Seguito da una scala per il dominio di interesse. Nel filtro radiale, la CDF

$$8 \int_0^{\rho_s} \square(\rho) \rho d\rho = 8 \int_0^{\rho_s} \rho d\rho = 4\rho_s^2 \tag{2.28}$$

Da cui si può applicare Inverse Transform Sampling

$$\xi = 4\rho_s^2 \longrightarrow \rho_s = \frac{\sqrt{\xi}}{2} \tag{2.29}$$

### 2.2.2 Gaussian Filter

Questo filtro convolve l'immagine non filtrata con un kernel gaussiano. In quanto non presenta lobi negativi, tale filtro sfoca l'immagine originale, il che può essere effetto indesiderato nella ricostruzione. Filtri più complicati, come il Mitchell-Netvali Filter, qui non trattati (si rimanda a [Peg16], [PJH23]), riescono ad ottenere un risultato di qualità simile al Gaussian Filter, preservandone la sharpness. L'equazione di tale filtro

$$f(x) = G(x; 0, \sigma) \tag{2.30}$$

dove

$$G(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.31}$$

$$\mathcal{F}\{G(x, \mu, \sigma)\} = \frac{e^{\frac{\sigma^2\nu^2}{2} + \iota\mu\nu}}{\sqrt{2\pi}\sqrt{\frac{1}{\sigma^2}}\sigma} = \frac{e^{\iota\mu\nu}}{\sigma} G\left(\nu; 0, \frac{1}{\sigma}\right) \tag{2.32}$$

$$\int G(x; \mu, \sigma) dx = \frac{1}{2} \left( 1 + \operatorname{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right) \right) \tag{2.33}$$

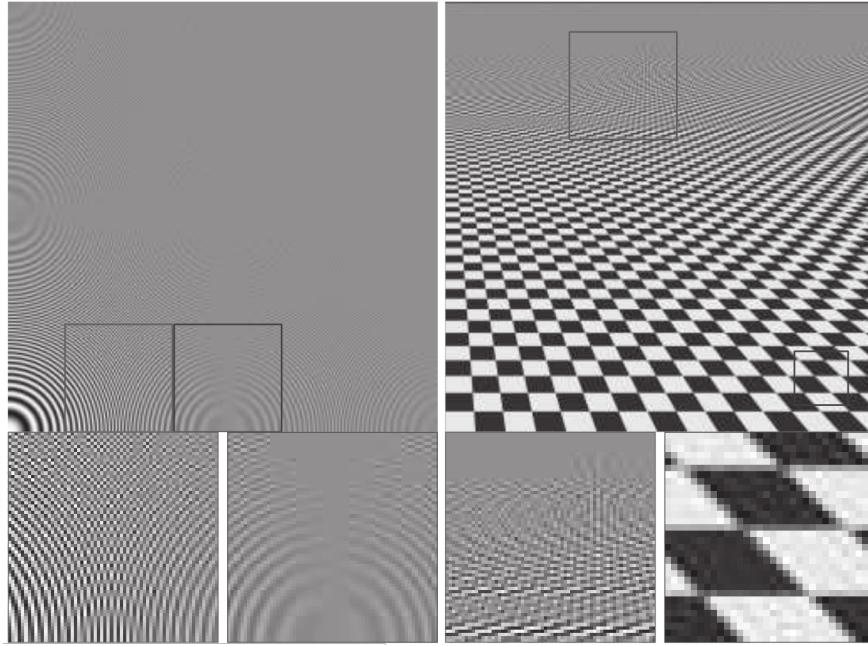


Figure 2.4: Risultato dell'applicazione del box filter ad un checkered pattern

La funzione di trasferimento del Gaussian Filter è dunque

$$F(\nu) = \frac{1}{\sigma} G\left(x; 0, \frac{1}{\sigma}\right) \quad (2.34)$$

Per far in modo che il filtro abbia un raggio definito  $x_b$  e vada a zero per un valore finito, raffiniamo il filtro come

$$f(x) = \begin{cases} G(x; 0, \sigma) - G(x_b; 0, \sigma) & |x| < x_b \\ 0 & \text{altrimenti} \end{cases} \quad (2.35)$$

Supponendo composizione bidimensionale come separable filter, possiamo campionare ciascuna dimensione separatamente. La CDF calcolata in un punto  $x_s$

$$2 \int_0^{x_s} G(x; 0, \sigma) - G(x_b; 0, \sigma) dx = \operatorname{erf}\left(\frac{x_s}{\sqrt{2}\sigma}\right) - \operatorname{erf}\left(\frac{x_b}{\sqrt{2}\sigma}\right) \quad (2.36)$$

Si evince che il suo campionamento risulta essere arduo con Inverse Transform Sampling. Ci sono diverse opzioni

- Utilizzare la versione classica del filtro, per il quale  $x_s = \sqrt{2}\sigma \operatorname{erf}^{-1}(\xi)$  [Peg16]
- Utilizzare una approssimazione polinomiale o piecewise constant, normalizzarla, e utilizzarla come CDF [PJH23]

Tale problema si ripropone nella composizione bidimensionale del Gaussian Filter, il quale nella sua forma classica può essere campionato mediante Inverse Transform Sampling (si rimanda a [Peg16]).

Il metodo alternativo che esaminiamo è campionare una approssimazione piecewise della funzione in questione. Esaminiamo una generica funzione costante definita in  $[0, 1]$  con  $N$

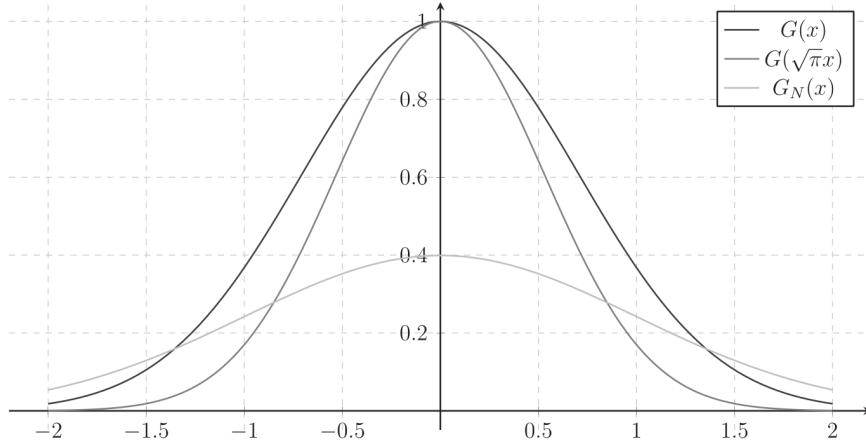


Figure 2.5: Illustrazione di una gaussiana

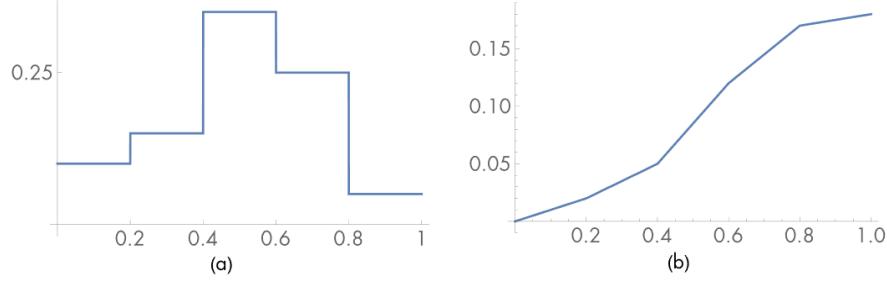


Figure 2.6: Illustrazione funzione definita a tratti. Immagine da [PJH23]

tratti 1D di equo spessore  $\Delta = 1/N$ ,

$$f(x) = \begin{cases} v_0 & x_0 \leq x \leq x_1 \\ v_1 & x_0 \leq x \leq x_2 \\ \vdots & \end{cases} \quad (2.37)$$

$$c = \int_0^1 f(x) dx = \sum_{i=0}^{N-1} \frac{v_i}{N} \quad (2.38)$$

Dunque possiamo costruire una PDF da  $f(x)$  come  $p(x) = \frac{f(x)}{c}$ . Essendo la PDF una funzione costante definita a tratti, la CDF risulterà essere una funzione lineare definita a tratti (Figura 2.6)

$$\begin{aligned} P(x_0) &= 0 \\ P(x_1) &= \int_{x_0}^{x_1} p(x) dx = \frac{v_0}{cN} = P(x_0) + \frac{v_0}{cN} \\ &\vdots \\ P(x) &= \sum_{j=1}^i \left( \frac{v_{j-1}}{cN} \right), \quad x \in [x_{i-1}, x_i] \end{aligned} \quad (2.39)$$

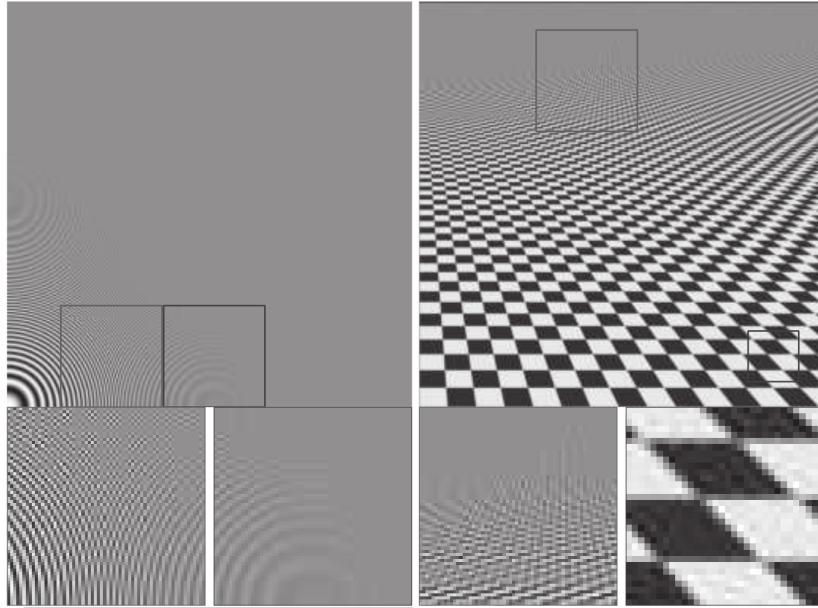


Figure 2.7: Risultato di applicazione del Gaussian Filter radiale ad un checkerboard pattern. Immagine da [Peg16]

Invertire tale funzione consiste nel trovare il valore  $x_s$  tale che

$$\xi = \int_0^{x_s} p(x)dx = P(x_s)$$

In quanto la CDF è crescente, possiamo tradurre tale task nel trovare una coppia di punti  $(x_i, x_{i+1})$  tali che  $P(x_i) \leq \xi$  e  $\xi \leq P_{i+1}$  effettuando una ricerca binaria. Dopodichè, possiamo calcolare quanto distante  $\xi$  è dall'estremo inferiore dell'intervallo definito dai punti trovati  $x_i$ ,  $\Delta\xi = \frac{\xi - x_i}{x_{i+1} - x_i}$ , dove si tiene cura di eseguire la divisione solo se  $x_{i+1} > x_i$ . Il campione per Inverse Transform Sampling può dunque essere posto pari a

$$x_s = x_i + \Delta\xi \quad (2.40)$$

## Chapter 3

# Light Transport Modello e Surface Reflection

Come già accennato in Capitolo 6, ciò che rende difficile la risoluzione della Rendering Equation è il suo legame implicito con la scena attraverso la ray tracing function  $t(\vec{p}, \hat{\omega})$ . L'irradianza spettrale risultante su un punto della superficie potrebbe presentare particolari tinte causate dall'illuminazione indiretta proveniente da un'altra superficie, e a sua volta entrambe sono illuminate da diverse sorgenti luminose. Gli algoritmi di rendering che considerano le mutue interazioni tra le superfici sono detti algoritmi di *global illumination*, mentre algoritmi che fanno uso solo delle informazioni locali alla superficie analizzata algoritmi di *local illumination*.

Gli algoritmi di Monte Carlo forniscono uno strumento potente per risolvere l'equazione di rendering, qui ripetuta

$$L_o(\vec{p}, \hat{\omega}_o) = L_e(\vec{p}, \hat{\omega}_o) + \int_{\mathcal{S}^s} f_s(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) L_i(\vec{p}, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i$$

La quale può essere risolta analiticamente solo nei casi più banali, ad esempio nel caso di una superficie lambertiana, dunque con BRDF pari a  $f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \rho_{hh}/\pi$ , per la quale la radianza incidente ed emessa sono uguali in tutti i punti e direzioni della superficie. Utilizzando tale ipotesi e sostituendo

$$L(\vec{p}, \hat{\omega}) = L_o(\vec{p}, \hat{\omega}) = L_i(\vec{p}, -\hat{\omega})$$

Posso scrivere

$$L = L_e + \rho_{hh} L = L_e + \rho_{hh}(L_e + \rho_{hh}(L_e + \dots)) = \sum_{i=0}^{\infty} L_e \rho_{hh}^i \quad (3.1)$$

Dato che  $\rho_{hh} < 1$  per la conservazione dell'energia, la serie converge a

$$L = \frac{L_e}{1 - \rho_{hh}} \quad (3.2)$$

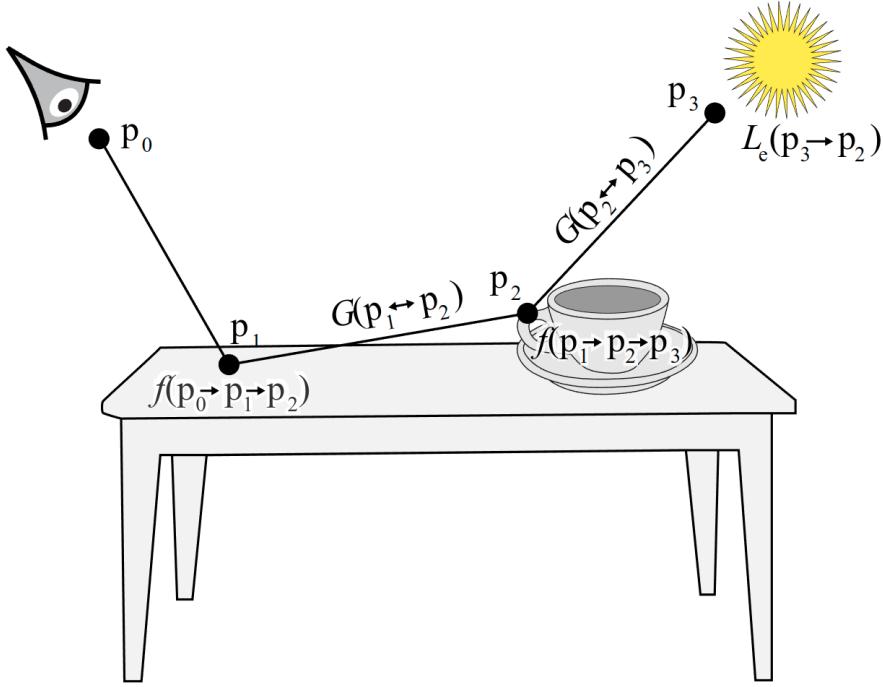


Figure 3.1: Illustrazione della Forma orientata su integrali di percorsi della rendering equation. Immagine da [PJH23]

### 3.1 Path Integral della Rendering Equation

Abbiamo espresso, con Equazione 6.34, la rendering equation in termini dell'unione di tutte le porzioni di superfici visibili dal punto  $\vec{p}$ , qui riportata

$$L(\vec{p}, \hat{pq}) = L_e(\vec{p}, \hat{pq}) + \int_A f_s(\vec{p}, \hat{\omega}_o, \hat{pq}) G(\vec{p}, \vec{q}) L(\vec{q}, -\hat{pq}) dA(\vec{q})$$

Si noti che un'equazione di questo tipo ha struttura nota come *Equazione Integrale di Fredholm del secondo tipo*.

Definendo notazione  $L_{o|e}(\vec{p}_{i+1} \rightarrow \vec{p}_i) = L_{o|e}(\vec{p}_{i+1}, \widehat{\vec{p}_i \vec{p}_{i+1}})$  e  $f_r(\vec{p}_i, \vec{p}_{i+1}, \vec{p}_{i+2}) = f_r(\vec{p}_{i+1}, \widehat{\vec{p}_i \vec{p}_{i+1}}, \widehat{\vec{p}_{i+1} \vec{p}_{i+2}})$ , posso riscrivere tale equazione nella *three-point form*

$$L(\vec{p}_1 \rightarrow \vec{p}_0) = L_e(\vec{p}_1 \rightarrow \vec{p}_0) + \int_A f_s(\vec{p}_0, \vec{p}_1, \vec{p}_2) G(\vec{p}_1, \vec{p}_2) L(\vec{p}_2 \rightarrow \vec{p}_1) dA(\vec{p}_2) \quad (3.3)$$

Dove ricordiamo il termine geometrico che lega le varie superfici è dato da Equazione 6.32. La differenza tra la Rendering Equation sulla sfera di direzioni e questa forma consiste nel fatto che per valutare la prima, campioneremmo da una distribuzione di direzioni per generare raggi e valutare la funzione integranda ricorsivamente. In questa forma, campioneremmo punti sulle superfici seguendo una distribuzione sulle aree, ne calcoleremmo il termine geometrico, e genereremmo raggi per valutare la funzione di visibilità  $V(\vec{p}_i, \vec{p}_{i+1})$ .

La forma più flessibile della Equazione 3.3 consiste in una formulazione che esprime la radianza come un integrale su dei percorsi, rappresentati come punti in uno spazio con alta di-

mensionalità (vedi Figura 3.1), il quale ci permette di stabilire strategie sulla scelta di paths da valutare per poter sviluppare algoritmi efficienti. Per poter esplicitare tale forma espandiamo la definizione ricorsiva di radianza, e raggruppiamo i termini per complessità

$$\begin{aligned}
 L(\vec{p}_1 \rightarrow \vec{p}_0) &= L_e(\vec{p}_1 \rightarrow \vec{p}_0) \\
 &+ \int_A L_e(\vec{p}_2 \rightarrow \vec{p}_1) f_s(\vec{p}_0, \vec{p}_1, \vec{p}_2) G(\vec{p}_1, \vec{p}_2) dA(\vec{p}_2) \\
 &+ \int_A \int_A L_e(\vec{p}_3 \rightarrow \vec{p}_2) f_s(\vec{p}_1, \vec{p}_2, \vec{p}_3) G(\vec{p}_2, \vec{p}_3) \\
 &\quad \times f_s(\vec{p}_0, \vec{p}_1, \vec{p}_2) G(\vec{p}_1, \vec{p}_2) dA(\vec{p}_2) dA(\vec{p}_3) \\
 &+ \dots \\
 &= \sum_{i=1}^{\infty} P(\{\dots, \vec{p}_n\})
 \end{aligned} \tag{3.4}$$

Dove  $\{\dots, \vec{p}_n\} = \{\vec{p}_0, \vec{p}_1, \dots, \vec{p}_n\}$ .

$P(\{\dots, \vec{p}_n\})$  restituisce la quantità di radianza scattered lungo un percorso con gli  $n + 1$  vertici  $\vec{p}_0, \vec{p}_1, \dots, \vec{p}_n$ , nei quali assumiamo  $\vec{p}_0$  è sul film plane e  $p_n$  è su una sorgente luminosa, pari a

$$\begin{aligned}
 P(\{\dots, \vec{p}_n\}) &= \\
 &\underbrace{\int_A \int_A \dots \int_A}_{n-1 \text{ volte}} L_e(\vec{p}_n \rightarrow \vec{p}_{n-1}) \left( \prod_{i=1}^{n-1} f_s(\vec{p}_{i-1}, \vec{p}_i, \vec{p}_{i+1}) G(p_{i+1}, p_i) \right) dA(\vec{p}_2) \dots dA(\vec{p}_n)
 \end{aligned} \tag{3.5}$$

$$P(\{\dots, \vec{p}_0\}) = L_e(\vec{p}_1, \vec{p}_0)$$

La produttoria evidenziata nell'n-esimo path integral è detta *throughput* della path

$$T(\{\dots, \vec{p}_n\}) = \prod_{i=1}^{n-1} f_s(\vec{p}_{i-1}, \vec{p}_i, \vec{p}_{i+1}) G(p_{i+1}, p_i) \tag{3.6}$$

Da cui

$$P(\{\dots, \vec{p}_n\}) = \underbrace{\int_A \int_A \dots \int_A}_{n-1 \text{ volte}} L_e(\vec{p}_n \rightarrow \vec{p}_{n-1}) T(\{\dots, \vec{p}_n\}) dA(\vec{p}_2) \dots dA(\vec{p}_n) \tag{3.7}$$

Notiamo che dei termini  $P(\{\dots, \vec{p}_i\})$  potrebbero contenere distribuzioni di Dirac all'interno della loro funzione integranda, dovuta a determinati modelli di sorgenti luminose o BSDF. Dunque le implementazioni devono gestire tale caso particolare in quanto riduce la quantità di termini della somma da stimare. Si consideri l'esempio da [PJH23] per l'illuminazione diretta con una sorgente puntiforme descritta da una distribuzione di Dirac.

Un approccio per partizionare la somma infinita di path integrals è la seguente:

$$L(\vec{p}_1 \rightarrow \vec{p}_0) = P(\{\dots, \vec{p}_1\}) + P(\{\dots, \vec{p}_2\}) + \sum_{i=3}^{\infty} P(\{\dots, \vec{p}_i\}) \tag{3.8}$$

Dove il primo termine è calcolato facilmente valutando la radianza emessa dal punto  $\vec{p}_1$  verso  $\vec{p}_0$ , il secondo integrale è stimato con accuratezza elevata e i restanti termini della somma sono stimati con un'approssimazione più grossolana. Inoltre, si potrebbero anche partizionare gli integrali a seconda della categoria di sorgenti luminose e BSDF, ad esempio per distinguere sorgenti grandi da sorgenti piccole, e BSDF contenenti  $\delta(x)$  dal resto.

## 3.2 Path Tracing

Avendo ricavato una espressione facilmente manovrabile, siamo pronti a formulare un'algoritmo per la risoluzione dell'equazione di Rendering.

Chiaramente nella nostra computazione non abbiamo la possibilità di valutare una serie, per cui applichiamo *Russian Roulette* (??), valutando la terminazione della valutazione della serie ad ogni termine con probabilità  $q_i$ , definite a priori. Un approccio per la scelta di tali probabilità è la seguente (giustificata in [PJH23])

$$q_j \sim \mathcal{U} \left( 0, 1 - \max_{\lambda \in \Lambda} \left\{ \eta \prod_{i=1}^{j-2} \frac{f_s(\vec{p}_{j-1}, \vec{p}_j, \vec{p}_{j+1})}{p_\omega(\hat{\omega}_j)} \right\} \right) \quad (3.9)$$

dove  $\hat{\omega}_j = (\vec{p}_{j+1} - \vec{p}_j) / \|\vec{p}_{j+1} - \vec{p}_j\|$  e  $p_\omega(\hat{\omega}_j)$  PDF della BSDF (6.3)

Applicando la compensazione per tale tecnica, lo stimatore unbiased diventa

$$\frac{1}{1 - q_1} \left( P(\{\dots, \vec{p}_1\}) + \frac{1}{1 - q_2} \left( P(\{\dots, \vec{p}_2\}) + \frac{1}{1 - q_3} (P(\{\dots, \vec{p}_3\}) + \dots) \right) \right) \quad (3.10)$$

In quanto il throughput totale della path

$$\beta = \prod_{i=1}^{j-2} \frac{f_s(\vec{p}_{j-1}, \vec{p}_j, \vec{p}_{j+1})}{p_\omega(\hat{\omega}_j)}$$

decresce man mano che il tracing continua, è garantito che la valutazione della path termini in un tempo limitato. Si noti che nonostante Russian Roulette aumenta l'MSE, in quanto si accumulano meno contributi, l'efficienza aumenta comunque per la notevole diminuzione del Running Time.

Per stimare ciascun path integral  $P(\{\dots, \vec{p}_n\})$ , necessitiamo di  $n+1$  vertici dei quali  $\vec{p}_0$  sia sul film plane e  $\vec{p}_n$  sia su una sorgente luminosa. Un approccio potrebbe essere campionare, dopo aver scelto il vertice sul film plane da cui partire,  $n - 2$  punti da tutte le superfici della scena con densità di distribuzione uniforme, e l'ultimo campione su una delle superfici luminose presenti nella scena con equa probabilità. Tale approccio però non conta la mutua visibilità tra le superfici e può condurre a risultati con alta varianza, per via della presenza di paths con contributo nullo, e risultati incorretti, senza contare il fatto che tale metodo non supporta una gestione per le distribuzioni contenenti il delta di Dirac. Una strategia valida è rappresentata invece dalla *Costruzione Incrementale del percorso* [PJH23]:

partendo da un punto del film plane  $\vec{p}_0$  (per il quale la direzione del raggio visivo è determinata geometricamente dal camera model), si traccia un raggio per intersecare una superficie in  $\vec{p}_1$ . Da qui si ripete il procedimento descritto a seguire.

da  $\vec{p}_i$  campiona una direzione dalla BSDF per ottenere una direzione  $\hat{\omega}_i$  la quale, tracciato un raggio lungo essa, che conduce ad una nuova intersezione  $\vec{p}_{i+1}$ .

L'ultimo punto del path  $\vec{p}_n$  deve terminare su una sorgente luminosa. L'appuccio utilizzato per la determinazione della sorgente da utilizzare è il Multiple Importance Sampling, utilizzando come densità di distribuzione la PDF della BSDF dell'ultima superficie,  $p_\omega$ , e la PDF della sorgente luminosa selezionata per il calcolo della radianza uscente dall'ultima superficie,  $p_l$ . Si ricordi che tali PDF hanno come dominio la sfera unitaria di possibili direzioni, ed in quanto il path integral è un integrale su un area, bisogna applicare la correzione (vedi Equazione 6.34)

$$p_A(\vec{p}_i) = p_\omega(\hat{\omega}_{j-1}) \frac{|\langle \hat{n}, \vec{p}_i \rangle|}{\|\vec{p}_{i-1} - \vec{p}_i\|^2} \quad (3.11)$$

andandosi a semplificare con il termine geometrico<sup>1</sup>

$$\begin{aligned}
 P(\{\dots, \vec{p}_n\}) &\approx L_e(\vec{p}_n \rightarrow \vec{p}_{n-1}) \prod_{i=1}^{n-1} \frac{f_s(\vec{p}_{i-1}, \vec{p}_i, \vec{p}_{i+1}) G(\vec{p}_{i+1}, \vec{p}_i)}{p_A(\vec{p}_{i+1})} \\
 &= L_e(\vec{p}_n \rightarrow \vec{p}_{n-1}) \prod_{i=1}^{n-1} \frac{f_s(\vec{p}_{i-1}, \vec{p}_i, \vec{p}_{i+1}) V(\vec{p}_{i+1}, \vec{p}_i) \frac{|\cos \theta_{i+1}| |\cos \theta_i|}{\|\vec{p}_{i+1} - \vec{p}_i\|^2}}{p_A(\vec{p}_{i+1})} \\
 &= L_e(\vec{p}_n \rightarrow \vec{p}_{n-1}) \prod_{i=1}^{n-1} \frac{f_s(\vec{p}_{i-1}, \vec{p}_i, \vec{p}_{i+1}) V(\vec{p}_{i+1}, \vec{p}_i) \frac{|\cos \theta_{i+1}| |\cos \theta_i|}{\|\vec{p}_i - \vec{p}_{i+1}\|^2}}{p_\omega(\omega_i) \frac{|\cos \theta_{i+1}|}{\|\vec{p}_i - \vec{p}_{i+1}\|^2}} \\
 &= L_e(\vec{p}_n \rightarrow \vec{p}_{n-1}) \prod_{i=1}^{n-1} \frac{f_s(\vec{p}_{i-1}, \vec{p}_i, \vec{p}_{i+1}) V(\vec{p}_{i+1}, \vec{p}_i) |\cos \theta_i|}{p_\omega(\omega_i)}
 \end{aligned} \tag{3.12}$$

Dopodichè suddividiamo il termine di Direct Lighting applicando MIS

$$\begin{aligned}
 P(\{\dots, \vec{p}_i\}) &\approx w_l(\hat{\omega}_l) \frac{L_e(\vec{p}_l \rightarrow \vec{p}_{i-1}) f_s(\vec{p}_{i-2}, \vec{p}_{i-1}, \vec{p}_l) |\cos \theta_l| V(\vec{p}_{i-1}, \vec{p}_l)}{p_l(\hat{\omega}_l)} \beta \\
 &\quad + w_b(\hat{\omega}_b) \frac{L_e(\vec{p}_b \rightarrow \vec{p}_{i-1}) f_s(\vec{p}_{i-2}, \vec{p}_{i-1}, \vec{p}_b) |\cos \theta_b| V(\vec{p}_{i-1}, \vec{p}_b)}{p_b(\hat{\omega}_b)} \beta
 \end{aligned} \tag{3.13}$$

dove i pesi sono calcolati con la *balance heuristic* o la *power heuristic* (Sezione 1.5.1).

L'algoritmo completo implementante le tecniche qui discusse può essere trovato in [PJH23].

### 3.3 Applicazione del Metodo di Monte Carlo: Esempio Path Tracing Semplificato

Al fine di chiarificare come tali metodologie sono applicate in pratica, si propone di seguito un Path Tracer semplificato, supportante

- Tre tipi di BRDF, in particolare Lambertiana (DIFF), Speculare opaca (SPEC), Speculare traslucente (REFL)
- Russian Roulette, applicata per terminare la valutazione di una path e per scegliere tra riflessione o trasmissione nel caso di BRDF traslucente
- Stratified Sampling, dividendo ciascun pixel cell in 4 caselle, e campionando per ciascuna di esse un numero di samples pari all'input da linea di comando.
- Ciascun sample in ciascun strata è scelto casualmente con PDF triangolare

Il quale crea una resa della celebre *Cornell Box*, scena utilizzata per testare 3D rendering software creata e presentata nel SIGGRAPH 1984 (vedi Figura 3.2).

In questa implementazione, la scena è composta esclusivamente da sfere, definite attraverso un raggio, posizione, radianza emessa, riflettanza spettrale (formalmente DHR o albedo, Equazione 6.4, informalmente "colore"), tipologia di materiale.

```

⟨Definizione Sphere struct⟩ +=

struct Sphere {
    double rad;
    Vec p, e, c;

```

<sup>1</sup>omesso l'Importance Sampling per semplicità e per evidenziare i passaggi seguiti

### 3.3. APPLICAZIONE DEL METODO DI MONTE CARLO: ESEMPIO PATH TRACING SEMPLIFICATO

---

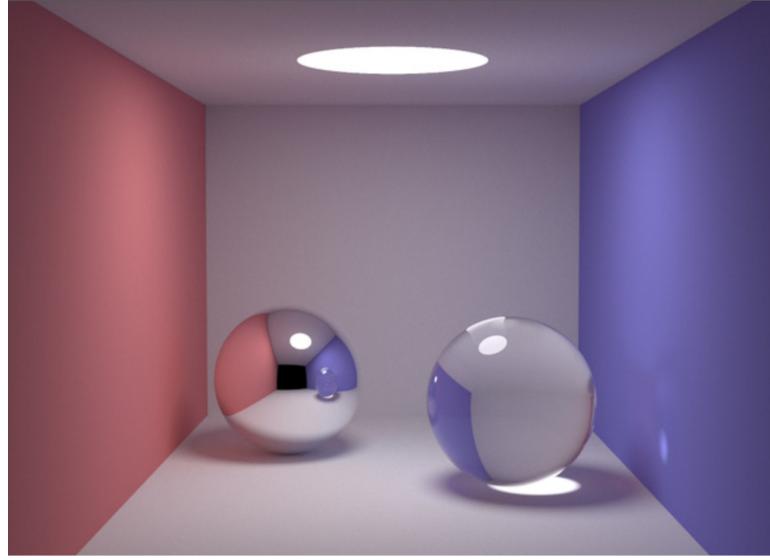


Figure 3.2: Cornell Box scene utilizzata per il test.

```
Refl_t refl;
⟨Costruttore (B.5)⟩
⟨Metodo per intersezione (B.5)⟩
};
```

Da cui possiamo definire i lati del cubo come sfere con raggio elevato, la sorgente luminosa come sfera con radianza emessa non nulla riflettanza spettrale nulla, e le comuni sfere, con radianza emessa nulla e riflettanza spettrale non nulla, definente la tinta del materiale

*(Definizione della scena)* +≡

```
Sphere spheres[] = {
    Sphere(1e5, Vec( 1e5+1,40.8,81.6), Vec(),Vec(.75,.25,.25),DIFF),
    Sphere(1e5, Vec(-1e5+99,40.8,81.6),Vec(),Vec(.25,.25,.75),DIFF),
    Sphere(1e5, Vec(50,40.8, 1e5), Vec(),Vec(.75,.75,.75),DIFF),
    Sphere(1e5, Vec(50,40.8,-1e5+170), Vec(),Vec(), DIFF),
    Sphere(1e5, Vec(50, 1e5, 81.6), Vec(),Vec(.75,.75,.75),DIFF),
    Sphere(1e5, Vec(50,-1e5+81.6,81.6),Vec(),Vec(.75,.75,.75),DIFF),
    Sphere(16.5,Vec(27,16.5,47), Vec(),Vec(1,1,1)*.999, SPEC),
    Sphere(16.5,Vec(73,16.5,78), Vec(),Vec(1,1,1)*.999, REF),
    Sphere(600, Vec(50,681.6,-27,81.6),Vec(12,12,12), Vec(), DIFF)
};
```

Il codice parte nella funzione `main`, la quale opera nel seguente modo

*(main outline)* +≡

```
int main(int argc, char *argv[]) {
    // definisci risoluzione griglia film plane (schermo) e numero di campioni
    // per pixel da linea di comando
    int w=1024, h=768, samps = argc==2 ? atoi(argv[1])/4 : 1;
```

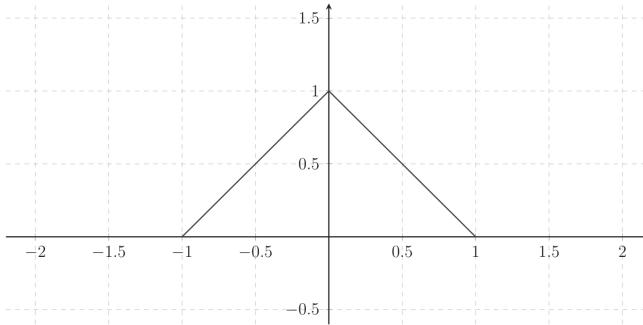


Figure 3.3: Grafico funzione triangolare  $\wedge(x)$ . Immagine da [Peg16]

```

// definisci posizione e direzione osservatore, utilizzando il
// Pinhole Camera Model
Ray cam(Vec(50,52,295.6), Vec(0,-0.042612,-1).norm());

// definisci direzione orizzontale cx e verticale cy
// della camera. "%" e' definito come prodotto vettoriale vedi B.5
Vec cx=Vec(w*.5135/h), cy=(cx/cam.d).norm()*.5135;

// variabile che conterra' la stima effettuata per ogni sample
Vec r;

// Buffer che conterra' l'immagine finale
Vec* c=new Vec[w*h];

// direttiva per la programmazione multicore con OpenMP
#pragma omp parallel for schedule(dynamic, 1) private(r)
// per ogni pixel
for (int y=0; y<h; y++) {
    printf(stderr, "\rRendering (%d spp) %5.2f%%", samps*4, 100.*y/(h-1));
    for (unsigned short x=0, Xi[3]={0,0,y*y*y}; x<w; x++)
        {Calcola il colore mostrato dal pixel}
}

// salva l'immagine sintetizzata in un file
FILE *f = fopen("image.ppm", "w");
fprintf(f, "P3\n%d %d\n%d\n", w, h, 255);
for (int i=0; i<w*h; i++)
    fprintf(f, "%d %d %d ",.toInt(c[i].x), toInt(c[i].y), toInt(c[i].z));
}

```

Per ogni pixel cell, ciascuno dei quattro strata viene campionato con un numero di samples pari all'input dato al programma in console (default: 1), con PDF pari alla funzione triangolare

### 3.3. APPLICAZIONE DEL METODO DI MONTE CARLO: ESEMPIO PATH TRACING SEMPLIFICATO

---

normalizzata, cioè

$$\begin{aligned} f(x, y) &= \wedge(x) \wedge (y) \\ \wedge(x) &= \max\{0, 1 - |x|\} \end{aligned} \quad (3.14)$$

Da cui Inverse Transform Sampling può essere applicato calcolandone CDF separabile

$$2 \int_0^{t_s} \wedge(t) dt = 2 \int_0^{t_s} 1 - |t| dt = 2 \left[ t - \frac{t^2}{2} \right]_{t=0}^{t_s} = t_s(2 - t_s) \quad (3.15)$$

e invertendola, dato un  $\xi \sim \mathcal{U}(0, 1)$  (vedi Sezione 1.4.1)

$$\xi = t_s(2 - t_s) \rightarrow t_s = 1 - \sqrt{1 - \xi} \quad (3.16)$$

Tale espressione può essere utilizzata per calcolare ascissa e ordinata del subpixel sample

*⟨Calcola il colore mostrato dal pixel⟩ +≡*

```
// per ogni strata
for (int sy=0, i=(h-y-1)*w+x; sy<2; sy++){
    for (int sx=0; sx<2; sx++, r=Vec()){
        // per ogni campione nello strata
        for (int s=0; s<samps; s++){
            // genera un numero casuale basato sul seed Xi con PDF
            // triangolare per ogni coordinata del campione...
            double r1=erand48(Xi), dx=1-sqrt(1-r1);
            double r2=erand48(Xi), dy=1-sqrt(1-r2);

            // ... e utilizzalo come scostamento dal centro dello strata
            // (tecnica nota come jittering), tramite il quale possiamo
            // definire la direzione del raggio da proiettare dall'osservatore, di
            // posizione cam.o
            Vec d = cx*( (sx+.5 + dx)/2 + x)/w - .5)
                + cy*( (sy+.5 + dy)/2 + y)/h - .5) + cam.d;

            // Applica lo Stimatore di Monte Carlo
        }
    }
}

// Il colore finale del pixel e' pari alla media dei quattro colori
// calcolati per ogni strata
c[i] = c[i] + Vec(clamp(r.x), clamp(r.y), clamp(r.z))*25;
}
```

Dove giungiamo finalmente all'applicazione dell'Equazione 1.9

*⟨Applica lo Stimatore di Monte Carlo⟩ +≡*

```
r = r + radience(Ray(cam.o+d*140,d.norm()),0,Xi)*(1./samps);
```

Dove la funzione radience è una funzione ricorsiva, che ritorna un campione della funzione immagine, i cui parametri sono

- Raggio da proiettare
- ray depth corrente, ovvero il numero di "rimbalzi" già effettuati

- seed per RNG
- booleano che indica se contare radianza emessa o meno (default: true)

La cui definizione è la seguente

```
<Funzione radiance> +≡

Vec radiance(const Ray &r, int depth, unsigned short *Xi){
    double t;
    int id=0;
    if (!intersect(r, t, id)) return Vec();

    const Sphere &obj = spheres[id];

    /*
     * Rispettivamente:
     * x <- punto di intersezione tra sfera e raggio
     * n <- normale della sfera
     * nl <- normale riorientata per formare un angolo acuto con l'osservatore
     * f <- riflettanza della sfera, modula il contributo della BRDF
     */
    Vec x=r.o+r.d*t, n=(x-obj.p).norm(), nl=n.dot(r.d)<0?n:n*-1, f=obj.c;

    // Applica Russian Roulette per tutti i raggi con depth > 5, dunque
    // - Dalla quinta superficie intersecata in poi, con probabilita' p.
    // - Applica la compensazione (Equazione 3.10) in caso il raggio persista.
    // - Confronta la probabilita' estratta con la componente massima della
    //   riflettanza per decidere se terminare il raggio o meno
    double p = f.x>f.y && f.x>f.z ? f.x:
                    f.y>f.z ? f.y:
                    f.z;
    if (++depth>5) {
        if (erand48(Xi)<p) f=f*(1/p);
        else return obj.e;
    }

    // Se la sfera ha BRDF lambertiana
    if (obj.refl == DIFF) {
        <Calcola contributo per BRDF Lambertiana e reitera>
    }

    // Se la sfera ha BRDF speculare opaca
    else if (obj.refl == SPEC)
        <Calcola contributo per BRDF Speculare Opaca e reitera>

    // Se la sfera ha BRDF speculare translucente
    <Calcola contributo per BRDF Speculare Translucente e reitera>
}
```

In tutti e tre i casi di BRDF, il risultato è pari alla radianza emessa sommata alla radianza riflessa/trasmessa, come prescrive la Rendering Equation (Equazione 6.30). Per quest'ultima,

### 3.3. APPLICAZIONE DEL METODO DI MONTE CARLO: ESEMPIO PATH TRACING SEMPLIFICATO

---

si esegue una chiamata ricorsiva alla funzione `radiance`, con direzione campionata dipendente dalla BRDF in questione.

Per una BRDF lambertiana, Equazione 6.37, il campionamento può seguire una cosine-lobe distribution di grado 1, approccio illustrato in Equazione 6.40.

In quanto un approccio basato sui path presentato nelle sezioni precedenti richiederebbe un codice molto lungo, ci limitiamo a seguire un approccio basato sulle direzioni, e supponendo BRDF Lambertiana la Rendering Equation diventa

$$\begin{aligned} L_o(\vec{p}, \hat{\omega}_o) &= L_e(\vec{p}, \hat{\omega}_o) + \int_{\mathcal{H}^2(\hat{n})} L_i(\vec{p}, \hat{\omega}_i) \frac{\rho(\vec{p})}{\pi} \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i \\ &= L_e(\vec{p}, \hat{\omega}_i) + \frac{\rho(\vec{p})}{\pi} \int_{\mathcal{H}^2(\hat{n})} L_i(\vec{p}, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \end{aligned} \quad (3.17)$$

applicando lo stimatore di Monte Carlo con un singolo campione ( $n = 1$ )

$$\begin{aligned} L_o(\vec{p}, \hat{\omega}_o) &\approx L_e(\vec{p}, \hat{\omega}_o) + \frac{\|\mathcal{H}^2(\hat{n})\|}{n} \frac{\rho}{\pi} \frac{L_i(\vec{p}, \hat{\omega}) |\langle \hat{n}, \hat{\omega} \rangle|}{p_\omega(\hat{\omega})} \\ &= L_e(\vec{p}, \hat{\omega}_o) + \rho \frac{L_i(\vec{p}, \hat{\omega}) |\langle \hat{n}, \hat{\omega} \rangle|}{p_\omega(\hat{\omega})} \end{aligned} \quad (3.18)$$

Che è l'equazione implementata nella riga finale del seguente codice

```
<Calcola contributo per BRDF Lambertiana e reitera> +=

// φ, angolo azimuthale (Figura 6.5)
double r1=2*M_PI*erand48(Xi);

// sin θ, dove θ angolo rispetto allo zenith (Figura 6.5)
double r2=erand48(Xi), r2s=sqrt(r2);

/*
 * definizione del sistema di riferimento con i tre versori
 * w <- normale, direzione dello zenith
 * u <- direzione tangente alla superficie
 * v <- direzione bitangente alla superficie
 */
Vec w=nl, u=((fabs(w.x)>.1?Vec(0,1):Vec(1))%w).norm(), v=w%u;

// conversione da coordinate sferiche a coordinate cartesiane
Vec d = (u*cos(r1)*r2s + v*sin(r1)*r2s + w*sqrt(1-r2)).norm();

// calcolo della radianza finale con ricorsione
// (f.mult indica prodotto di Hadamard)
return obj.e + f.mult(radiance(Ray(x,d),depth,Xi));
```

Una BRDF speculare opaca non calcola alcuna direzione casuale, per via della distribuzione di Dirac presente nella sua BRDF (Equazione 6.45), dunque riflette solamente nella direzione

indicata in Equazione 6.7

$$\begin{aligned}
 L_o(\vec{p}, \hat{\omega}_o) &= L_e(\vec{p}, \hat{\omega}_o) + \int_{\mathcal{H}^2(\hat{n})} L_i(\vec{p}, \hat{\omega}_i) F_r(|\langle \hat{n}, \hat{\omega}_o \rangle|) \frac{\delta(\hat{\omega}_i - \hat{\omega}_r)}{|\langle \hat{n}, \hat{\omega}_i \rangle|} |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \\
 &= L_e(\vec{p}, \hat{\omega}_o) + F_r(|\langle \hat{n}, \hat{\omega}_o \rangle|) L_i(\vec{p}, \hat{\omega}_r) \\
 &= L_e(\vec{p}, \hat{\omega}_o) + \bar{F}_r L_i(\vec{p}, \hat{\omega}_r)
 \end{aligned} \tag{3.19}$$

Dove  $\bar{F}_r$  indica il valor medio della Fresnel Reflectance, rappresentato dal parametro `c` memorizzato dalle Spheres, approssimazione valida solamente per i dielettrici.

```
<Calcola contributo per BRDF Speculare Opaca e reitera> +=
    return obj.e + f.mult(radiance(Ray(x,r.d-n*2*n.dot(r.d)),depth,Xi));
```

Infine, la BRDF speculare traslucente è pensata per modellare una superficie vetrosa. Infatti, il suo IOR è assunto pari a  $\eta_t = 1.5$ , mentre si assume indice di rifrazione unitario per l'ambiente circostante. Le equazioni sono simili alle precedenti con l'aggiunta del termine di radianza trasmessa. Il funzionamento è il seguente:

1. calcola l'indice di rifrazione relativo, pari al rapporto IOR entrante e IOR uscente. Calcola anzi tempo la direzione riflessa
2. Se si è nel caso di riflessione interna, si fa ricorsione con la direzione riflessa
3. Altrimenti calcola anzi tempo la direzione trasmessa (Equazione 6.10)
4. Calcola la riflettanza per la direzione di incidenza con l'approssimazione di Schlick, utilizzabile in quanto si sta supponendo vetro come materiale, che è un dielettrico (Equazione 6.18).
5. Calcola il contributo di raggio riflesso e raggio trasmesso
6. Russian Roulette: se `depth > 2`, allora uno dei due raggi, riflesso o trasmesso, è terminato

Il codice è il seguente:

```
<Calcola contributo per BRDF Speculare Translucente e reitera> +=
// Calcola direzione riflessa
Ray reflRay(x, r.d-n*2*n.dot(r.d));

// calcola indice di rifrazione relativo e ddn = <n, ω_o>
bool into = n.dot(nl)>0;
double nc=1, nt=1.5, nnt=into?nc/nt:nt/nc, ddn=r.d.dot(nl)

// test per riflessione interna totale
double cos2t = 1-nnt*nnt*(1-ddn*ddn);
if (cos2t<0)
    return obj.e + f.mult(radiance(reflRay,depth,Xi));

// calcola direzione di trasmisione
Vec tdir = (r.d*nnt - n*((into?1:-1)*(ddn*nnt+sqrt(cos2t)))).norm();

// calcola coefficienti di fresnel di riflessione e trasmisione
double a=nt-nc, b=nt+nc, R0=a*a/(b*b), c = 1-(into?-ddn:tdir.dot(n));
double Re=R0+(1-R0)*c*c*c*c*c;
double Tr=1-Re;
```

### 3.3. APPLICAZIONE DEL METODO DI MONTE CARLO: ESEMPIO PATH TRACING SEMPLIFICATO

---

```
// Applicazione Russian Roulette: se depth del raggio e' maggiore
// di 2, allora uno dei due raggi e' terminato. La scelta tra il raggio
// riflesso reflRay e raggio trasmesso Ray(x,tdir) e'
// dettata dalla riflettanza di Fresnel Re: piu' e' alta, e piu'
// la probabilita' P di scegliere di terminare
// il raggio rifratto e' alta
double P=.25+.5*Re;

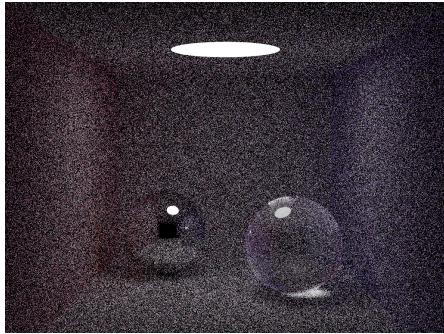
// Applica la compensazione prevista dalla tecnica Russian Roulette
double RP=Re/P;
double TP=Tr/(1-P);

// Calcola il contributo di radianza finale
return obj.e + f.mult(depth>2 ? (erand48(Xi)<P ? radiance(reflRay,depth,Xi)*RP:
                                         radiance(Ray(x,tdir),depth,Xi)*TP)
                                         : radiance(reflRay,depth,Xi)*Re
                                         + radiance(Ray(x,tdir),depth,Xi)*Tr);
```

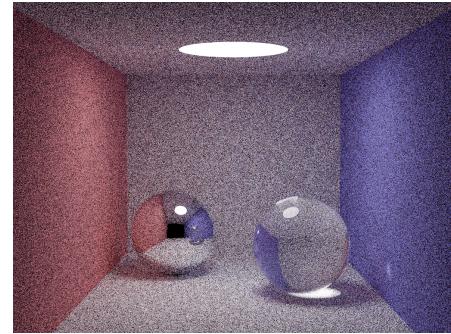
I risultati dell'algoritmo sono mostrati in Figura 3.4, seguite da Tabella 3.1. Quest'ultima riporta, per ogni simulazione eseguita, con numero differente di pixel per stratum, campionati con PDF triangolare, la varianza di ciascuna immagine risultante per red channel, green channel, blue channel, e luminance channel. Quest'ultimo, in particolare, è stato calcolato per ogni pixel come luma channel definito dal color space standardizzato ITU BT.709  $YC_B C_R$

$$Y = 0.2126R + 0.7152G + 0.0722B$$

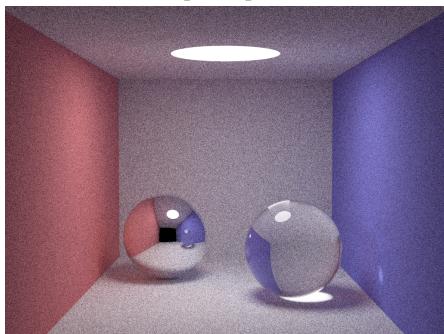
Si pone l'accento su esso in quanto sono visibili incrementi notevoli di qualità dell'immagine ogni qual volta la varianza per tale canale diminuisce di 0.01 Il codice completo può essere trovato in Appendice B.5 assieme ai comandi per la compilazione.



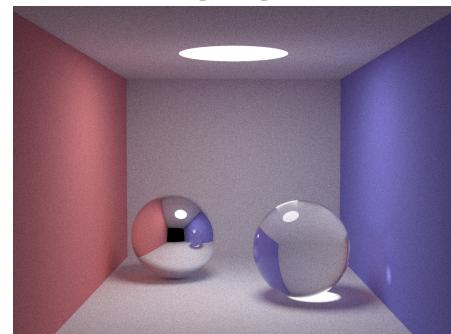
(a) 8 campioni per stratum



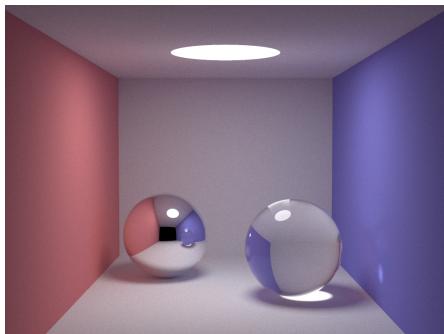
(b) 40 campioni per stratum



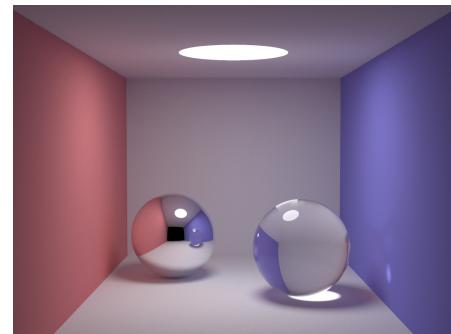
(c) 200 campioni per stratum



(d) 1000 campioni per stratum



(e) 5000 campioni per stratum



(f) 25000 campioni per stratum

Figure 3.4: Risultati dell'esecuzione dell'algoritmo con numero crescente di campioni per pixel

### 3.3. APPLICAZIONE DEL METODO DI MONTE CARLO: ESEMPIO PATH TRACING SEMPLIFICATO

---

Varianza trovata per le immagini risultanti <sup>1</sup>	
8 campioni per stratum	
Red channel variance	0.07344142
Green channel variance	0.06190188
Blue channel variance	0.07339846
Luminance variance	0.06380390
40 campioni per stratum	
Red channel variance	0.05132978
Green channel variance	0.04714374
Blue channel variance	0.05175589
Luminance variance	0.04621717
200 campioni per stratum	
Red channel variance	0.02482136
Green channel variance	0.02453323
Blue channel variance	0.02517796
Luminance variance	0.02276842
1000 campioni per stratum	
Red channel variance	0.02023584
Green channel variance	0.02019825
Blue channel variance	0.02031221
Luminance variance	0.01840096
5000 campioni per stratum	
Red channel variance	0.01904086
Green channel variance	0.01936775
Blue channel variance	0.01930308
Luminance variance	0.01754279
25000 campioni per strata	
Red channel variance	0.01885027
Green channel variance	0.01919554
Blue channel variance	0.01911878
Luminance variance	0.01737635

Table 3.1: Tabella delle varianze

<sup>1</sup> Ciascuna componente RGB è assunta normalizzata

**Part II**

**Fondamenti Fisici**

## Chapter 4

# Richiami sull'elettromagnetismo e Radiometria

### 4.1 Radiometria

Branche dell'ottica prese come riferimento dal physically based rendering sono la *Radiometria* e *Radiative Transfer*. La *Radiometria* è il modello matematico per studiare e misurare la propagazione delle radiazioni elettromagnetiche. In tale modello, ci disinteressiamo degli aspetti ondulatori delle onde elettromagnetiche e guardiamo la radiazione come energia fluente nello spazio. Ciò suggerisce che la Radiometria opera a livello dell'ottica geometrica, assumendo dunque propagazione rettilinea della luce e trascurando fenomeni dovuti alla natura ondulatoria della radiazione. Si considerano solamente gli effetti della riflessione, trasmissione, scattering ed assorbimento. In particolare, le assunzioni fatte sono le seguenti[[PJH23]]:

Linearità:	La combinazione lineare di più input ad un sistema ottico è pari alla combinazione lineare degli effetti dei singoli input
Conservazione dell'energia:	Quando una fonte di radiazioni scatters da una superficie o mezzo trasmissivo, gli <i>scattering events</i> [[Peg16]], non possono produrre più energia di quella iniziale
No polarizzazione:	Ignoriamo la polarizzazione del campo elettromagnetico di una radiazione, rendendo l'unica proprietà rilevante di una radiazione la sua <i>Distribuzione Spettrale</i> [Radiazione di Corpo Nero]
No fluorescenza o fosforescenza:	Il comportamento della radiazione ad una lunghezza d'onda è indipendente dal comportamento della radiazione ad altre lunghezze d'onda
A regime:	Il <i>Light Field</i> (Radiometria) nell'ambiente è assunto a regime, indipendente dal tempo. Nota che la fosforescenza viola anche questo presupposto

Il *Radiative Transfer*, qui solo accennato, è lo studio dei fenomeni legati al trasferimento di energia radiante, basato sulla radiometria, ed utilizzato nella computer grafica per introdurre fenomeni legati all'ottica fisica e modellare la propagazione in un mezzo diverso dal vuoto.

Le unità fondamentali della radiometria sono riassunte nella tabella 4.1

Quantità	Simbolo	unità S.I.	Note
Energia Radiante	$Q_e$	J	Energia aggregata di tutti i fotoni in una radiazione elettromagnetica
Flusso Radiante	$\Phi_e$	W	Energia Radiante <i>emessa, riflessa, trasmessa, incidente</i> , per unità di tempo
Intensità Radiante	$I_{e,\Omega}$	W/sr	Flusso Radiante emesso, riflesso, trasmesso, per unia di angolo solido
Irradianza	$E_e$	W/m <sup>2</sup>	Flusso Radiante incidente per unità di area
Emittanza Radiante	$M_e$	W/m <sup>2</sup>	Flusso Radiante emesso per unità di area
Radiosità	$J_e$	W/m <sup>2</sup>	Flusso Radiante <i>uscente</i> (emesso, riflesso e trasmesso) per unità di area
Radianza	$L_{e,\Omega}$	W/(sr · m <sup>2</sup> )	Flusso Radiante emesso, riflesso, trasmesso, incidente da/su una superficie, per unità di angolo solido per unità di <i>area proiettata</i>

Table 4.1: nomenclatura e misure delle quantità radiometriche per noi rilevanti

**Def. 4.1.1.** — L'*Energia Radiante*  $Q_e$  è l'energia aggregata di tutti i fotoni, ognuno ad una particolare lunghezza d'onda con contributo energetico diverso, *verso/attraverso/da* una superficie in un dato intervallo di tempo

Ciascuno di questi fotoni trasporta energia pari all'Equazione di plank-Einstein  $Q_\lambda = \frac{hc}{\lambda}$

**Def. 4.1.2.** — L'*Energia Radiante Spettrale*  $Q_{e,\lambda}$ <sup>1</sup> è il contributo di una singola lunghezza d'onda all'energia radiante complessiva

$$Q_{e,\lambda} = \frac{\partial Q_e}{\partial \lambda}$$

Data l'assunzione di sistema a regime, siamo interessati a misurare l'energia radiante in un istante piuttosto che per un periodo definito di tempo. Dunque definiamo

**Def. 4.1.3.** — Il *Flusso Radiante*  $\Phi_e$  è la quantità di energia radiante passante attraverso una superficie per unità di tempo

$$\Phi_e = \frac{\partial Q_e}{\partial t}$$

La cui derivata rispetto alle frequenze fornisce il *Flusso Radiante Spettrale*

$$\Phi_{e,\lambda} = \frac{\partial \Phi_e}{\partial \lambda}$$

Ogni misurazione di un flusso richiede un area ben definita per attribuirle senso. Nei sistemi di rendering, non è insolito invece compiere delle misure "puntuali" di un flusso radiante, cioè *flusso radiante per unità di area*.

**Def. 4.1.4.** — La densità superficiale del flusso radiante (energia per unità di tempo, per unità di superficie) è detta *Irradianza*  $E_e$  se tale flusso è entrante la superficie, *Emittanza Radiante*  $M_e$

<sup>1</sup>Le grandezze spettrali possono essere espresse in funzione della lunghezza d'onda  $\lambda$  ( e.g.  $Q_{e,\lambda}$  ) o del suo inverso, la frequenza ( e.g.  $Q_{e,\nu}$  )

se emesso dalla superficie, *Radiosità*  $J_e$  se uscente (emesso, riflesso o trasmesso) dalla superficie

$$E_e|J_e|M_e = \frac{\partial\Phi_e}{\partial A}$$

La cui derivata rispetto alle frequenze fornisce la rispettiva grandezza spettrale

$$E_{e,\lambda}|J_{e,\lambda}|M_{e,\lambda} = \frac{\partial E_e|J_e|M_e}{\partial\lambda}$$

Si noti infine che l'Irradianza, per definizione, è pari alla media temporale in un periodo del vettore di Pointing perpendicolare alla superficie

$$E_e = \langle |\vec{S}| \rangle$$

Piuttosto che concentrarci su un unico punto della superficie, possiamo concentrarci su una singola direzione, considerando il flusso radiante per unità di angolo solido.

**Def. 4.1.5.** — La *Intensità Radiante*  $I_{e,\Omega}$  è la densità angolare del flusso radiante emesso, riflesso, trasmesso o incidente da/verso un punto dello spazio e propagante lungo una direzione specificati

$$I_{e,\Omega} = \frac{\partial\Phi_e}{\partial\Omega}$$

L'*Intensità (Radiante) Spettrale* è l'intensità radiante per unità di lunghezza d'onda

$$I_{e,\Omega,\lambda} = \frac{\partial I_{e,\Omega}}{\partial\lambda}$$

Tale grandezza è utile per la modellazione, ad esempio, delle sorgenti luminose puntuali omnidirezionali, le quali distribuiscono il loro flusso radiante emesso, inversamente proporzionale al quadrato della distanza dalla sorgente, in modo uniforme il ogni direzione  $I_{e,\Omega} = \frac{\Phi_e}{4\pi}$ .

La descrizione più granulare dell'energia emessa da un'onda elettromagnetica è fornita dalla *Radianza*, definita come

**Def. 4.1.6.** — La *Radianza*  $L_{e,\Omega}$  è definita come flusso radiante emesso, riflesso, trasmesso o incidente da/su una dato punto della superficie, da/verso una data direzione, dunque per unità di angolo solido per unità di area *proiettata*

$$L_{e,\Omega} = \frac{\partial^2\Phi_e}{\partial\Omega\partial(A\cos\theta)}$$

dove  $\theta$  angolo tra la direzione perpendicolare alla superficie sul punto considerato e la direzione del flusso

Significativa ai nostri fini è la relazione inversa che sussiste tra Irradianza e Radianza, che sfrutteremo per definire l'equazione del rendering

$$E_e(\vec{p}) = \int_{\Omega} L_{e,\Omega}(\vec{p}, \hat{\omega}) \cos\theta d\hat{\omega}^2 = \int_{\theta_1}^{\theta_2} \int_{\varphi_1}^{\varphi_2} L_{e,\Omega}(\vec{p}, \theta, \varphi) \cos\theta \sin\theta d\theta d\varphi \quad (4.1)$$

<sup>2</sup>Si noti la distinzione tra  $\hat{\omega}$  e  $d\hat{\omega}$ . Il primo è un vettore unitario dal punto  $\vec{p}$  ad un punto della porzione di sfera unitaria costituente il dominio di integrazione, il secondo è un elemento di angolo angolo solido. Un angolo solido è un numero adimensionale rappresentante la porzione di area di sfera coperta, nell'intervallo  $[0, 4\pi]$ . Il suo differenziale rappresenta un piccolo quadratino della superficie della sfera, spesso approssimato ad un quadrato ai fini di dimostrazioni geometriche

### 4.1.1 Radianza incidente ed uscente

La radianza  $L$  è generalmente una funzione discontinua sulle superfici di separazione tra due mezzi, dette interfacce.

$$L^+ = \lim_{t \rightarrow 0^+} L(\vec{p} + t\hat{n}, \hat{\omega}) \quad (4.2)$$

$$L^- = \lim_{t \rightarrow 0^-} L(\vec{p} + t\hat{n}, \hat{\omega}) \quad (4.3)$$

dunque, piuttosto che lavorare con un'unica funzione discontinua, si preferisce distinguere tra *radianza incidente*  $L_i$  e *radianza uscente*<sup>3</sup>.

$$L_i(\vec{p}, \hat{\omega}) = \begin{cases} L^+(\vec{p}, -\hat{\omega}), & \langle \omega, \hat{n} \rangle > 0 \\ L^-(\vec{p}, -\hat{\omega}), & \langle \omega, \hat{n} \rangle < 0 \end{cases} \quad (4.4)$$

$$L_o(\vec{p}, \hat{\omega}) = \begin{cases} L^+(\vec{p}, \hat{\omega}), & \langle \omega, \hat{n} \rangle > 0 \\ L^-(\vec{p}, \hat{\omega}), & \langle \omega, \hat{n} \rangle < 0 \end{cases} \quad (4.5)$$

Si noti che in punti dello spazio non giacenti su una superficie, la radianza è invece continua,  $L^+ = L^-$ , da cui

$$L_o(\vec{p}, \hat{\omega}) = L_i(\vec{p}, -\hat{\omega}) \quad (4.6)$$

Si può riscrivere la conservazione della radiazione di base in termini della radianza incidente e radianza uscente

$$\frac{L_o(\vec{p}, \hat{\omega})}{\eta_1^2} = \frac{L_i(\vec{q}, \hat{\omega})}{\eta_2^2} \quad (4.7)$$

## 4.2 Radiazione di Corpo Nero

Tutti i corpi con temperatura superiore allo zero assoluto emettono radiazione, in quantità proporzionale alla temperatura assoluta in quanto essa corrisponde allo stato di agitazione delle molecole, causando accelerazione di elettroni e protoni degli atomi costituenti, portando all'emissione di campo elettromagnetico secondo le equazioni di Maxwell. Tale flusso radiante spettrale emesso costituisce una *Spectral Power Distribution* (SPD).

Un *Corpo Nero* corpo ideale che assorbe ogni tipo di radiazione incidente, indipendentemente da frequenza o angolo di incidenza. In equilibrio termico, esso emette radiazioni che seguono la *Legge di Planck*. Esso è un *Radiatore Ideale e diffusore*, cioè emette radiazioni uniformemente in tutte le direzioni (superficie lambertiana).

per tali corpi neri, l'Emittanza radiante  $M_{bb}$  dalla superficie obbedisce alla *Legge di Stefan-Boltzmann*

$$M_{bb} = \sigma T^4 \quad (4.8)$$

dove  $\sigma = 5.67032 \cdot 10^{-8} \text{ W}/(\text{m}^2 \text{K}^4)$  costante di Stefan-Boltzmann

Dunque tale corpo nero è una superficie opaca (trasmittanza nulla), perfettamente assorbente (absorptance unitaria) che non riflette alcuna radiazione incidente (riflettanza nulla). Ciò vuol dire che emette tutta la radiazione che assorbe ( $A + R + T = 1$ ).

<sup>3</sup>emessa, riflessa, trasmessa

<sup>4</sup>Si noti come per convenzione, la radianza incidente la direzione  $\hat{\omega}$  sia uscente dal punto *vecp*, nonostante il flusso radiante abbia direzione opposta. In altre parole, la radianza incidente ha come argomento la direzione di provenienza del flusso

## 4.2. RADIAZIONE DI CORPO NERO

---

Essendo una superficie lambertiana, anche la radianza emessa deve essere costante. Sapendo che un emisfera unitaria ha integrale pari a  $\pi$ ,

$$M_{bb\lambda} = \pi L_{bb\lambda} \quad (4.9)$$

In equilibrio termico, tale radianza ha espressione detta *Legge di Planck*

$$L_{bb\lambda}(T, \lambda) = \frac{2hc^2}{\lambda^5 \left( e^{\frac{hc}{\lambda kT}} - 1 \right)} \quad (4.10)$$

dove

$$h = 6.626176 \cdot 10^{-34} \text{ J/s costante di Planck} \quad (4.11)$$

$$k = 1.380662 \cdot 10^{-23} \text{ J/K costante di Boltzmann} \quad (4.12)$$

È facile dimostrare come, integrando nell'emisfera unitaria, si ottiene un'emitanza radiante corrispondente alla legge di Stefan-Boltzmann

Tali corpi neri si dimostrano utili non solo come modelli per parametrizzare, tramite temperatura, una sorgente luminosa, ma risultano rilevanti anche per la colorimetria, in quanto costituenti il luogo plankiano.

# Chapter 5

## Fotometria, Colore e Camera

### 5.1 Fotometria

L'occhio umano non è equamente sensibile a tutte le lunghezza d'onda della luce visibile, la quale è un sottoinsieme dello spettro elettromagnetico corrispondente all'intervallo  $[360, 830] \text{ nm}$ , come accennato nel primo capitolo II. Dunque, si introduce

**Def. 5.1.1.** — La *Fotometria* è la scienza che studia la misurazione della luce in termini della intensità percepita dall'occhio umano.

Tale definizione permette di capire che la fotometria si distingue dalla radiometria in quanto, data una sorgente luminosa  $\Sigma$  con una certa SPD, lo scopo è pesarne il contributo di ciascuna lunghezza d'onda secondo la percezione umana.

L'occhio umano possiede due tipi di fotorecettori: *coni* e *bastoncini*

I *coni* sono responsabili per la percezione in ambienti illuminati, detta *visione fotopica*

I *bastoncini* sono responsabili per la percezione in ambienti bui, detta *visione scotopica*

A seguito di esperimenti condotti nel 1931 dalla CIE, è stato associato un modello di risposta spettrale standard, e sono chiamate *Efficacia Luminosa Spettrale Fotopica*  $V_p(\lambda)$  e *Efficacia Luminosa Spettrale Scotopica*  $V_s(\lambda)$ .

Tali funzioni, in ciascuna lunghezza d'onda, rappresentano un peso  $\in [0, 1]$ , il quale può essere integrato con una grandezza radiometrica spettrale per ottenere l'analogia grandezza fotometrica.<sup>1</sup>

$$\text{Energia Luminosa } Q_v = K \int_{\Lambda} Q_{e,\lambda}(\lambda) V(\lambda) d\lambda \quad (5.1)$$

$$\text{Flusso Luminoso } \Phi_v = K \int_{\Lambda} \Phi_{e,\lambda}(t, \lambda) V(\lambda) d\lambda \quad (5.2)$$

$$\text{Illuminanza} | \text{Emittanza Luminosa } E_v | M_v = K \int_{\Lambda} E_{e,\lambda} | M_{e,\lambda}(\vec{p}, \lambda) V(\lambda) d\lambda \quad (5.3)$$

$$\text{Intensità Luminosa } I_v = K \int_{\Lambda} I_{e,\Omega,\lambda}(\hat{\omega}, \lambda) V(\lambda) d\lambda \quad (5.4)$$

$$\text{Luminanza}^2 Y = K \int_{\Lambda} L_{e,\Omega,\lambda}(\vec{p}, \hat{\omega}, \lambda) V(\lambda) d\lambda \quad (5.5)$$

---

<sup>1</sup>Si noti che tutte le formule viste finora per le grandezze radiometriche, come la relazione tra radianza e etendue ??, o le relazioni tra grandezze radiometriche

dove  $\Lambda = [380, 830]\text{nm}$ , cioè intervallo nel quale le funzioni  $V(\lambda) \neq 0$

Nota che non si è indicato quale funzione di efficacia luminosa si sta utilizzando, in quanto le formule sono universalmente valide. Le costanti, invece, che rappresentano l'*Efficacia Luminosa di una radiazione* cambiano valore:

$$K = 683.002 \text{ lm/W Efficacia luminosa fotopica}$$

$$K \approx 1700 \text{ lm/W Efficacia luminosa scotopica}$$

Tali costanti rappresentano l'efficacia luminosa, cioè il fattore di conversione  $\text{W} \rightarrow \text{lm}$  per una radiazione monocromatica a lunghezza d'onda 555nm.

### 5.1.1 fotometria e sistemi di rendering

La fotometria, nel contesto di Physically based Rendering, è utile per la modellazione di sorgenti luminose, per specificare parametri come intensità luminosa, light falloff, e così via.

Includendo parametri fotometrici si può ottenere una descrizione più vicina all'uomo della percezione della luce mantenendone il rigore fisico.

## 5.2 Colorimetria

Il colore è un fenomeno la cui origine è sia fisica, determinato dalla distribuzione nelle lunghezze d'onda della radianza spettrale, sia psicobiologica, in quanto la sua percezione è artefatto dell'interpretazione di tali lunghezze d'onda nella corteccia cerebrale posteriore.

Ciò suggerisce che il colore *non è una proprietà degli oggetti*, ma della luce da essi riflessa e dunque dipendente alle lunghezze d'onda contenute nella radiazione incidente.

Nonostante specificare un colore sottoforma di spettro di frequenza del flusso radiante sia la modalità più fedele e ottimale, essa

- Non è una interfaccia human-friendly e orientata alla percezione
- Non corrisponde alla rappresentazione utilizzata dai display odierni

Dunque, nel fornire un'interfaccia per un sistema di rendering, bisogna implementare la possibilità di specificare colore in modo intuitivo, convertire tale specifica in uno spettro di frequenze per effettuare la computazione, e convertire nella rappresentazione nativa per mostrare a schermo i risultati.

**Def. 5.2.1.** — Risulta dunque conveniente specificare delle *funzioni di base* per poter costituire uno spazio vettoriale tramite il quale, specificandone i componenti, si specifica un determinato colore. Tale spazio vettoriale è chiamato *Color Space*.

Un *color stimulus* può essere ricostruito dalla combinazione lineare da tre stimoli (distribuzioni spettrali) di base indipendenti [Sch07], in quanto dei fotorecettori umani che distinguono il colore, i coni, ce ne sono di *tre tipi* (vedi figura 5.1)a. Tale mappa da SPD a *tristimulus values* non è biunivoca, infatti, nei diversi color spaces che definiamo, esistono valori tristimolo che possono risultare dalla scomposizione di diverse SPDs. Tali colori sono detti *metameri*.

Dunque l'obiettivo è quello di ottenere distribuzioni spettrali  $\langle R \rangle, \langle G \rangle, \langle B \rangle$  tali che ogni colore possa essere rappresentato come

$$C = R\langle R \rangle + G\langle G \rangle + B\langle B \rangle \quad (5.6)$$

tale compito non è scontato in quanto, da come si osserva in figura 5.1b, quasi tutte le lunghezze d'onda stimolano con diversa intensità almeno due tipi di coni. Dunque, nel 1931,

<sup>2</sup>Nota come piuttosto che seguire la nomenclatura ISO per la Luminanza  $L_v$ , si è scelto di adottare  $Y$ . I motivi appariranno chiari nei cenni sulla colorimetria 5.2

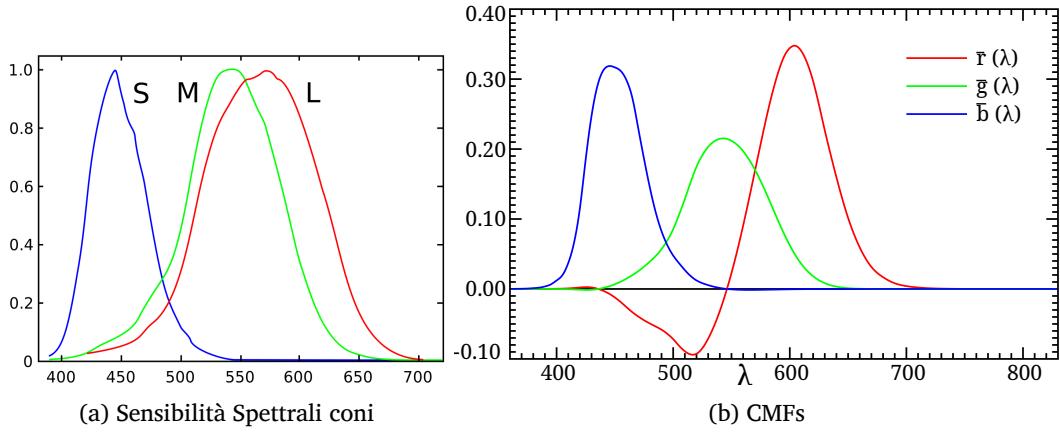


Figure 5.1: A sinistra, Sensibilità spettrali per i tre tipi di coni dell'occhio umano. A destra, color matching functions del CIE 1931 RGB Color Space

fu derivato il CIE 1931 RGB Trichromatic system, cioè una terna di color matching functions  $\bar{r}(\lambda)$ ,  $\bar{g}(\lambda)$ ,  $\bar{b}(\lambda)$ , derivate compiendo dei test con un *osservatore standard colorimetrico*, cioè limitando il field of view dei soggetti testati a  $2^\circ$  dentro la fovea, per eliminare la variabilità di percezione causata dal field of view dell'osservatore. Tali soggetti sono stati esposti a luci monocromatiche, i.e. i colori primari scelti, alle frequenze 700 nm (rosso), 546.1 nm (verde), 435.8 nm (blu). L'intensità di queste tre sorgenti primarie erano tali che, se addizionate, restituiscono uno spettro costante la cui intensità è pari alla Luminanza della sorgente complessiva

$$R = \int_{\Lambda} \bar{r}(\lambda) L_{e,\Omega,\lambda}(\lambda) d\lambda \quad (5.7)$$

$$G = \int_{\Lambda} \bar{g}(\lambda) L_{e,\Omega,\lambda}(\lambda) d\lambda \quad (5.8)$$

$$B = \int_{\Lambda} \bar{b}(\lambda) L_{e,\Omega,\lambda}(\lambda) d\lambda \quad (5.9)$$

$$L_v = 1.0000R + 4.5907G + 0.0601B \text{ [cd/m}^2\text{]} \quad (5.10)$$

Le tre color matching functions (CMFs) così ottenute sono mostrate in figura 5.1b.

Ai tempi della standardizzazione di un sistema tricromatico, in assenza di computers, risultava complicato fare calcoli con queste tre CMFs per via dei lobi negativi. Dunque, sempre nel 1931, fu standardizzato il *CIE 1931 XYZ Color System*, nel quale sono stati scelti tre *primari immaginari*<sup>3</sup> a partire dai primari del sistema CIE RGB 1931 in modo tale che

- uno spettro costante dia luogo a componenti  $X = Y = Z$
- la componente  $Y$  sia la luminanza della radiazione  $Y = L_v$
- la CMF  $\bar{y}$  sia uguale alla funzione di efficacia luminosa fotopica  $\bar{y}(\lambda) = V_p(\lambda)$
- $Z$  è quasi uguale al blu di CIE RGB
- $X$  è un mix delle tre CMFs di CIE RGB

Dai requisiti la seguente trasformazione lineare CIE RGB 1931 → CIE XYZ 1931 è stata rica-

<sup>3</sup>non percepibili all'occhio umano

vata

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 2.768892 & 1.751748 & 1.130160 \\ 1.000000 & 4.590700 & 0.060100 \\ 0 & 0.056508 & 5.594292 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (5.11)$$

La quale è anche valida per trasformare una CMF del sistema CIE RGB in una CMF del sistema CIE XYZ 1931.

Per passare da radianza spettrale ai tristimulus values, distinguiamo il caso in cui si sta analizzando una radiazione/superficie emissiva

$$X = K \int_{\Lambda} L_{e,\Omega,\lambda}(\lambda) \bar{x}(\lambda) d\lambda \quad (5.12)$$

$$Y = K \int_{\Lambda} L_{e,\Omega,\lambda}(\lambda) \bar{y}(\lambda) d\lambda \quad (5.13)$$

$$Z = K \int_{\Lambda} L_{e,\Omega,\lambda}(\lambda) \bar{z}(\lambda) d\lambda \quad (5.14)$$

Ed il caso in cui si sta analizzando un colore di una radiazione riflessa o trasmessa (fonti luminose secondarie)

$$X = \frac{1}{\int_{\Lambda} L_{e,\Omega,\lambda}(\lambda) \bar{y}(\lambda) d\lambda} \int_{\Lambda} S(\lambda) L_{e,\Omega,\lambda}(\lambda) \bar{x}(\lambda) d\lambda \quad (5.15)$$

$$Y = \frac{1}{\int_{\Lambda} L_{e,\Omega,\lambda}(\lambda) \bar{y}(\lambda) d\lambda} \int_{\Lambda} S(\lambda) L_{e,\Omega,\lambda}(\lambda) \bar{y}(\lambda) d\lambda \quad (5.16)$$

$$Z = \frac{1}{\int_{\Lambda} L_{e,\Omega,\lambda}(\lambda) \bar{y}(\lambda) d\lambda} \int_{\Lambda} S(\lambda) L_{e,\Omega,\lambda}(\lambda) \bar{z}(\lambda) d\lambda \quad (5.17)$$

$$(5.18)$$

dove

$$S(\lambda) = \begin{cases} R(\lambda) & \text{se riflessione} \\ T(\lambda) & \text{se trasmissione} \end{cases} \quad (5.19)$$

Si noti che le Color Matching Functions  $\bar{x}(\lambda)$ ,  $\bar{y}(\lambda)$ ,  $\bar{z}(\lambda)$  sono tabulate ad intervalli di 1 nm, rendendo possibile l'applicazione del metodo di Monte Carlo come indicato in Equazione 1.9. Spesso si preferisce normalizzare i valori tristimolo per ottenere le *coordinate di cromaticità*

$$x = \frac{X}{X + Y + Z} \quad (5.20)$$

$$y = \frac{Y}{X + Y + Z} \quad (5.21)$$

$$z = \frac{Z}{X + Y + Z} \quad (5.22)$$

dove, in quanto  $x + y + z = 1$ , soltanto le coordinate  $xy$  sono necessarie per una completa descrizione del colore.

Il diagramma di cromaticità è riportato in figura 5.2. Il contorno curvo rappresenta l'insieme di punti nel diagramma corrispondente alle radiazioni monocromatiche, mentre la curva evidenziata rappresenta il *Luogo Planckiano*, insieme di punti corrispondenti ad uno spettro di emissione di un corpo nero plankiano (formula 4.10), con temperatura da  $[1000, \infty]$  K.

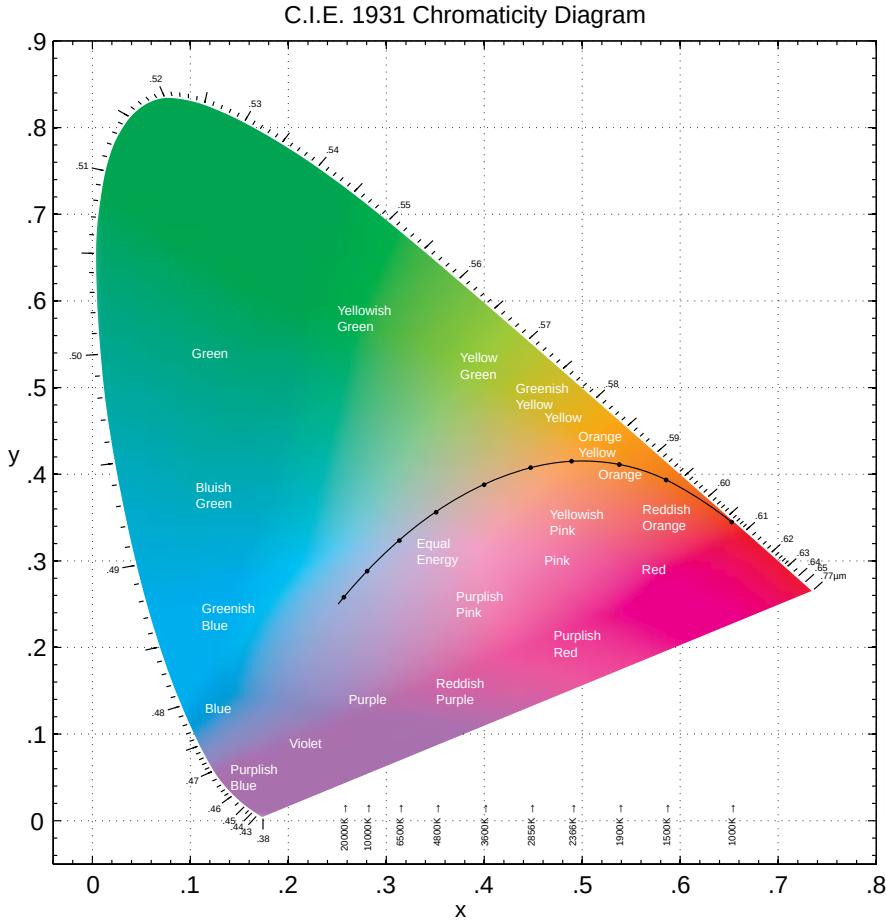


Figure 5.2: xy Chromaticity Diagram del sistema tricromatico CIE XYZ 1931

### 5.2.1 sRGB color space

Mentre per memorizzare image data un qualsiasi color space può essere utilizzato (infatti, si possono memorizzare anche dati relativi allo spettro direttamente [al21]), per mostrare colore a schermo si preferisce scegliere 3 primari ed un punto bianco, punto nel quale i 3 primari danno contributo massimo. La scelta dei primari è guidata dalla percentuale di gamut che si desidera coprire con tutte le combinazioni lineari dei tre primari scelti e limitazioni fisiche. Lo standard adottato per monitors e World Wide Web nel 1996 da IEC è *sRGB*. Esso è un color space additivo basato sui tre primari rosso, verde, blu, e white point, le cui coordinate di cromaticità sono riportate in Tabella 5.1.

Altro componente per la specifica di un color space per un display è la funzione di trasferimento del display, in particolare

**Def. 5.2.2.** — La *electro-optical transfer function* (EOTF) è un funzione di trasferimento che converte un segnale immagine in input in intensità luminosa in output. Essa non è lineare

Tale funzione di trasferimento non è lineare per l'operazione di *gamma correction* compiuta dai displays, la quale include sempre un elevamento a potenza  $v_{in} = Av_{out}^\gamma$  (di solito  $\gamma = 2.2$ ).

Cromaticità	Rosso	Verde	Blu	White point(D65)
$x$	0.64	0.30	0.15	0.3127
$y$	0.33	0.60	0.06	0.329
$Y$	0.2126	0.7152	0.0722	1

Table 5.1: Coordinate degli stimoli primari e del white point del sRGB color space

Tale operazione è compiuta per ottimizzare l'uso dei bits nella codifica dell'immagine, dando più importanza ai toni più scuri, in accordo con la percezione umana in grado di apprezzarli con più sensibilità.

Dunque se l'output finale del sistema di rendering è un colore sRGB, dobbiamo essere capaci di convertire a/da un colore in sRGB color space da/a un colore in XYZ color space. Sia  $C_{srgb} = R_{srgb}|G_{srgb}|B_{srgb} \in [0, 1]$  componenti del colore nello spazio sRGB gamma encoded. Simile definizione per il colore nello spazio sRGB gamma corrected  $C_{linear}$  ed il colore  $[X, Y, Z]^T$  nello spazio XYZ.

$$\text{Applica gamma expansion}^4: C_{linear} = \begin{cases} \frac{C_{srgb}}{12.92}, & C_{srgb} \leq 0.04045 \\ \left( \frac{C_{srgb} + 0.055}{1.055} \right)^{2.4}, & C_{srgb} > 0.04045 \end{cases} \quad (5.23)$$

$$\text{Applica trasformazione lineare: } \begin{bmatrix} X_{D65} \\ Y_{D65} \\ Z_{D65} \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{linear} \\ G_{linear} \\ B_{linear} \end{bmatrix} \quad (5.24)$$

I valori tristimolo  $X_{D65}, Y_{D65}, Z_{D65}$  sono infine scalati in modo tale che l'illuminante standard D65, utilizzato come bianco di riferimento, abbia luminanza unitaria, cioè moltiplicate per  $\approx 3.039513678$ , ottenendo  $X, Y, Z$ .

### 5.2.2 Conversione da XYZ o RGB a SPD

I color spaces sono spazi vettoriali, per i quali sono definite le operazioni di addizione tra colori (vettori) e moltiplicazione scalare colore. *La moltiplicazione tra più colori non è un'operazione ben definita*. Nel trasporto della luce, la moltiplicazione tra due SPD è una operazione fondamentale. Si potrebbe pensare di definire arbitrariamente una operazione analoga in uno spazio RGB qualsiasi e definire la moltiplicazione tra due colori come *prodotto di Hadamard*  $\odot$ .

Tale operazione risulta problematica perché essa non è consistente tra color spaces, producendo risultati differenti (a volte al di fuori del color gamut, producendo un colore impossibile, si veda in figura 5.3 ProPhoto RGB), ma più importante, produce colori eccessivamente più scuri e saturati.

Si giustifica dunque l'affermazione precedente di dover interpretare i colori in input negli arbitrari formati/color spaces supportati in uno spettro, eseguire la computazione, ed infine convertire il risultato per ogni pixel in sRGB per display. Mentre la conversione SPD  $\rightarrow$  Color Space è ben definita (vedi 5.19 e 5.12), non vale lo stesso per l'operazione inversa ed è ad oggi oggetto di ricerca [al22]. Ciò che complica tale compito sono i requisiti di

<sup>4</sup>la semplice funzione potenza viene modificata secondo standards, come ITU-R per poter evitare problematiche. Per esempio, per evitare di avere derivata infinita nello zero, si definisce la funzione di gamma expansion per valori piccoli come divisione per una costante definita dallo standard

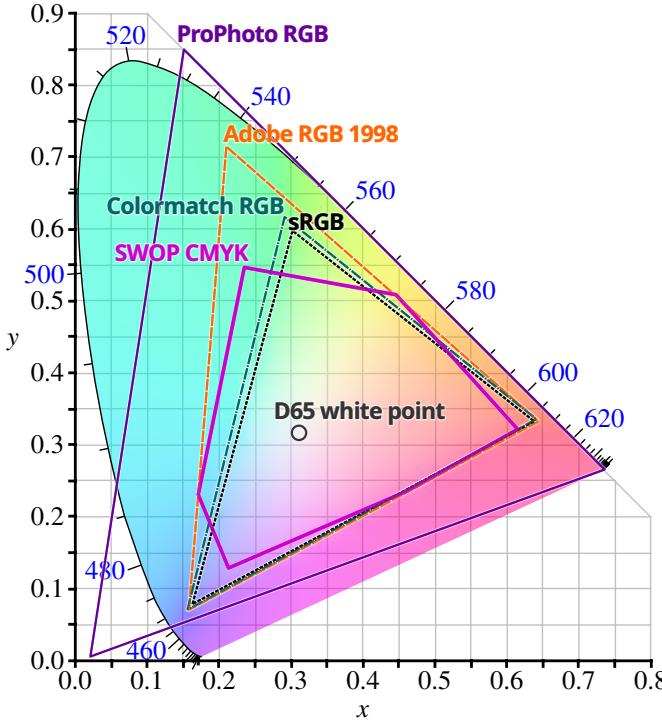


Figure 5.3: comparazione tra i diversi colorspace, ed i colori percepibili che riescono a rappresentare. Notiamo che sRGB copre all'incirca 35% del color gamut

*Identità* La conversione da RGB a spettro, seguita dalla ben definita operazione inversa, deve restituire lo stesso risultato

*Smoothness* Lo spettro ottenuto deve essere derivabile con continuità affinchè, sotto nessune condizioni di luce, uno spettro di riflessione presenti seams visibili

*Conservazione dell'energia* Lo spettro ottenuto deve avere integrale  $\in [0, 1]$  se  $[R, G, B]^T \in [0, 1]^3$

L'approccio qui seguito è quello proposto da [PJH23] e [Wen19], il quale consiste nel ricostruire un valore RGB utilizzando una parabola parametrizzata da tre coefficienti  $c_0, c_1, c_2$ , la quale è prima resa limitata da una funzione sigmoide per poter ottenere un risultato limitato in  $[0, 1]$ . Vedi figura 5.4.

$$S(\lambda) = s(c_0\lambda^2 + c_1\lambda + c_2) \quad (5.25)$$

$$s(x) = \frac{1}{2} + \frac{x}{2\sqrt{1+x^2}} \quad (5.26)$$

Tali coefficienti sono calcolati a partire da un colore nell'sRGB color space, il quale è formulato come problema di ottimizzazione con scopo la minimizzazione della metrica di residuo seguente

$$\begin{bmatrix} c_0^* \\ c_1^* \\ c_2^* \end{bmatrix} = \arg \min_{c_0, c_1, c_2} \left\| \begin{bmatrix} r \\ g \\ b \end{bmatrix} - \int_{\Lambda} \begin{bmatrix} R(\lambda) \\ G(\lambda) \\ B(\lambda) \end{bmatrix} S(\lambda, c_0, c_1, c_2) W_{D_{65}}(\lambda) d\lambda \right\| \quad (5.27)$$

dove  $\|\cdot\|$  è la CIE76  $\Delta E$  color distance. La completa dimostrazione e dettagli implementativi sono rimandati a [PJH23]

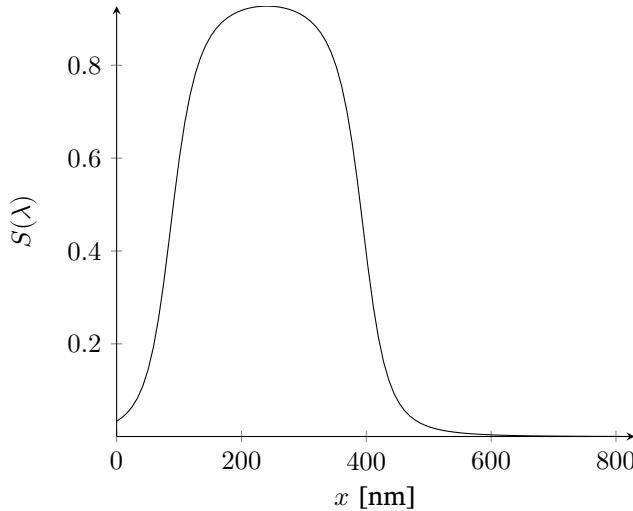


Figure 5.4: esempio di  $S(\lambda)$  con  $c_0 = -0.000073447$ ,  $c_1 = 0.035354$ ,  $c_2 = -2.607$

## 5.3 Camera

Affinchè si possa associare ad ogni punto 2D nell'*color buffer* che conterrà eventualmente l'immagine output uno o più coppie  $(\vec{p}_0, \hat{\omega}_0)$  punto e direzione di partenza, necessitiamo di un *camera model*

**Def. 5.3.1.** — Un *camera model* descrive la relazione matematica tra le coordinate 3D di un punto nella scena dal quale la luce proviene e le coordinate 2D della sua proiezione nel *film plane*<sup>5</sup> [Ike14]

### 5.3.1 pinhole camera model

Modello di camera ideale secondo la quale la scena è osservata attraverso una apertura infinitesima. Tale modello genera immagini con messa a fuoco nitida ovunque, trascurando tutte le distorsioni geometriche e effetti di aberrazione tipici delle lenti finite.<sup>6</sup>

Considerazioni Geometriche:

Origine:	punto $\vec{o}$ , nel quale è posizionata l'apertura della camera
Film plane:	posizionato a distanza $n$ (f in figura), <i>distanza focale</i> della camera, dall'origine $\vec{o}$ . Definisce il <i>field of view</i> . Anche chiamato <i>image plane</i> o <i>near plane</i>
Punti:	$\vec{p}$ e $\vec{q}$ , rispettivamente punto da proiettare, "trasportante rianza", e punto $\vec{q}$ , proiezione del punto $\vec{p}$ sul film plane

Definiamo inoltre dei Coordinate Spaces:

Object Space: spazio tridimensionale in cui è definito ciascun oggetto

<sup>5</sup>film è il termine che utilizziamo per il piano dell'immagine finale [PJH23]

<sup>6</sup>descrivere ed implementare modelli di camera realistici è fuori dallo scope

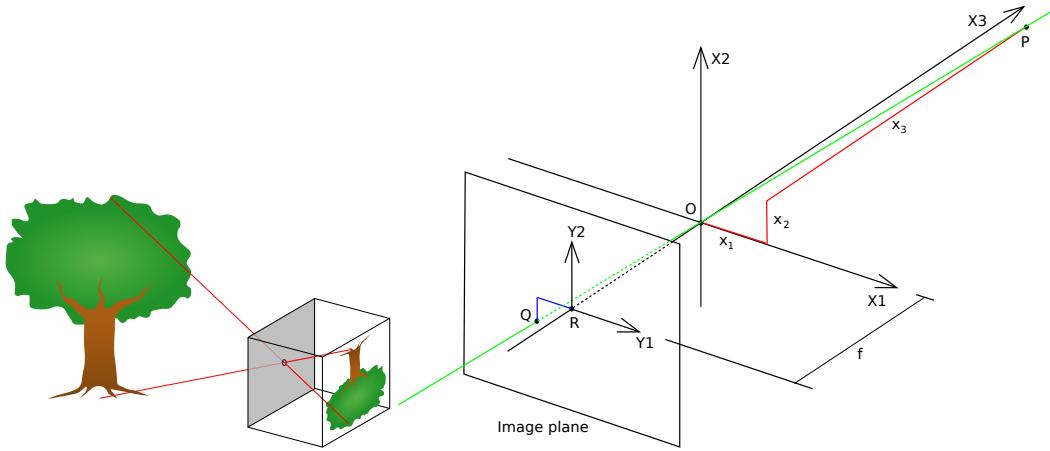


Figure 5.5: A sinistra, diagramma del pinhole camera model. A destra, geometria del pinhole camera model con un sistema di riferimento left-handed. Di solito l'asse z punta verso la camera, risiedente in  $z=0$

World Space:

spazio tridimensionale utilizzato dalla scena

Camera Space:

spazio tridimensionale avente come origine il punto di apertura della camera, asse x e y direzioni parallele al film plane, e z direzione perpendicolare al film plane

Rendering Space:

spazio tridimensionale avente come origine il punto di apertura della camera, ma preservante le direzioni del world space. Lo utilizziamo negli algoritmi basati su ray tracing in quanto, fa in modo che gli oggetti vicini alla camera abbiano posizioni rappresentati da numeri piccoli, dunque sfruttanti precisione floating point maggiore, mentre le direzioni preservate permettono un migliore test di intersezione mediante *Axis Aligned Bounding Boxes*

Image Space:

spazio bidimensionale corrispondente a coordinate in  $[-1, 1]$  nel film plane, con origine il centro, e assi x, y paralleli alle direzioni x,y del Camera Space

Le formule che seguono utilizzano, per comodità, il Camera Space.

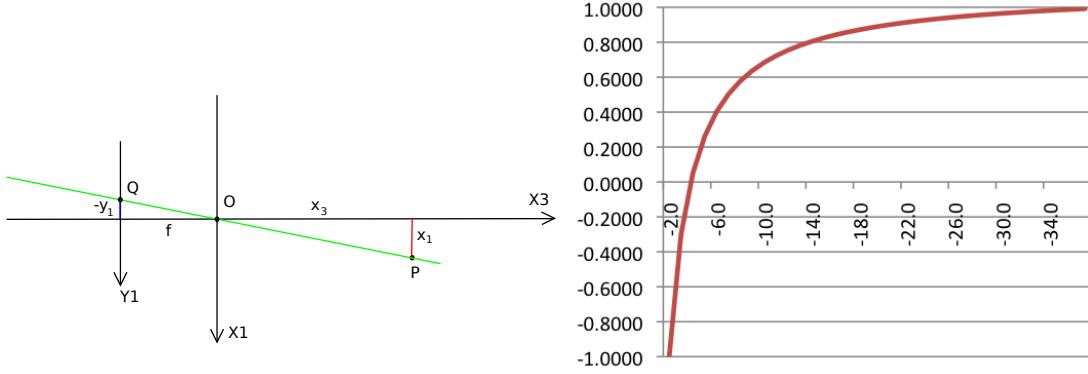
La mappa  $\vec{q} = [y_1, y_2]^T$  da  $\vec{p} = [x_1, x_2, x_3]^T$ . Siano  $l, r, b, t, n, f$  rispettivamente left, right, bottom, top, near, far.

Al fine di poter rappresentare una immagine finale che restituisca la percezione di uno spazio tridimensionale, ciascun punto della scena intersecato dal raggio visivo del campione considerato viene prima proiettato in un punto sul film plane.

Possiamo ricavare una *proiezione prospettica*<sup>8</sup> esprimendo il punto  $\vec{p}$  in coordinate omogenee e concatenando una trasformazione di *shearing*<sup>9</sup> per sovrapporre far plane  $[l, r] \times [b, t]$  con il near plane, seguito da una scala affinchè i due piani sovrapposti abbiano dimensione tale da

<sup>7</sup>Immagine da [http://learnwebgl.brown37.net/08\\_projections/projections\\_perspective.html](http://learnwebgl.brown37.net/08_projections/projections_perspective.html)

<sup>8</sup>Weak perspective projection



(a) Geometria del pinhole camera model visto dall'asse x

(b) Mostra la mappa nonlineare della depth<sup>7</sup>

Figure 5.6: proiezione prospettica su pinhole camera model

ottenere un angolo di visione 90°.

$$\begin{bmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\frac{r+l}{2n} & 0 \\ 0 & 1 & -\frac{t+b}{2n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{t-b} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.28)$$

che mappa

$$\begin{aligned} [l, b, n, 1]^T &\mapsto [-n, -n, n, 1]^T \\ [r, f, n, 1]^T &\mapsto [n, n, n, 1]^T \end{aligned}$$

Dunque il prossimo step è la normalizzazione delle tre coordinate spaziali, cioè la divisione di tutte le tre coordinate per la coordinata \$z\$, affinché più lontano sia un oggetto e più piccolo esso appare (step detto *perspective divide*) il che è ottenuto ponendo un coefficiente non nullo nell'elemento di indice (4, 3) nella matrice di trasformazione. A seguire, centriamo la media armonica  $\frac{2fn}{f+n}$  nell'origine e normalizziamo l'intervallo delle distanze, in modo tale che la mappa  $[-n, f] \mapsto [-1, 1]$  sia nonlineare, dando più importanza ai valori vicini alla camera. Vedi figura 5.6 a

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{f+n}{2} & -fn \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.29)$$

Combinando 5.29 e 5.28 si ottiene una trasformazione che mappa il view frustum, piramide retta a base rettangolare con dimensioni arbitrarie, ad un cubo  $[-1, 1]^3$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{t-b} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & -\frac{r+l}{t-b} & 0 \\ 0 & \frac{2n}{t-b} & -\frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.30)$$

<sup>9</sup>[https://en.wikipedia.org/wiki/Shear\\_mapping](https://en.wikipedia.org/wiki/Shear_mapping)

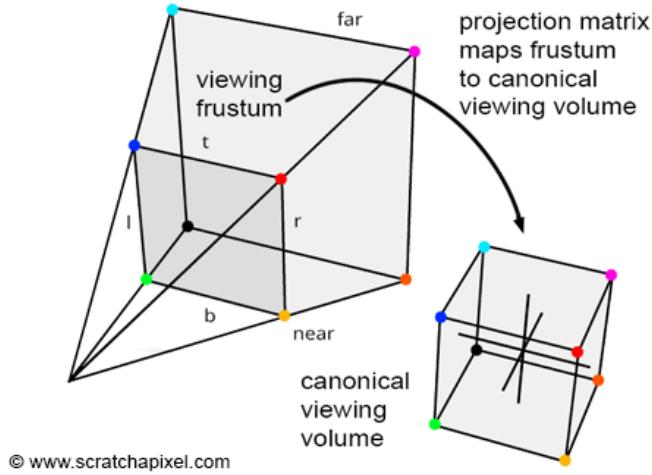


Figure 5.7: Illustrazione della mappa da view frustum a cubo unitario. Immagine da [www.scratchapixel.com](http://www.scratchapixel.com)

### 5.3.2 filtrare image samples

Dopo il campionamento di un pixel, e il calcolo della radianza spettrale in ogni pixel sample, bisogna aggregare tutti i contributi di ogni campione. Tutti i contributi dovrebbero avere un peso inversamente proporzionale alla distanza dal centro. Ciò suggerisce l'uso di un *filtro passa basso*  $f(x - x', y - y')$ , dove  $(x, y)$  coordinate in cui si calcola la *filtered image function*  $r_f(x, y)$  e  $(x', y')$  displacement.

Idealmente, per ottenere la migliore stima possibile, si dovrebbero estrarre  $\infty$  samples ed integrarli nella pixel cell area  $A_{px}$

$$r_f(x, y) = \int_{A_{px}} f(x - x', y - y') r(x', y') dA \quad (5.31)$$

Come meglio trattato in capitolo 1, lo *stimatore* di monte carlo ci permette di stimare tale integrale come

$$r_f(x, y) \approx \frac{1}{n} \sum_{i=1}^n \frac{f(x - x_i, y - y_i) r(x_i, y_i)}{p(x_i, y_i)} \quad (5.32)$$

se consideriamo il sample point come vettore aleatorio con distribuzione uniforme, allora l'equazione diventa

$$r_f(x, y) \approx \frac{\|A_{px}\|}{n} \sum_{i=1}^n f(x - x_i, y - y_i) r(x_i, y_i) \quad (5.33)$$

Equazione 5.33 può essere migliorata in termini di varianza pesando ciascun campione affinché la somma dei valori filtrati sia pari a 1 anche con campioni finiti (la sua aspettazione è 1). Tale estimatore è detto *weighted importance sampling Monte Carlo estimator*

$$r_f(x, y) \approx \frac{\sum_{i=1}^n f(x - x_i, y - y_i) r(x_i, y_i)}{\sum_{i=1}^n f(x - x_i, y - y_i)} \quad (5.34)$$

Tali formule hanno il difetto di richiedere tutti i campioni nel film plane per stimare la radianza spettrale in un singolo pixel. La soluzione consiste nel scegliere opportunamente un filtro che

lascia passare solo i campioni all'interno del pixel stesso, e campionare la distribuzione di samples con una distribuzione simile alla filter function. Tale approccio è nominato *filter importance sampling* (vedi 1).

$$\begin{aligned} r_f(x, y) &\approx \frac{1}{n} \sum_{i=1}^n \left( \frac{f(x - x_i, y - y_i)}{p(x_i, y_i)} r(x_i, y_i) \right) \\ &= \frac{\left( \int_{\mathbb{R}^2} |f(x', y')| dA \right)}{n} \sum_{i=1}^n \text{sgn}(f(x - x_i, y - y_i)) r(x_i, y_i) \end{aligned}$$

Combinando tale approccio con *weighted importance sampling*

$$r_f(x, y) \approx \frac{\sum_{i=1}^n w(x - x_i, y - y_i) r(x_i, y_i)}{\sum_{i=1}^n w(x - x_i, y - y_i)} \quad (5.35)$$

con

$$w = \frac{f(x, y)}{p(x, y)}$$

Con il primo approccio, 5.34, il vantaggio è quello di calcolare una stima per ogni pixel basata su un numero maggiore di campioni distribuiti su tutto il film plane. Tale approccio ha i suoi svantaggi nella performance, e nel fatto che se i campioni sono ben distribuiti una tale ricostruzione non è necessaria, rendendo 5.35 un buon approccio in questo caso.

# Chapter 6

## Modelli di Fonti Luminose e Superfici

### 6.1 Modellazione di Superfici

Quando un flusso radiante incide su una superficie, i tre processi che avvengono sono *riflessione*, *trasmissione*, *assorbimento*, oltre all'emissione per tutti i materiali al di sopra dello zero assoluto. In particolare, durante la propagazione di un flusso radiante in un mezzo, parte della sua energia viene assorbita o deviata dal mezzo<sup>1</sup>. Quando la luce incontra un oggetto, in particolare la sua superficie esterna, detta *interfaccia*, tale flusso radiante può essere trasmesso e/o riflesso in diverse direzioni e con diversa intensità.

Per il principio di conservazione dell'energia, all'interfaccia deve valere la proprietà<sup>2</sup>

$$\bar{\rho} + \bar{\tau} = 1 \quad (6.1)$$

Si noti l'assenza del coefficiente di assorbimento, in quanto esso non è rilevante nell'interazione luce-interfaccia, ma partecipa in processi che coinvolgono la propagazione all'interno del volume del materiale, nel quale vale, per ogni lunghezza d'onda, in ogni istante, in ogni

<sup>1</sup>d'ora in poi in questo capitolo si trascura la propagazione nel mezzo e si considera propagazione nel vuoto

<sup>2</sup>notazione introdotta in seguito

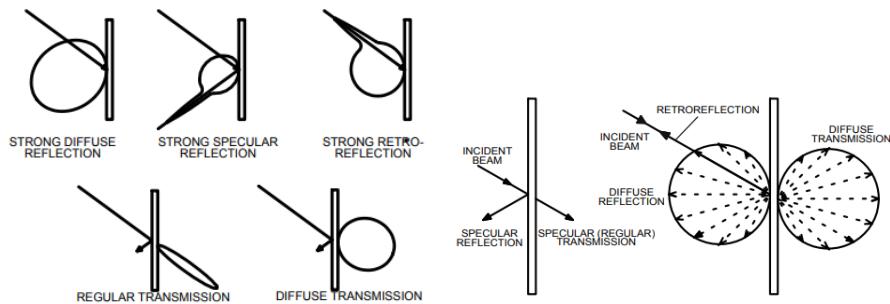


Figure 6.1: tipi di riflessione e trasmissione. Immagine da [Jam09]

punto dello spazio, ancora la conservazione dell'energia

$$\alpha + \rho + \tau = 1 \quad (6.2)$$

Le proprietà ottiche che coinvolgono tali processi sono suddivise in spettrali e non, ed in estinseche (suffisso "-ance") ed intrinseche (suffisso "-ivity"), queste ultime caratterizzanti il materiale ed utilizzate esclusivamente per materiali puri. In particolare seguono le definizioni del CIE Lighting Vocabulary

Reflectance:	Per una data radiazione con data SPD, polarizzazione, geometria: Rapporto tra flusso radiante o luminoso riflesso e flusso radiante o riflesso incidente. Nel primo caso, le grandezze possono essere spettrali o cumulative. Misura adimensionale. Simbolo $\rho$
Trasmittance:	Per una data radiazione con data SPD, polarizzazione, geometria: Rapporto tra flusso radiante o luminoso trasmesso e flusso radiante o riflesso incidente, nelle date condizioni (come la lunghezza del percorso della radiazione nell mezzo che si considera). Nel primo caso, le grandezze possono essere spettrali o cumulative. Misura adimensionale. Simbolo $\tau$
Absorptance:	Rapporto tra flusso radiante o luminoso assorbito e flusso radiante o luminoso incidente in date condizioni (come la lunghezza del percorso nel mezzo considerato). Simbolo $\alpha$

(vedi 6.1.1) Nota dalla tabella 6.1 come tali coefficienti dipendano dagli angoli solidi considerati per il calcolo del flusso incidente e riflesso/trasmesso. Distinguere tale computazione in categorie è più significativo per la reflectance (vedi 6.2).

Per caratterizzare macroscopicamente le proprietà di riflessione e trasmissione del materiale utilizziamo un approccio probabilistico, utilizzando una distribuzione<sup>3</sup> che trasforma Irradianza incidente in Radianza riflessa o trasmessa.

Nell'ipotesi che il materiale sia a regime, senza effetti di fosforescenza e fluorescenza (restringendoci all'ottica geometrica<sup>4</sup>), possiamo modellare il materiale con una densità di distribuzione che modella la tendenza del materiale alla riflessione e/o trasmissione della luce incidente, e in quali direzioni, trascurando il fenomeno di subsurface scattering e supponendo che essa fuoriesca esclusivamente dal punto di incidenza (dunque soffermandoci soltanto sui fenomeni a livello dell'interfaccia). Tale distribuzione è detta Bidirectional Reflectance Distribution Function(BRDF)

**Def. 6.1.1.** — La Bidirectional Reflectance Distribution Function(BRDF)  $f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i)$  Esprime la proporzione tra irradianza (spettrale) incidente nel punto  $\vec{p}$  da un angolo solido centrato in  $\hat{\omega}_i$  e radianza (spettrale) riflessa in direzione  $\hat{\omega}_o$ . Misurata in  $\text{sr}^{-1}$

$$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \frac{dL_r(\vec{p}, \hat{\omega}_o)}{dE_i(\vec{p}, \hat{\omega}_i)} = \frac{dL_r(\vec{p}, \hat{\omega}_o)}{L_i(\vec{p}, \hat{\omega}_i) \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i} \quad (6.3)$$

Le proprietà e casi d'uso di entrambe queste due distribuzioni saranno definite in seguito.

Riconosciamo, a seconda della distribuzione spaziale della BRDF/BSSRDF, quattro tipologie di riflessione fondamentali, in particolare

<sup>4</sup>Non propriamente una PDF, perchè non ha integrale unitario

<sup>4</sup>In quanto le distribuzioni qui definite, nell'ipotesi di ottica geometrica, non cambiano la lunghezza d'onda della radiazione incidente, la dipendenza di radianza, irradianza, flusso, con la lunghezza d'onda è omessa per comodità

Coefficiente	Formula
Spectral Reflectance	$\rho_\lambda(\lambda) = \frac{\Phi_{e,\lambda}^r(\lambda)}{\Phi_{e,\lambda}^i(\lambda)}$
Reflectance	$\rho = \frac{\Phi_e^r}{\Phi_e^i} = \frac{\int_0^\infty \rho_\lambda(\lambda) \Phi_{e,\lambda}^i d\lambda}{\int_0^\infty \Phi_{e,\lambda}^i d\lambda} \neq \int_0^\infty \rho_\lambda(\lambda) d\lambda$
Spectral Transmittance	$\rho_\lambda(\lambda) = \frac{\Phi_{e,\lambda}^r(\lambda)}{\Phi_{e,\lambda}^i(\lambda)}$
Trasmittance	$\tau = \frac{\Phi_e^t}{\Phi_e^i} = \frac{\int_0^\infty \tau_\lambda(\lambda) \Phi_{e,\lambda}^i d\lambda}{\int_0^\infty \Phi_{e,\lambda}^i d\lambda} \neq \int_0^\infty \tau_\lambda(\lambda) d\lambda$
Spectral Absorptance	$\alpha_\lambda(\lambda) = \frac{\Phi_{e,\lambda}^a(\lambda)}{\Phi_{e,\lambda}^i(\lambda)}$
Absorptance	$\alpha = \frac{\Phi_e^a}{\Phi_e^i} = \frac{\int_0^\infty \rho_\lambda(\lambda) \Phi_{e,\lambda}^i d\lambda}{\int_0^\infty \Phi_{e,\lambda}^i d\lambda} \neq \int_0^\infty \alpha_\lambda(\lambda) d\lambda$

Table 6.1: formule per le definizioni 6.1

*Diffuse Reflection* la superficie si comporta come lambertiana rispetto alla riflessione, cioè distribuisce equamente tutta<sup>5</sup> il flusso incidente

*Glossy Specular Reflection* la superficie predilige un sottoinsieme di direzioni per la riflessione

*Perfectly Specular Reflection* la superficie riflette secondo la legge della riflessione per le superfici otticamente lisce

*Retroreflective Material* la superficie riflette gran parte del flusso nelle direzioni vicine a quella di incidenza

Si noti che la trasmissione, modellata con una funzione analoga alla BRDF, la BTDF, è categorizzata in tre tipologie analoghe alle prime tre sopraindicate per la riflessione. Vedi figura 6.1.

Come precedentemente accennato, per il calcolo della reflectance, vengono presi diversi angoli solidi come riferimento, vedi Tabella 6.2.

---

<sup>5</sup>Senza contare assorbimento

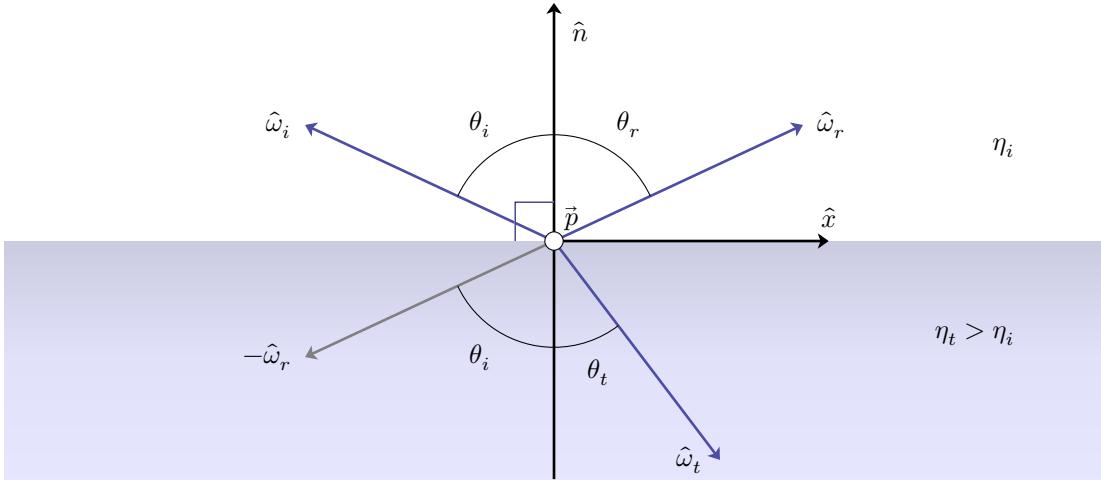


Figure 6.2: Illustrazione geometrica riflessione e rifrazione speculare

Table 6.2: Tipi di reflectance proposti da [al77]. Nota che di solito il coseno dell'angolo rispetto allo zenith della direzione incidente non è preso con valore assoluto, in quanto si assume  $\hat{n}$  come la normale che forma un angolo acuto con  $\hat{\omega}_i$ . Seguendo la convenzione di [PJH23], qui invece la normale è assunta sempre uscente dalla superficie.

Coefficiente	Formula
Hemispherical-Directional Reflectance	
	$\rho(\vec{p}, \hat{\omega}_o) = \int_{\mathcal{H}^2(\hat{n})} f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i)  \langle \hat{n}, \hat{\omega}_i \rangle  d\hat{\omega}_i \quad (6.4)$
Bi-Hemispherical Reflectance	
	$\rho(\vec{p}) = \frac{1}{\pi} \int_{\mathcal{H}^2(\hat{n}_i)} \int_{\mathcal{H}^2(\hat{n}_o)} f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i)  \langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle  d\hat{\omega}_o d\hat{\omega}_i \quad (6.5)$

### 6.1.1 Proprietà ottiche delle interfacce

Trascurando i fenomeni di scattering e assorbimento che avvengono all'interno del materiale, e concentrando solo sulle interazioni a livello di interfaccia. Partiamo analizzando con queste convenzioni le superfici perfettamente speculari, la cui propagazione delle radiazioni, con assunzione di ottica geometrica, è interamente governata dalle

$$\left. \begin{array}{l} \text{Legge della riflessione} \quad \theta_i = \theta_r \\ \text{Legge di Snell}^6 \quad \eta_i \sin \theta_i = \eta_t \sin \theta_t \end{array} \right\} \varphi_o = \varphi_i + \pi \quad (6.6)$$

Ricordiamo che le implementazioni per la Legge di Snell devono tener conto della possibilità di *riflessione interna totale*, nel caso in cui  $\eta_t < \eta_i$  e  $\theta_i \geq \theta_c = \arcsin(\eta_i/\eta_t)$ .

Le direzioni di riflessione e rifrazione sono computate mediante osservazioni geometriche (Figura 6.2)

$$\hat{\omega}_r = \tau_r(\hat{n}, \hat{\omega}_i) = -(\hat{\omega}_i - 2\langle \hat{n}, \hat{\omega}_i \rangle) = 2\langle \hat{n}, \hat{\omega}_i \rangle - \hat{\omega}_i \quad (6.7)$$

Mentre, per la trasmissione, sapendo che geometricamente

$$\sin \theta_i \hat{x} = \cos \theta_i \hat{n} - \hat{\omega}_i \quad (6.8)$$

e che per la legge di Snell

$$\cos \theta_t = \sqrt{1 - \sin^2 \theta_t} = \sqrt{1 - \frac{\eta_i^2}{\eta_t^2} \sin^2 \theta_i} = \sqrt{1 - \frac{\eta_i^2}{\eta_t^2} (1 - \cos^2 \theta_i)} \quad (6.9)$$

allora la direzione di trasmissione

$$\begin{aligned} \hat{\omega}_t &= \tau_t(\hat{n}, \hat{\omega}_i) = \sin \theta_t \hat{x} - \cos \theta_t \hat{n} \\ &\stackrel{(6.8)}{=} \frac{\sin \theta_t}{\sin \theta_i} (\cos \theta_i \hat{n} - \hat{\omega}_i) - \cos \theta_t \hat{n} \\ &\stackrel{(6.6)}{=} \left( \frac{\eta_i}{\eta_t} \cos \theta_i - \cos \theta_t \right) \hat{n} - \frac{\eta_i}{\eta_t} \hat{\omega}_i \\ &\stackrel{(6.9)}{=} \frac{\eta_i}{\eta_t} \left( \left( \cos \theta_i - \frac{\eta_t}{\eta_i} \sqrt{1 - \frac{\eta_i^2}{\eta_t^2} (1 - \cos^2 \theta_i)} \right) \hat{n} - \hat{\omega}_i \right) \\ &= \frac{\eta_i}{\eta_t} \left( \left( \cos \theta_i - \sqrt{\frac{\eta_t^2}{\eta_i^2} - 1 + \cos^2 \theta_i} \right) \hat{n} - \hat{\omega}_i \right) \\ &= \frac{\eta_i}{\eta_t} \left( \left( \langle \hat{n}, \hat{\omega}_i \rangle - \sqrt{\frac{\eta_t^2}{\eta_i^2} - 1 + \langle \hat{n}, \hat{\omega}_i \rangle^2} \right) \hat{n} - \hat{\omega}_i \right) \end{aligned} \quad (6.10)$$

Anche le frazioni di radiazione incidente trasmessa e riflessa dipendono dall'indice di rifrazione, secondo le *Equazioni di Fresnel*, che forniscono le relazioni per il rapporto tra rispettivamente campo elettrico riflesso e campo elettrico incidente, e campo elettrico trasmesso e campo elettrico incidente, per ognuna delle due componenti di polarizzazione<sup>7</sup>. Tali equazioni possono essere anche espresse per relazionare i rapporti tra potenza riflessa e incidente, e potenza

<sup>6</sup>Nota come i contributi delle varie lunghezze d'onda della radiazione incidente sono trasmessi ad angoli rispetto alla normale  $\theta_t$  differenti, effetto noto come *Dispersione*. Ciò vuol dire che per ogni lunghezza d'onda campionata bisogna generare un nuovo raggio

<sup>7</sup>polarizzazione *s* (simbolo  $\perp$ ), che indica che il campo elettrico oscilla con polarizzazione planare perpendicolarmente al piano di incidenza, e polarizzazione *p* (simbolo  $\parallel$ ), che indica che il campo elettrico oscilla con polarizzazione planare parallelamente al piano di incidenza

trasmessa e incidente<sup>8</sup>.

$$\bar{\rho}_{\perp}(\mu) = \frac{(a - \mu)^2 + b^2}{(a + \mu)^2 + b^2} \in [0, 1] \quad (6.12)$$

$$\bar{\rho}_{\parallel}(\mu) = \frac{(a - \mu)^2 + b^2}{(a + \mu)^2 + b^2} \frac{(a - \frac{1-\mu^2}{\mu})^2 + b^2}{(a - \frac{1+\mu^2}{\mu})^2 + b^2} \in [0, 1] \quad (6.13)$$

$$\bar{\tau}_{\perp}(\mu) = 1 - \bar{\rho}_{\perp} \quad (6.14)$$

$$\bar{\tau}_{\parallel}(\mu) = 1 - \bar{\rho}_{\parallel} \quad (6.15)$$

dove<sup>9</sup>

$$\begin{aligned} a &= \sqrt{\frac{\sqrt{c^2 + 4\eta\kappa} + c}{2}} & \mu &= \cos \theta_i = \langle \hat{n}, \hat{\omega}_i \rangle \\ b &= \sqrt{\frac{\sqrt{c^2 + 4\eta\kappa} - c}{2}} & \eta &= \frac{\eta_i \eta_t + \kappa_t \kappa_i}{\eta_i^2 + \kappa_i^2} \\ c &= \eta^2 - \kappa^2 - (1 - \mu^2) & \kappa &= \frac{\kappa_t \eta_i - \eta_t \kappa_i}{\eta_i^2 + \kappa_i^2} \end{aligned}$$

Per luce non polarizzata tali coefficienti sono pari alla media tra i coefficienti per polarizzazione-s e polarizzazione-p. Di seguito la formula per la riflessione.<sup>10</sup>

$$F_r(\mu) = \frac{\bar{\rho}_{\perp}(\mu) + \bar{\rho}_{\parallel}(\mu)}{2} \quad (6.16)$$

Si noti che la reazione di una interfaccia ad una radiazione incidente varia considerevolmente con il variare dell'indice di rifrazione complesso, il quale ricordiamo varia con la lunghezza d'onda considerata, a tal punto che si distinguono due classi di materiali

Conduttori:

Caratterizzati da alto indice di assorbimento  $\kappa_a$  e basso indice di rifrazione reale  $\eta$ , il che determina una reflectance considerevole, ma al contempo altamente variabile con la lunghezza d'onda (vedi Figura 6.3). Si spiega la tinta tipica di molti metalli. La restante parte che viene trasmessa viene rapidamente assorbita, rendendola trascurabile

Semiconduttori:

Li trascuriamo

Dielettrici:

Essi non conducono elettricità, hanno indice di rifrazione reale, con  $\kappa_a$  trascurabile, il che significa che, durante la trasmissione al loro interno, la radianza è in quota parte assorbita, ma non del tutto. A livello dell'interfaccia, le caratteristiche sopracitate determinano una reflectance considerabile pressoché costante con l'angolo di incidenza  $\theta_i$  (vedi Figura 6.3)

<sup>8</sup>Come prima detto, per la conservazione dell'energia, tali coefficienti devono sommarsi a 1, il che accade solo se si trascura il coefficiente di assorbimento  $\alpha$ , il che è un'ipotesi per l'analisi della sola interfaccia.

<sup>9</sup> $\eta$  e  $\kappa$  si dicono rispettivamente *Indice relativo di rifrazione* e *Indice relativo di absorption*

<sup>10</sup>Si precisa nuovamente che tale formula è accurata soltanto per interfacce e half space materials (vedi 6.1, reflectivity), ciò che attuiamo è una approssimazione. Inoltre si trascura la polarizzazione-s della luce riflessa che si verifica per angolo di incidenza pari all'Angolo di Brewster

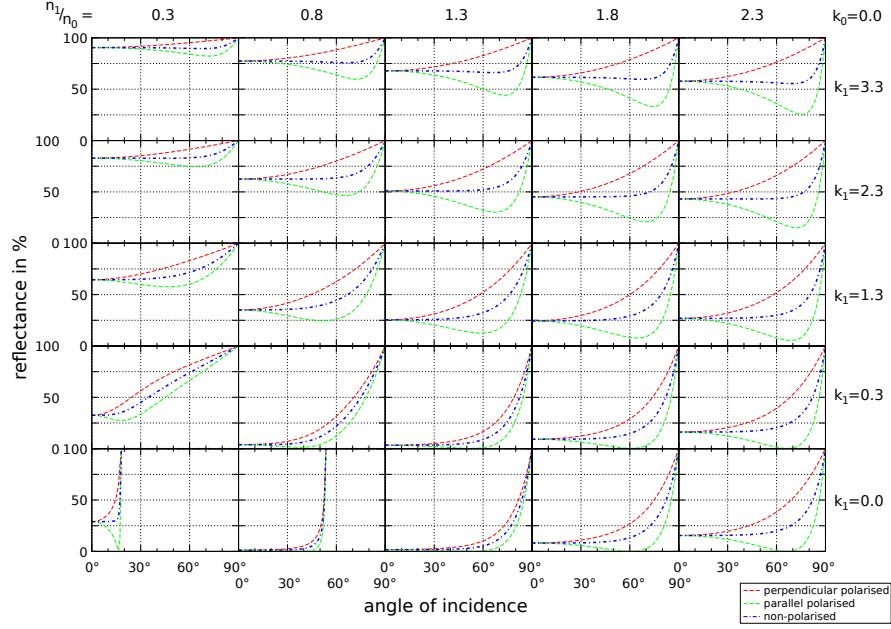


Figure 6.3: Coefficiente di Riflessione di Fresnel per una interfaccia al variare dell'indice di rifrazione complessa.

Approssimazioni delle equazioni di Fresnel, per abbattere il costo computazionale, includono l'approssimazione di Lazányi [LS05]

$$F_r(\mu) \approx \frac{(\eta - 1)^2 + \kappa^2 + 4\eta(1 - \mu)^5}{(\eta + 1)^2 + \kappa^2} \quad (6.17)$$

e, per i dielettrici, l'approssimazione di Schlick [Sch94], che richiede la conoscenza della reflectance per incidenza normale  $F_0$

$$F_r(\mu) \approx F_0 + (1 - F_0)(1 - \mu)^5 \quad (6.18)$$

con

$$F_0 = \frac{(\eta - 1)^2 + \kappa^2}{(\eta + 1)^2 + \kappa^2} \underset{\kappa \approx 0}{=} \left( \frac{\eta - 1}{\eta + 1} \right)^2$$

Inoltre, per la categoria dei dielettrici, si possono utilizzare delle formule semplificate

$$\bar{\rho}_{\perp} = \left( \frac{\eta \cos \theta_t - \cos \theta_i}{\eta \cos \theta_t + \cos \theta_i} \right)^2 \quad (6.19)$$

$$\bar{\rho}_{\parallel} = \left( \frac{\cos \theta_t - \eta \cos \theta_i}{\cos \theta_t + \eta \cos \theta_i} \right)^2 \quad (6.20)$$

### 6.1.2 BRDF

Come già definito prima, la Bidirectional Reflectance Distribution Function<sup>11</sup> esprime la densità di distribuzione emisferica di radianza incidente all'occorenza di un evento di scattering

dalla superficie

$$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \frac{dL_r(\vec{p}, \hat{\omega}_o)}{dE_i(\vec{p}, \hat{\omega}_i)} = \frac{dL_r(\vec{p}, \hat{\omega}_o)}{L_i(\vec{p}, \hat{\omega}_i) \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i}$$

Tale che l'integrale nel suo dominio sia pari all'*albedo* della superficie

$$\frac{1}{\bar{\rho}(\vec{p}, \hat{\omega}_o)} \int_{\mathcal{H}^2(\hat{n})} f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i = 1$$

Affinchè sia fisicamente plausibile, inoltre, deve obbedire alle tre proprietà di

Non Negatività	$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) \geq 0$
Conservazione Energia	$\frac{\int_{\mathcal{H}^2(\hat{n})} L_i(\vec{p}, \hat{\omega}_i) \bar{\rho}(\vec{p}, \hat{\omega}_i) \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i}{\int_{\mathcal{H}^2(\hat{n})} L_i(\vec{p}, \hat{\omega}_i) \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i} \leq 1$
Helmontz Reciprocity	$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = f_r(\vec{p}, \hat{\omega}_i, \hat{\omega}_o)$

(6.21)

Densità di Distribuzione analoga è la BTDF  $f_t(\vec{p}, \hat{\omega}_o, \hat{\omega}_i)$ , la quale è non nulla per superfici non opache, e gode di tutte le proprietà sopracitate, ma la reversibilità del cammino ottico viene generalizzata come<sup>12</sup>

$$\frac{f_s(\vec{p}, \hat{\omega}_o, \hat{\omega}_i)}{\eta_o^2} = \frac{f_s(\vec{p}, \hat{\omega}_i, \hat{\omega}_o)}{\eta_i^2} \quad (6.22)$$

Dalle definizioni di BRDF e BTDF (6.3), possiamo ricavare una equazione integrale che esprime la radianza riflessa e trasmessa al punto  $\vec{p}$ , la cui valenza è ristretta nell'assunzione di trascurare i processi di assorbimento e scattering che avvengono all'interno del materiale, dette *reflectance equation* e *transmittance equation*

$$L_r(\vec{p}, \hat{\omega}_o) = \int_{\mathcal{H}^2(\hat{n})} L_i(\vec{p}, \hat{\omega}_i) f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \quad (6.23)$$

$$L_t(\vec{p}, \hat{\omega}_o) = \int_{\mathcal{H}^2(-\hat{n})} L_i(\vec{p}, \hat{\omega}_i) f_t(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{n}, -\hat{\omega}_i \rangle| d\hat{\omega}_i \quad (6.24)$$

Le quali, definendo la distribuzione *Bidirectional Scattering Distribution Function* (BSDF), possiamo riunire le due densità di distribuzione

$f_s(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) + f_t(\vec{p}, \hat{\omega}_o, \hat{\omega}_i)$ , dove

$$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = 0, \text{ se } \langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle \leq 0 \quad (6.25)$$

$$f_t(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = 0, \text{ se } \langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle \geq 0 \quad (6.26)$$

per la quale possiamo aggregare i due integrali nella *Scattered Radiance*<sup>13</sup>

$$L_s(\vec{p}, \hat{\omega}_o) = \int_{S^2} L_i(\vec{p}, \hat{\omega}_i) f_s(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \quad (6.27)$$

Dove, aggiungendo l'eventuale radianza emessa dalla superficie, otteniamo un'equazione nota come *Rendering Equation*, la quale ci proponiamo di risolvere numericamente

$$L_o(\vec{p}, \hat{\omega}_o) = L_e(\vec{p}, \hat{\omega}_o) + \int_{S^2} L_i(\vec{p}, \hat{\omega}_i) f_s(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \quad (6.28)$$

<sup>11</sup>implementare una BSSRDF è fuori scope

<sup>12</sup>quando usiamo  $f_s$ , ci riferiamo a entrambe BRDF e BTDF, [Peg16]

<sup>13</sup>considerazioni sul campionamento della funzione integranda saranno fatte in seguito

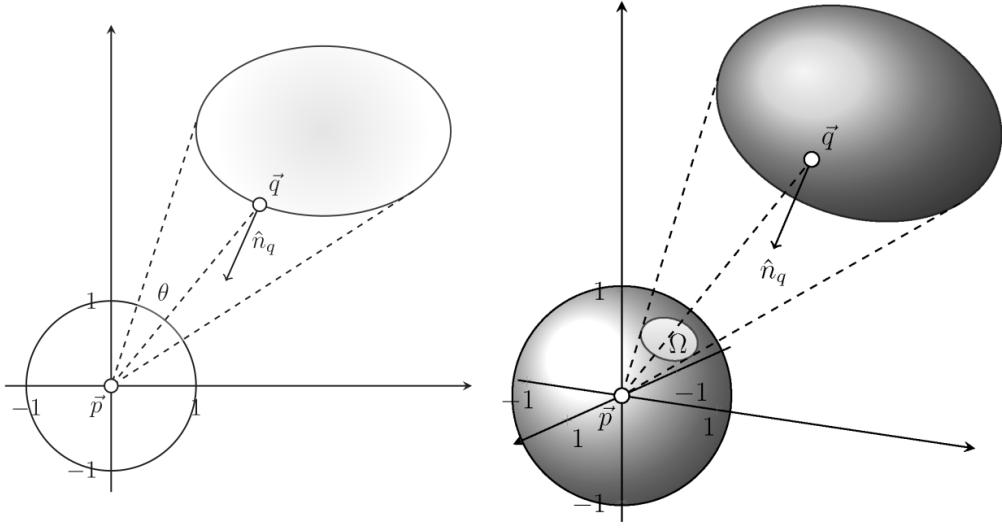


Figure 6.4: Illustrazione dell'angolo (a sinistra) e angolo solido (a destra) sottesi da un punto dalla superficie visibile di un oggetto arbitrario. Immagine da [Peg16]

*Assumiamo propagazione nel vuoto*, affinché la radianza si conservi nel suo percorso. Si noti la natura ricorsiva della equazione soprascritta. La radianza incidente su una superficie  $L_i$  risulta essere a sua volta radianza osservata uscente da un'altra superficie o fonte luminosa della scena, tale che, se definiamo la  $t(\vec{p}, \hat{\omega})$  ray casting function, la quale, dato un punto  $\vec{p}$  e una direzione  $\hat{\omega}$ , restituisce il primo punto di intersezione che una semiretta definita da tale punto e direzione incontra.<sup>14</sup>

$$L_i(\vec{p}, \hat{\omega}) = L_o(t(\vec{p}, -\hat{\omega})) \quad (6.29)$$

Da cui riscriviamo la Rendering Equation

$$L_o(\vec{p}, \hat{\omega}_o) = L_e(\vec{p}, \hat{\omega}_o) + \int_{S^2} L_o(t(\vec{p}, \hat{\omega}_i), -\hat{\omega}_i) f_s(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \quad (6.30)$$

Come meglio spiegato in seguito (Capitolo 3), Equazione 6.30 è complessa anche perché la relazione tra gli oggetti nella scena è implicita, incapsulata nella ray tracing function  $t(\vec{p}, \hat{\omega})$ . Una forma più trattabile di tale equazione integrale la si può trovare trasformando l'integrale su un angolo solido in un integrale su un area (vedi Figura 6.4). L'angolo solido sotteso da un oggetto ad un punto  $\vec{p}$  è una misura di *area proiettata*, di una superficie parte di un oggetto visibile da  $\vec{p}$ , sulla sfera unitaria centrata in  $\vec{p}$ , ed è funzione sia dell'orientamento della normale della superficie osservata  $\hat{n}_q$  ad ogni punto  $\vec{q} = \vec{p} + \|\vec{q} - \vec{p}\|\hat{\omega} = t((\vec{p}, \hat{\omega}_i), -\hat{\omega}_i)$ , tale che  $\hat{p}\hat{q} = \hat{\omega}$ , e sia della sua distanza da  $\vec{p}$ . Per esprimere tale relazione per ogni direzione dello spazio, definiamo  $V(\vec{p}, \vec{q})$  funzione di visibilità binaria tra  $\vec{p}$  e  $\vec{q}$ , tale che  $V(\vec{p}, \vec{q}) = 1$  se  $\vec{q}$  appartiene all'insieme di superfici visibili da  $\vec{p}$ , e  $V(\vec{p}, \vec{q}) = 0$  se  $\vec{p}$  e  $\vec{q}$  sono mutualmente occlusi. Calcolare tale area proiettata consiste nel proiettare ciascun elemento di superficie su una sfera di raggio  $\|\vec{q} - \vec{p}\|$  moltiplicando per il coseno dell'angolo formato tra la congiungente  $\vec{p}$  a  $\vec{q}$ , e la normale  $\hat{n}_q$ , e dividere per il quadrato dell'area della sfera, in accordo con la definizione di angolo

<sup>14</sup>tale semiretta è definita come "raggio", per come essa è implementata in codice: si fa avanzare un punto lungo tale retta a piccoli incrementi, e ad ogni step si controlla per intersezione con un oggetto della scena

solido

$$d\hat{\omega} = \frac{V(\vec{p}, \vec{q})}{\|\vec{p} - \vec{q}\|^2} \langle \hat{n}_q, -\hat{p}q \rangle dA(\vec{q}) \quad (6.31)$$

Tale relazione ricavata consiste nel jacobiano da utilizzare nell'integrale 6.30 per la sostituzione delle variabili. Inoltre, definiamo il Termine Geometrico  $G(\vec{p}, \vec{q})$ , aggiungendo ai fattori della precedente formula il coseno dell'angolo tra direzione congiungente e  $\hat{n}_p$ , effettivamente proiettando angolo solido  $d\hat{\omega}$  sul piano equatoriale della sfera unitaria centrata in  $\vec{p}$

$$G(\vec{p}, \vec{q}) = \langle \hat{n}_p, \hat{p}q \rangle \frac{V(\vec{p}, \vec{q})}{\|\vec{p} - \vec{q}\|^2} \langle \hat{n}_q, -\hat{p}q \rangle \quad (6.32)$$

tale che

$$\langle \hat{n}_p, \hat{\omega} \rangle d\hat{\omega} = \langle \hat{n}_p, \hat{p}q \rangle \frac{V(\vec{p}, \vec{q})}{\|\vec{p} - \vec{q}\|^2} \langle \hat{n}_q, -\hat{p}q \rangle = G(\vec{p}, \vec{q}) dA(\vec{q}) \quad (6.33)$$

Possiamo dunque riformulare la Rendering Equation per essere un integrale nell'area dell'unione di tutte le superfici visibili dal punto  $\vec{p}$

$$L_o(\vec{p}, \hat{p}q) = L_e(\vec{p}, \hat{p}q) + \int_A f_s(\vec{p}, \hat{\omega}_o, \hat{p}q) G(\vec{p}, \vec{q}) L_o(\vec{q}, -\hat{p}q) dA(\vec{q}) \quad (6.34)$$

Torneremo in seguito su tale argomento.

### 6.1.3 Modelli di BRDF

Modelli di BRDFs derivano da diverse fonti, in particolare possono essere dati misurati in laboratorio, da una simulazione di particelle, o da modelli analitici.

Come si osserva dall'Equazione 6.27, necessitiamo di strategie per campionare punti della funzione integranda per date direzioni  $\hat{\omega}_i$ , affinché si assicuri una buona convergenza allo Stimatore di Monte Carlo per integrale<sup>15</sup>. Una delle strategie proposte ed approfondite in seguito, è quella di campionare la funzione integranda con densità di probabilità  $p$  proporzionale alla BRDF

$$p(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \frac{f_s(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle|}{\bar{\rho}(\vec{p}, \hat{\omega}_o)} \quad (6.35)$$

Dunque, assumendo di poter campionare da una distribuzione uniforme una osservazione della variabile aleatoria  $\xi \sim \mathcal{U}(0, 1)$ , la cui implementazione si rimanda a Appendice A, analizziamo come poter trasformare tale variabile aleatoria per poter campionare con PDF proporzionale alla BRDF.

Analizziamo dunque due modelli fenomenologici, per poi passare a quelli basati su microgeometry

#### Lambertian BRDF

Ricordando l'equazione per *Hemispherical-Directional Reflectance* (Equazione 6.4), supponiamo che la BRDF sia costante nel dominio di integrazione, ovvero che la superficie sia lambertiana rispetto alla riflessione e opaca<sup>16</sup>, producendo riflessione diffusa perfetta.

<sup>15</sup>importance Sampling

<sup>16</sup>Inoltre, come sempre, ci limitiamo a considerare, con l'ottica geometrica, superfici solamente a livello dell'interfaccia

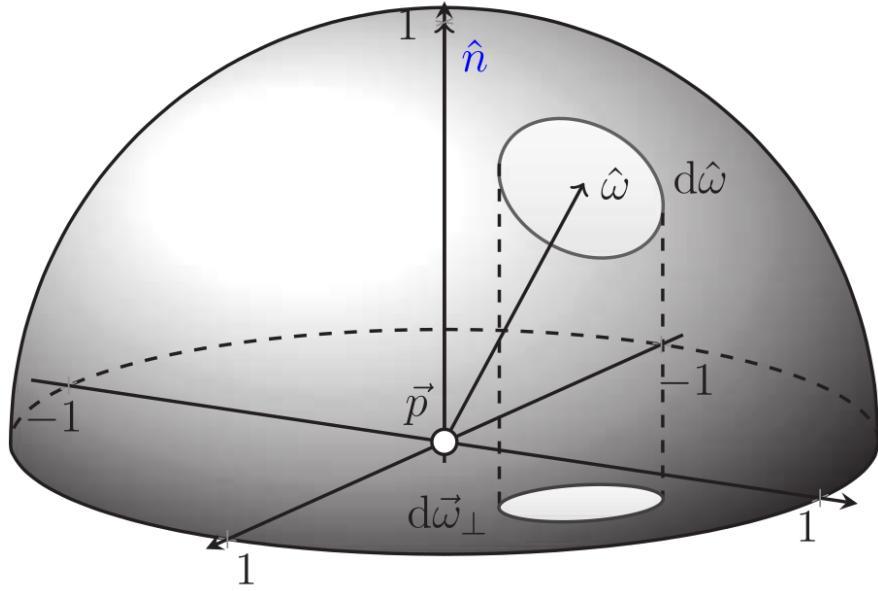


Figure 6.5: Illustrazione dell'area proiettata dell'emisfera secondo l'equazione  $d\hat{\omega}_\perp = \langle \hat{n}, \hat{\omega} \rangle d\hat{\omega}$ . Figura da [Peg16]

$$\rho(\vec{p}, \hat{\omega}_o) = f_r(\vec{p}) \int_{\mathcal{H}^2(\hat{n})} |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i = f_r(\vec{p}) \pi = \rho(\vec{p}) \quad (6.36)$$

da cui

$$f_r(\vec{p}) = \frac{\rho(\vec{p})}{\pi} \quad (6.37)$$

Il campionamento di tale BRDF può essere compiuto estraendo una direzione casuale nell'emisfera proiettata (vedi Figura 6.5). Tale proiezione è compiuta in quanto la BRDF è semplice, ed è moltiplicata per tale coseno. La PDF in questione è la seguente: Siano  $\hat{\omega}_z$  direzione dello zenith,  $\hat{\omega}_i$  direzione incidente, la *Cosine-lobe distribution* centrata attorno alla direzione dello zenith, con ripidità  $n$ , è

$$c_n(\omega_z) = \frac{n+1}{2\pi} \langle \hat{\omega}_z, \hat{\omega}_i \rangle^n \quad (6.38)$$

Applichiamo la tecnica di *Inverse Transform Sampling*, che, come approfondito in seguito (Capitolo 1), permette di trasformare una realizzazione campionata con una data PDF (nel nostro caso,  $\mathcal{U}(0, 1)$ ) ad una arbitraria, per la quale bisogna però essere in grado di calcolare analiticamente la CDF ed invertirla. Per distribuzioni multivariate, tale tecnica è applicabile esclusivamente se la CDF è *separabile*.

L'integrale in un angolo solido  $\Omega_i$  per la *Cosine-Lobe Distribution* risulta essere separabile

nella forma

$$\begin{aligned} \int_{\Omega_i} c_n(\hat{\omega}_z, \hat{\omega}_i) d\hat{\omega}_i &= \int_0^{\varphi_i} \frac{1}{2\pi} d\varphi \int_{\cos \theta_i}^1 (n+1) \mu^n d\mu \\ &= \left[ \frac{\varphi_i}{2\pi} \right]_{\varphi=0}^{\varphi_i} \left[ \mu^{n+1} \right]_{\mu=\cos \theta_i}^1 \\ &= \frac{\varphi_i}{2\pi} (1 - \cos^{n+1} \theta_i) \end{aligned} \quad (6.39)$$

Applicando *Inverse Transform Sampling* permette di compiere *Importance Sampling* come

$$\xi_\varphi = \frac{\varphi_i}{2\pi} \rightarrow \varphi_i = 2\pi\xi_\varphi \quad (6.40)$$

$$\xi_\theta = 1 - \cos^{n+1} \theta_i \rightarrow \theta_i = \arccos \left( \sqrt[n+1]{1 - \xi_\theta} \right) \quad (6.41)$$

Il che permette di ottenere le coordinate sferiche della direzione campionata. Dunque, nel caso della Lambertian BRDF, campioniamo secondo la PDF

$$p(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = c_1(\hat{n}, \hat{\omega}_i) \quad (6.42)$$

### Specular BDRF

Possiamo derivare una BRDF per una superficie che mostra riflessione speculare imponendo la condizione<sup>17</sup>

$$L_o(\vec{p}, \hat{\omega}_o) = \int_{\mathcal{H}^2(\hat{n})} f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) L_i(\vec{p}, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \quad (6.43)$$

$$= \begin{cases} F_r(\cos \theta_r) L_i(\vec{p}, \hat{\omega}_r) & \text{se } \hat{\omega}_o = \tau_r(\hat{n}, \hat{\omega}_r) \\ 0 & \text{altrimenti} \end{cases} \quad (6.44)$$

La quale risulta essere evidente come una Distribuzione di dirac moltiplicata per un opportuno fattore<sup>18</sup>

$$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = F_r(\langle \hat{n}, \hat{\omega}_o \rangle) \frac{\delta(\hat{\omega}_i - \hat{\omega}_r)}{|\langle \hat{n}, \hat{\omega}_i \rangle|} \quad (6.45)$$

dove  $\hat{\omega}_r = \tau_r(\hat{n}, \hat{\omega}_o)$

Similmente, usando la legge di Snell, possiamo ricavare una specular BTDF, ricordando che per la conservazione dell'energia

$$\frac{d^2\Phi_t(\vec{p}, \hat{\omega}_t)}{d^2\Phi_i(\vec{p}, \hat{\omega}_i)} = \frac{L_t(\vec{p}, \hat{\omega}_t) dA(\vec{p}) \langle \hat{n}, \hat{\omega}_t \rangle d\hat{\omega}_t}{L_i(\vec{p}, \hat{\omega}_i) dA(\vec{p}) \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i} \quad (6.46)$$

---

<sup>17</sup>Si noti che  $F_r(\langle \hat{n}, \hat{\omega}_i \rangle) = F_r(\langle \hat{n}, \hat{\omega}_o \rangle) =$

<sup>18</sup>Le distribuzioni di Dirac, sia per le superfici che per le sorgenti luminose, devono essere riconosciute e trattate in maniera differente nelle routines di sampling

Possiamo calcolare il jacobiano  $d\hat{\omega}_t/d\hat{\omega}_i$  mediante osservazioni geometriche e la legge di Snell

$$\begin{aligned}
 \frac{d\hat{\omega}_t}{d\hat{\omega}_i} &= \frac{\sin \theta_t d\theta_t d\varphi_t}{\sin \theta_i d\theta_i d\varphi_i} \\
 &= \frac{\sin \theta_t}{\sin \theta_i} \frac{d \arcsin \left( \frac{\eta_i}{\eta_t} \sin \theta_i \right)}{d\theta_t} \\
 &= \frac{\eta_i}{\eta_t} \frac{\frac{\eta_i}{\eta_t} \cos \theta_i}{\sqrt{1 - \frac{\eta_i^2}{\eta_t^2} \sin^2 \theta_t}} \\
 &= \frac{\eta_i^2}{\eta_t^2} \frac{\cos \theta_i}{\sqrt{1 - \sin^2 \theta_t}} \\
 &= \frac{\eta_i^2}{\eta_t^2} \frac{\cos \theta_i}{\cos \theta_t}
 \end{aligned} \tag{6.47}$$

da cui

$$\frac{L_t(\vec{p}, \hat{\omega}_t) dA(\vec{p}) \langle \hat{n}, \hat{\omega}_t \rangle d\hat{\omega}_t}{L_i(\vec{p}, \hat{\omega}_i) dA(\vec{p}) \langle \hat{n}, \hat{\omega}_i \rangle d\hat{\omega}_i} = \frac{L_t(\vec{p}, \hat{\omega}_t) \eta_i^2}{L_i(\vec{p}, \hat{\omega}_i) \eta_t^2} = T_{\frac{\eta_t}{\eta_i}}(\cos \theta_i) \tag{6.48}$$

Da quest'ultima relazione si può ricavare che la transmittance  $\eta_i \rightarrow \eta_t$ , notazione  $T_{\frac{\eta_t}{\eta_i}}$ , calcolata in  $\cos \theta_i$ <sup>19</sup>, è uguale alla transmittance  $T_{\frac{\eta_i}{\eta_t}}$  calcolata in  $\cos \theta_o$ . Dunque la condizione

$$L_o(\vec{p}, \hat{\omega}_o) = \int_{\mathcal{H}^2(-\hat{n})} f_t(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) L_i(\vec{p}, \hat{\omega}_i) |\langle \hat{n}, -\hat{\omega}_i \rangle| d\hat{\omega}_i \tag{6.49}$$

$$= \begin{cases} \frac{\eta_t^2}{\eta_i^2} T_{\frac{\eta_i}{\eta_t}}(\langle \hat{n}, \hat{\omega}_o \rangle) L_i(\vec{p}, \hat{\omega}_i) & \text{se } \hat{\omega}_o = \tau_t(\hat{n}, \hat{\omega}_t) \\ 0 & \text{altrimenti} \end{cases} \tag{6.50}$$

dove  $T_{\frac{\eta_i}{\eta_t}}(\langle \hat{n}, \hat{\omega}_o \rangle) = T_{\frac{\eta_t}{\eta_i}}(\langle \hat{n}, \hat{\omega}_i \rangle) = 1 - F_r(\langle \hat{n}, \hat{\omega}_i \rangle) = 1 - F_r(\langle \hat{n}, \hat{\omega}_o \rangle)$ . Da cui, con ragionamento analogo al precedente, si può ricavare la BTDF come delta di Dirac

$$f_t(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \frac{\delta(\hat{\omega}_i - \hat{\omega}_t)}{-|\langle \hat{n}, \hat{\omega}_i \rangle|} T_{\frac{\eta_i}{\eta_t}}(|\langle \hat{n}, \hat{\omega}_o \rangle|) \frac{\eta_t^2}{\eta_i^2} \tag{6.51}$$

dove  $\hat{\omega}_t = \tau(\hat{n}, \hat{\omega}_o)$

Il campionamento di tali BSDFs è gestito in modo speciale, in quanto solo se la PDF di campionamento è anch'essa una distribuzione di Dirac, con picco nella stessa posizione, allora la sampling routine ritorna un valore non nullo.

#### 6.1.4 Modelli di superficie secondo Microfacet Theory

La *Microfacet Theory* è un approccio basato sull'ottica geometrica che modella una superficie, come una collezione di facce planari microscopiche, tipicamente con riflessione/trasmissione speculare (eccetto per il modello di Oren-Nayar, come si vedrà tra poco).

---

<sup>19</sup> $\cos \theta_i = \cos \theta_t$  se la condizione del delta di Dirac è soddisfatta

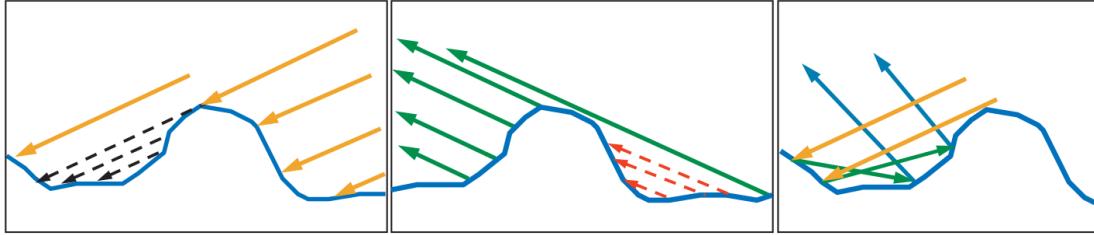


Figure 6.6: Illustrazione dei tre effetti di interazione tra microfacets e radiazione incidente. A sinistra, *Shadowing*, per il quale alcuni microfacets sono occlusi rispetto alla luce incidente da altri microfacets. Al centro, l'effetto complementare, *Masking*, per il quale alcuni microfacets sono occlusi rispetto all'osservatore da un altri microfacets. A destra, *Interreflection*, per il quale luce incidente può "rimbalzare" su diversi microfacets prima di raggiungere l'osservatore. Questi tre effetti complicano la definizione di una BRDF. Immagine da [AHH18]

Affinchè l'assunzione di ottica geometrica sia valida, queste micro asperità sono assunte molto più piccole (o molto più grandi) di una lunghezza d'onda. Una BRDF ricavata con tale teoria è un modello modellante lo scattering della luce in una grande collezione di microfacets con un *approccio probabilistico*. Tale caratterizzazione è complicata dalle interazioni tra i microfacets illustrate in Figura 6.6

Un elemento di area della superficie  $dA$  è composto da un gran numero di microfacets, ciascuno di essi con la propria normale  $\hat{m}$ . Possiamo caratterizzare tali normali come distribuite secondo una PDF proporzionale alla cosiddetta *Normal Distribution Function* (NDF)  $D(\vec{p}, \hat{m}) \geq 0$ . Maggiore è la varianza di tale distribuzione più la superficie è "rough". Per l'interazione dell'*Interreflection*, la luce può rimbalzare diverse volte prima di giungere all'osservatore, e tale contributo assume significatività diversa a seconda della categoria del materiale [AHH18]

- *Dielettrici*: Contributi da riflessioni multiple sono man mano meno significativi per via del loro medio/basso Fresnel factor (reflectance)
- *Conduttori*: Il colore mostra i suoi maggior contributi nei rimbalzi successivi, in quanto alta reflectance e alto coefficiente di assorbimento wavelength-selective fa sì che nei rimbalzi multipli la luce assume una polarizzazione e shift del colore più significativi

Inoltre, ciascun materiale reagisce differentemente alla penetrazione da parte della luce nella superficie dando luogo a *subsurface scattering* di diversa intensità. Nel caso in cui i microfacets sono molto più piccoli della lunghezza d'onda, possiamo approssimare affermando che la luce è uscita tutta nello stesso punto da cui è entrata, assunzione non precisa ma necessaria per usufruire del nostro modello di BRDF.

Ogni microfacet di normale  $\hat{m}$ , possiede anche una sua *Micro BRDF*  $f_\mu(\hat{m}, \hat{\omega}_o, \hat{\omega}_i)$ , i cui contributi sono cumulati per ottenere la BRDF.

La *Normal Distribution Function*, per convenzione, ha integrale pari al rapporto tra area della microsuperficie e macrosuperficie

$$\int_{\mathcal{S}^2} D(\vec{p}, \hat{m}) d\hat{m} = \frac{A_m}{A_n} \geq 1 \quad (6.52)$$

Inoltre, nel caso tipico in cui la NDF sia un *Height Field*, cioè

$$D(\vec{p}, \hat{m}) = 0, \quad \forall \hat{m} \mid \langle \hat{n}, \hat{m} \rangle \leq 0 \quad (6.53)$$

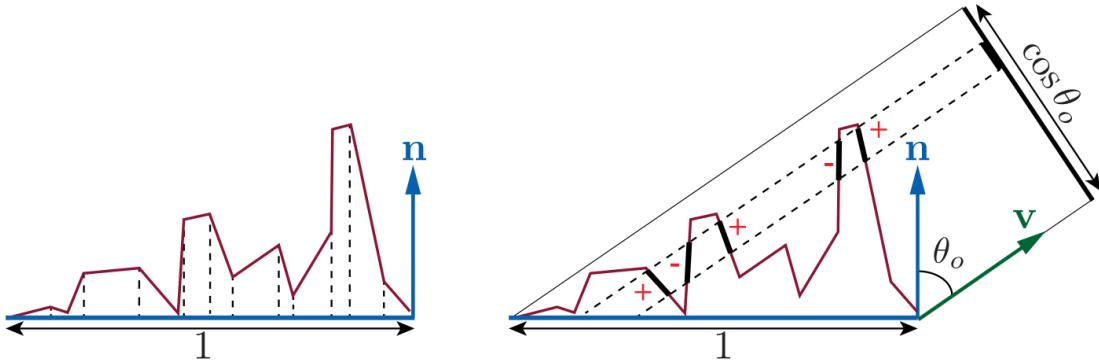


Figure 6.7: Illustrazione del significato della Normal Distribution Function. Nota a destra, che nell'integrale della NDF proiettato, i contributi occlusi si presentano sempre in coppia, uguali e con segno opposto, dunque non contribuiscono all'integrale. Immagine da [AHH18]

Allora l'area proiettata sulla macrosuperficie è unitaria

$$\int_{\mathcal{H}^2(\hat{n})} D(\vec{p}, \hat{m}) \langle \hat{n}, \hat{m} \rangle d\hat{m} = 1 \quad (6.54)$$

Più in generale, l'area della microsuperficie proiettata su un piano perpendicolare ad una direzione  $\hat{\omega}$  (tipicamente quella dell'osservatore), pari all'*area con segno* della microsuperficie (Figura 6.7) diviso l'area della macrosuperficie, è pari al coseno dell'angolo tra direzione e normale della macrosuperficie

$$\int_{\mathcal{H}^2(\hat{n})} D(\vec{p}, \hat{m}) \langle \hat{\omega}, \hat{m} \rangle d\hat{m} = \langle \hat{n}, \hat{\omega} \rangle \quad (6.55)$$

I *Masked microfacets* sono contributi che si semplificano nell'integrale, dunque esso è pari all'integrale dei *microfacets visibili*. Possiamo modellare ciò matematicamente definendo una *Masking Function* o *Mono-Static Visibility Function* (MVF), la quale, data normale microscopica  $\hat{m}$  e direzione  $\hat{\omega}$ , definisce la percentuale aggregata di area di tutti i microfacets con normale  $\hat{m}$ , visibile da  $\hat{\omega}$ , fratto l'area aggregata totale di tutti i microfacets con normale  $\hat{m}$ . Essa rappresenta dunque la *probabilità* di un microfacet con normale  $\hat{m}$  di essere visibile da  $\hat{\omega}$ ,

$$G_1(\vec{p}, \hat{m}, \hat{\omega}) = \Pr(\hat{\omega}) \quad (6.56)$$

Tale funzione  $\in [0, 1]$  e  $G_1(\vec{p}, \hat{m}, \hat{\omega}) = 0$ , se  $\langle \hat{n}, \hat{\omega} \rangle \langle \hat{m}, \hat{\omega} \rangle \leq 0$

Dunque, per definizione,  $G_1(\vec{p}, \hat{m}, \hat{\omega}_o) D(\vec{p}, \hat{m})$  è la distribuzione dei microfacets visibili, tale che

$$\int_{\mathcal{H}^2(\hat{n})} G_1(\vec{p}, \hat{m}, \hat{\omega}_o) D(\vec{p}, \hat{m}) \langle \hat{m}, \hat{\omega}_o \rangle^+ d\hat{m} = \langle \hat{n}, \hat{\omega}_o \rangle \quad (6.57)$$

dove  $x^+ = \max(0, x)$

Anche la funzione per modellare lo *shadowing* è una *Mono-Static Visibility Function*, infatti gode delle stesse proprietà della Masking Function, sostituendo  $\hat{\omega}_i$  ad  $\hat{\omega}_o$ . Queste due funzioni sono combinate in qualche modo per ottenere la cosiddetta *Joint Shadowing-Masking Function* o *Bi-Static Visibility Function* (BVF)  $G_2$ , definita come percentuale di area aggregata di microfacets

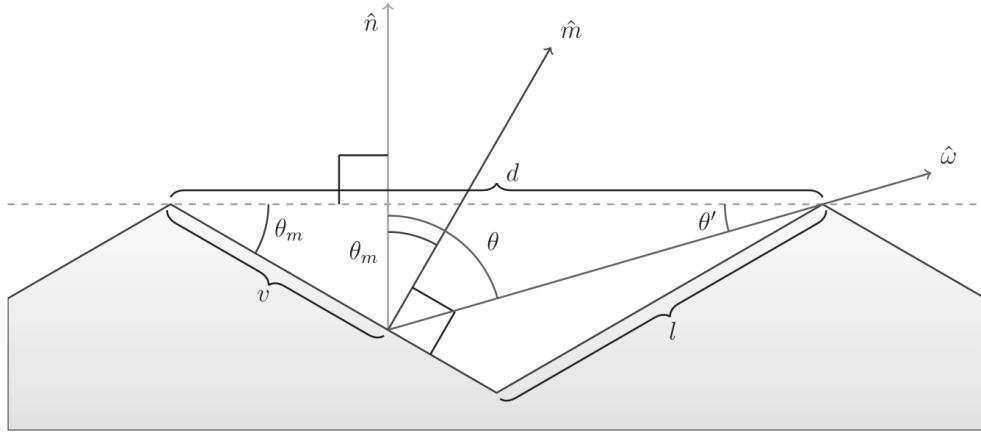


Figure 6.8: Illustrazione della geometria delle V-Cavities. Immagine da [Peg16]

con normale  $\hat{m}$  simultaneamente visibile dalle due direzioni  $\hat{\omega}_o$  e  $\hat{\omega}_i$ , frutto l'area totale di tutti i microfacets con normale  $\hat{m}$ . In termini di probabilità,

$$G_2(\vec{p}, \hat{m}, \hat{\omega}_o, \hat{\omega}_i) = \Pr(\hat{\omega}_o \cap \hat{\omega}_i) \quad (6.58)$$

Se si assume che shadowing e masking sono incorrelati, il che può essere buona approssimazione solo per direzioni con differenza di angolo azimuthale considerevole, allora

$$G_2(\vec{p}, \hat{m}, \hat{\omega}_o, \hat{\omega}_i) = \Pr(\hat{\omega}_o) \Pr(\hat{\omega}_i) = G_1(\vec{p}, \hat{m}, \hat{\omega}_o) G_1(\vec{p}, \hat{m}, \hat{\omega}_i) \quad (6.59)$$

il che tende a provocare over darkening della superficie<sup>20</sup>. Per essere fisicamente plausibile, la probabilità per un microfacet di essere visibile da due direzioni è minore, al più uguale se l'angolo azimuthale tra  $\hat{\omega}_i$  e  $\hat{\omega}_o$  è zero, a quella di essere visibile da una singola direzione

$$G_2(\vec{p}, \hat{m}, \hat{\omega}_o, \hat{\omega}_i) \in [0, \min(G_1(\vec{p}, \hat{m}, \hat{\omega}_o), G_1(\vec{p}, \hat{m}, \hat{\omega}_i))] \quad (6.60)$$

Inoltre la microstruttura è visibile solo dalla emisfera la cui normale macroscopica forma un angolo acuto, da entrambe le direzioni

$$G_2(\vec{p}, \hat{m}, \hat{\omega}_o, \hat{\omega}_i) = 0, \text{ se } \langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{m}, \hat{\omega}_o \rangle \leq 0 \quad \langle \hat{n}, \hat{\omega}_i \rangle \langle \hat{m}, \hat{\omega}_i \rangle \leq 0 \quad (6.61)$$

Infine soddisfa la proprietà di reciprocità  $G_2(\vec{p}, \hat{m}, \hat{\omega}_o, \hat{\omega}_i) = G_2(\vec{p}, \hat{m}, \hat{\omega}_i, \hat{\omega}_o)$ .

Analizziamo brevemente un modello di visibilità, la cui dimostrazione completa si rimanda a [Peg16]: *V-Cavity Function*.

### V-Cavity Function

Assumiamo che le micro strutture si arrangino parallele le une alle altre in due direzioni  $\hat{m}_\parallel$  e  $\hat{m}_\perp$ . La Mono-Static Visibility  $v$  è espressa in termini di rapporto tra lunghezza visibile e lunghezza totale dei lati della cavità

$$\begin{aligned} v(\vec{p}, \hat{m}, \hat{\omega}) &= \frac{v}{l} = \frac{d}{l} \frac{\sin \theta'}{\sin(\pi - \theta_m - \theta')} \\ &= \frac{2 \langle \hat{n}, \hat{m} \rangle \langle \hat{n}, \hat{\omega} \rangle}{\langle \hat{m}, \hat{\omega} \rangle} \end{aligned} \quad (6.62)$$

<sup>20</sup>infatti se  $\hat{\omega}_o = \hat{\omega}_i$ , allora dovremmo ottenere  $G_1$ . Invece, con tale metodo, otteniamo  $G_1^2$

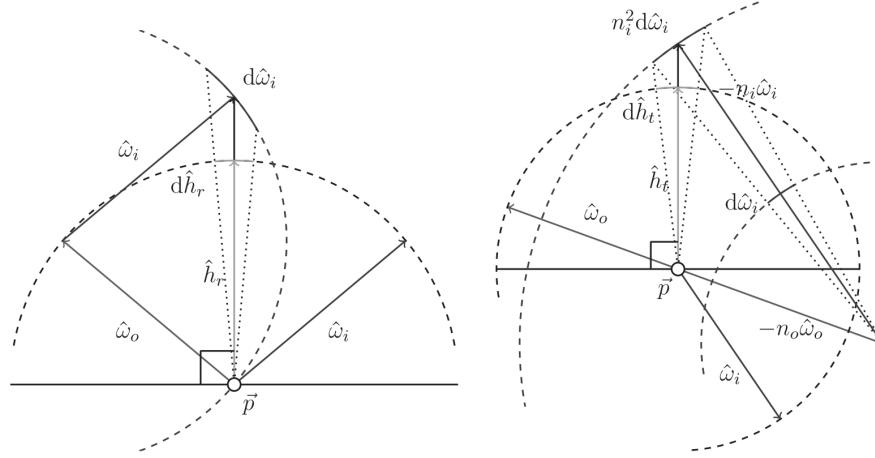


Figure 6.9: Illustrazione della costruzione dei Jacobiani per trasformazioni in seguito a riflessione e trasmissione. Immagine da [Peg16]

La quale, per soddisfare tutte le proprietà di una MVF, si definisce

$$V(\vec{p}, \hat{m}, \hat{\omega}) = \min\{1, H(\langle \hat{n}, \hat{\omega} \rangle \langle \hat{m}, \hat{\omega} \rangle) v(\vec{p}, \hat{m}, \hat{\omega})\} \quad (6.63)$$

E si definisce un metodo per combinare shadowing e masking supponendo masking e shadowing perfettamente correlati, il che è buona approssimazione per differenza tra angoli azimuthali di direzione entrante e uscente piccola, nella Bi-Static Visibility<sup>22</sup>

$$\frac{G(\vec{p}, \hat{m}, \hat{\omega}_o, \hat{\omega}_i)}{4\langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle} = \quad (6.64)$$

$$\min \left\{ \frac{1}{4\langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle}, \frac{H(\langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{m}, \hat{\omega}_o \rangle) 2\langle \hat{n}, \hat{m} \rangle}{4\langle \hat{m}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle}, \frac{H(\langle \hat{n}, \hat{\omega}_i \rangle \langle \hat{m}, \hat{\omega}_i \rangle) 2\langle \hat{n}, \hat{m} \rangle}{4\langle \hat{m}, \hat{\omega}_i \rangle \langle \hat{n}, \hat{\omega}_o \rangle} \right\} \quad (6.65)$$

Dove, se  $\hat{m} = \hat{h}_r$

$$\frac{G(\vec{p}, \hat{h}_r, \hat{\omega}_o, \hat{\omega}_i)}{4\langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle} = \min \left\{ \frac{1}{4 \cos \theta_o \cos \theta_i}, \frac{2 \cos \theta_h}{4 \cos \theta_d \cos \theta_i}, \frac{2 \cos \theta_h}{4 \cos \theta_d \cos \theta_o} \right\} \quad (6.66)$$

dove  $\cos \theta_d$  coseno dell'angolo formato tra  $\hat{\omega}_{o/i}$  e  $\hat{h}_r$ , vedi Figura 6.10.

### BRDF: Modello di Cook-Torrance

**Riflessione:** Assumendo che i microfacets siano superfici perfettamente speculari, i contributi che arrivano all'osservatore sono quelli dei microfacets che hanno come normale  $\hat{m}$  il versore "a metà strada" tra  $\hat{\omega}_o$  e  $\hat{\omega}_i$

$$\hat{h}_r = \frac{\hat{\omega}_o + \hat{\omega}_i}{\|\hat{\omega}_o + \hat{\omega}_i\|} \quad (6.67)$$

affinchè  $\hat{\omega}_o = \tau_r(\hat{h}_r, \hat{\omega}_i)$ ,  $\hat{\omega}_i = \tau_r(\hat{h}_r, \hat{\omega}_o)$

<sup>21</sup>Legge dei seni

<sup>22</sup>Si veda il modello di Cook-Torrance per capire il perchè è stata espressa così

## 6.1. MODELLAZIONE DI SUPERFICI

Inoltre, supponendo che tutti i punti di tutte le microsuperfici abbiano identica micro-BRDF/BTDF, si può dimostrare che la BRDF/BTDF macroscopica è pari a [Peg16]

$$f_{r|t}(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \quad (6.68)$$

$$\frac{1}{|\langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle|} \int_{\mathcal{H}^2(\hat{n})} D(\vec{p}, \hat{m}) G(\vec{p}, \hat{m}, \hat{\omega}_o, \hat{\omega}_i) f_\mu(\hat{m}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{m}, \hat{\omega}_o \rangle \langle \hat{m}, \hat{\omega}_i \rangle| d\hat{m} \quad (6.69)$$

Dove, per le nostre assunzioni, la micro-BRDF è quella delle superfici speculari

$$f_\mu(\hat{m}, \hat{\omega}_o, \hat{\omega}_i) = \frac{\delta(\hat{\omega}_i - \tau_r(\hat{m}, \hat{\omega}_o))}{\langle \hat{m}, \hat{\omega}_i \rangle} F_r(\langle \hat{m}, \hat{\omega}_o \rangle) \quad (6.70)$$

Si può dimostrare ([Peg16]) che tali assunzioni conducono alla seguente BRDF, diffusissima nelle applicazioni moderne

$$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \frac{D(\vec{p}, \hat{h}_r) G(\vec{p}, \hat{h}_r, \hat{\omega}_o, \hat{\omega}_i)}{|\langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle|} \frac{F_r(\langle \hat{h}_r, \hat{\omega}_o \rangle)}{4} \quad (6.71)$$

Seguendo gli stessi passaggi, possiamo ottenere espressioni analoghe per la BTDF

$$\hat{h}_t = \text{sgn} \left( 1 - \frac{\eta_i}{\eta_t} \right) \frac{\eta_t \hat{\omega}_o + \eta_i \hat{\omega}_i}{\|\eta_t \hat{\omega}_o + \eta_i \hat{\omega}_i\|} \quad (6.72)$$

$$f_\mu(\hat{m}, \hat{\omega}_o, \hat{\omega}_i) = \frac{\delta(\hat{h}_t - \hat{m})}{-\langle \hat{m}, \hat{\omega}_i \rangle} \frac{-\eta_t^2 \langle \hat{h}_t, \hat{\omega}_i \rangle T_{\frac{\eta_i}{\eta_t}}(\langle \hat{m}, \hat{\omega}_o \rangle)}{(\eta_t \langle \hat{h}_t, \hat{\omega}_o \rangle + \eta_i \langle \hat{h}_t, \hat{\omega}_i \rangle)^2} \quad (6.73)$$

$$f_t(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \frac{D(\vec{p}, \hat{h}_t) G(\vec{p}, \hat{h}_t, \hat{\omega}_o, \hat{\omega}_i)}{|\langle \hat{n}, \hat{\omega}_o \rangle \langle \hat{n}, \hat{\omega}_i \rangle|} \frac{-\eta_t^2 \langle \hat{h}_t, \hat{\omega}_i \rangle T_{\frac{\eta_i}{\eta_t}}(\langle \hat{h}_t, \hat{\omega}_o \rangle)}{(\eta_t \langle \hat{h}_t, \hat{\omega}_o \rangle + \eta_i \langle \hat{h}_t, \hat{\omega}_i \rangle)^2} \quad (6.74)$$

Si analizzano di seguito due alternative di NDF, anello mancante del modello presentato. *Beckmann NDF* e *Trowbridge-Reitz NDF*.

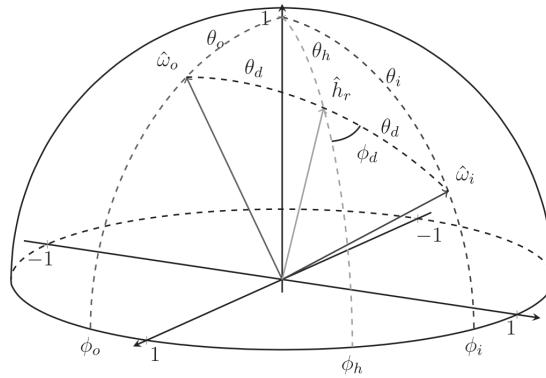


Figure 6.10: Illustrazione della parametrizzazione della BRDF in termini di angoli. Immagine da [Peg16]

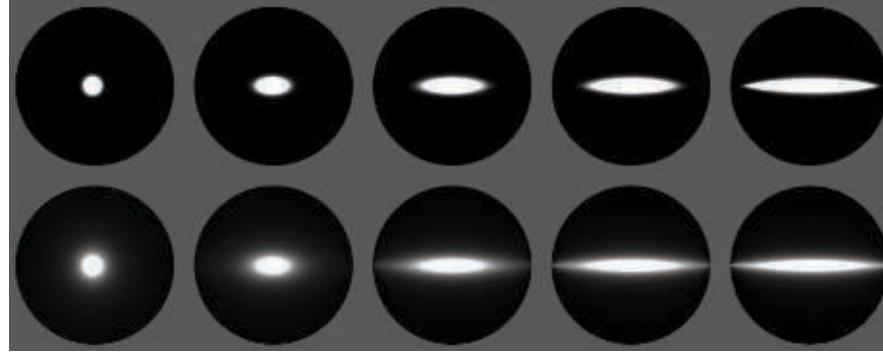


Figure 6.11: Confronto tra NDFs anisotropiche (progressivamente verso destra) e isotropiche, dove riga in alto usa la Beckmann NDF e riga in basso la Trowbridge-Reitz NDF. Immagine da [AHH18].

### Beckmann NDF

Una densità di distribuzione di normali utilizzata è una guassiana bidimensionale *isotropica* modificata, chiamata *Beckmann-Spizzichino NDF* ed è definita come

$$D(\vec{p}, \hat{m}) = \frac{e^{-\frac{\tan^2 \theta_m}{\alpha_b^2}}}{\pi \alpha_b^2 \cos^4 \theta_m} = \frac{e^{-\frac{1 - \langle \hat{n}, \hat{m} \rangle^2}{\alpha_b^2 (\hat{n}, \hat{m})}}}{\pi \alpha_b^2 \langle \hat{n}, \hat{m} \rangle^4} \quad (6.75)$$

dove  $\alpha_b$  parametro che controlla la roughness della superficie. Tale NDF è isotropica, cioè non dipendente dall'angolo azimuthale di  $\hat{m}$ , e dunque rotazionalmente simmetrica (vedi Figura 6.11).

Per campionare secondo la densità di distribuzione, ottenuta dalla NDF *cosine weighted*<sup>23</sup>, tramite Inverse Transform Sampling, sia  $\mu = \langle \hat{n}, \hat{m} \rangle$

$$\begin{aligned} \int_{\Omega_m} D(\vec{p}, \hat{m}) \langle \hat{n}, \hat{m} \rangle d\hat{m} &= \int_0^{\varphi_m} \frac{1}{2\pi} d\varphi \int_{\cos \theta_m}^1 2 \frac{e^{\frac{1-\mu^2}{\alpha_b^2 \mu^2}}}{\alpha_b^2 \mu^3} d\mu \\ &= \left[ \frac{\varphi}{2\pi} \right]_{\varphi=0}^{\varphi_m} \left[ e^{\frac{1-\mu^2}{\alpha_b^2 \mu^2}} \right]_{\cos \theta_m}^1 \\ &= \frac{\varphi_m}{2\pi} \left( 1 - e^{-\frac{1-\cos^2 \theta_m}{\alpha_b^2 \cos^2 \theta_m}} \right) \end{aligned} \quad (6.76)$$

siano  $\xi_\varphi, \xi_\theta \sim \mathcal{U}(0, 1)$ , allora Inverse Transform Sampling che permette Importance Sampling rispetto alla NDF è

$$\xi_\varphi = \frac{\varphi_m}{2\pi} \rightarrow \varphi_m = 2\pi \xi_\varphi \quad (6.77)$$

$$\xi_\theta = 1 - e^{-\frac{1-\cos^2 \theta_m}{\alpha_b^2 \cos^2 \theta_m}} \rightarrow \theta_m = \arccos \left( \frac{1}{\sqrt{1 - \alpha_b^2 \ln(1 - \xi_\theta)}} \right) \quad (6.78)$$

<sup>23</sup>Come detto prima, la NDF, pesata per il coseno tra normale microscopica e normale macroscopica, ha massa unitaria. Equazione 6.54

**Trowbridge-Reitz NDF**

La Trowbridge-Reitz NDF, anche detta GGX NDF, è una NDF isotropica che modella la superficie microscopica come composta da un gran numero di dossi elisoidali, con espressione

$$D(\vec{p}, \hat{m}) = \frac{\alpha_g^2}{\pi(\alpha_g^2 + \tan^2 \theta_m)^2 \cos^4 \theta_m} = \frac{\alpha_g}{\pi(1 + (\alpha_g^2 - 1)\langle \hat{n}, \hat{m} \rangle)^2} \quad (6.79)$$

Per la quale la sua PDF normalizzata può essere integrata ed espressa in forma separabile. (sia  $\mu = \cos \theta_m = \langle \hat{n}, \hat{m} \rangle$ )

$$\begin{aligned} \int_{\Omega_m} D(\vec{p}, \hat{m}) &= \int_0^{\varphi_m} \frac{1}{2\pi} d\varphi \int_{\cos \theta_m}^1 \frac{2\alpha_g^2}{(1 + (\alpha_g^2 - 1)\mu)^2} \mu d\mu \\ &= \left[ \frac{\varphi}{2\pi} \right]_0^{\varphi_m} \left[ \frac{\alpha_g^2}{\alpha_g^2 - 1} \frac{-1}{1 + (\alpha_g^2 - 1)\mu^2} \right]_{\cos \theta_m}^1 \\ &= \frac{\varphi_m}{2\pi} \frac{1}{\alpha_g^2 - 1} \left( \frac{\alpha_g^2}{1 + (\alpha_g^2 - 1)\mu^2} - 1 \right) \\ &= \frac{\varphi_m}{2\pi} \frac{1}{\alpha_g^2 - 1} \frac{\alpha_g^2 - 1 - (\alpha_g^2 - 1)\cos^2 \theta_m}{1 + (\alpha_g^2 - 1)\cos^2 \theta_m} \\ &= \frac{\varphi_m}{2\pi} \frac{1 - \cos^2 \theta_m}{1 + (\alpha_g^2 - 1)\cos^2 \theta_m} \end{aligned} \quad (6.80)$$

Dunque applicando Inverse Transform Sampling per poter campionare la BRDF secondo direzioni con probabilità proporzionali alla NDF (Importance Sampling)

$$\xi_\varphi = \frac{\varphi_m}{2\pi} \rightarrow \varphi_m = 2\pi\xi_\varphi \quad (6.81)$$

$$\xi_\theta = \frac{1 - \cos^2 \theta_m}{1 + (\alpha_g^2 - 1)\cos^2 \theta_m} \rightarrow \theta_m = \arccos \left( \sqrt{\frac{1 - \xi_\theta}{1 - \xi_\theta + \xi_\theta \alpha_g^2}} \right) \quad (6.82)$$

La generalizzazione anisotropica della Trowbridge-Reitz NDF, anche chiamata *ATR NDF*, ha espressione

$$D(\vec{p}, \hat{m}) = \frac{1}{\pi \alpha_u \alpha_v \left( \cos^2 \theta_m + \frac{\sin^2 \theta_m \cos^2 \varphi_m}{\alpha_u^2} \frac{\sin^2 \theta_m \sin^2 \varphi_m}{\alpha_v^2} \right)^2} \quad (6.83)$$

Il campionamento secondo la PDF  $D(\vec{p}, \hat{m})\langle \hat{n}, \hat{m} \rangle$  procede in modo analogo a Ward NDF. Sia  $a = \frac{\cos^2 \varphi}{\alpha_u^2} + \frac{\sin^2 \varphi}{\alpha_v^2}$

$$\begin{aligned} \int_{\Omega_m} D(\vec{p}, \hat{m})\langle \hat{n}, \hat{m} \rangle d\hat{m} &= \int_0^{\varphi_m} \int_{\cos \theta_m}^1 \frac{\mu}{\pi \alpha_u \alpha_v (\mu^2 + (1 - \mu^2)a)^2} d\mu d\varphi \\ &= \int_0^{\varphi_m} \frac{1}{\pi \alpha_u \alpha_v} \left[ \frac{1}{2(a - 1)(a - \mu^2 a + \mu^2)} \right]_{\cos \theta_m}^1 d\varphi \\ &= \int_0^{\varphi_m} \frac{1}{2\pi \alpha_u \alpha_v} \left( \frac{1 - \cos^2 \theta_m}{a - \cos^2 \theta_m a + \cos^2 \theta_m} \right) d\varphi \\ &:= p(\varphi, \theta_m) \end{aligned} \quad (6.84)$$

Da cui la PDF marginale di  $\varphi_m$  è

$$p(\varphi, \pi/2) = \frac{1}{2\pi\alpha_u\alpha_v a} \quad (6.85)$$

Da cui segue la sua CDF marginale

$$\begin{aligned} \int_0^{\varphi_m} p(\varphi, \pi/2) d\varphi &= \int_0^{\varphi_m} \frac{1}{2\pi\alpha_u\alpha_v} \frac{1}{\frac{\cos^2 \varphi}{\alpha_u^2} + \frac{\sin^2 \varphi}{\alpha_v^2}} d\varphi \\ &= \frac{1}{2\pi\alpha_u\alpha_v} \left[ \alpha_u\alpha_v \arctan \left( \frac{\alpha_u}{\alpha_v} \tan \varphi \right) \right]_{\varphi=0}^{\varphi_m} \\ &= \frac{1}{2\pi} \arctan \left( \frac{\alpha_u}{\alpha_v} \tan \varphi_m \right) \end{aligned} \quad (6.86)$$

Da cui la CDF condizionale di  $\theta_m$  dato  $\varphi_m$  è pari a

$$\frac{p(\varphi, \theta_m)}{p(\varphi, \pi/2)} = \frac{1}{1 + \frac{\cos^2 \theta_m}{(1 - \cos^2 \theta_m) \left( \frac{\cos^2 \varphi}{\alpha_u^2} + \frac{\sin^2 \varphi}{\alpha_v^2} \right)}} \quad (6.87)$$

Da cui Inverse Transform Sampling può essere effettuato per eseguire Importance Sampling della BRDF secondo la NDF tramite

$$\begin{aligned} \xi_\varphi &= \frac{1}{2\pi} \arctan \left( \frac{\alpha_u}{\alpha_v} \tan \varphi_m \right) \\ \rightarrow \varphi_m &= \arctan2(\alpha_v \sin(2\pi\xi_\varphi), \alpha_u \cos(2\pi\xi_\varphi)) \end{aligned} \quad (6.88)$$

$$\begin{aligned} \xi_\theta &= \frac{p(\varphi, \theta_m)}{p(\varphi, \pi/2)} = \frac{1}{1 + \frac{\cos^2 \theta_m}{(1 - \cos^2 \theta_m) \left( \frac{\cos^2 \varphi}{\alpha_u^2} + \frac{\sin^2 \varphi}{\alpha_v^2} \right)}} \\ \rightarrow \theta_m &= \arctan \left( \sqrt{\frac{\xi_\theta}{(1 - \xi_\theta) \left( \frac{\cos^2 \varphi}{\alpha_u^2} + \frac{\sin^2 \varphi}{\alpha_v^2} \right)}} \right) \end{aligned} \quad (6.89)$$

Come nota Finale per i modelli basati su microfacet theory seguienti il Cook-Torrance BRDF model, l'importance sampling guidato dalla NDF normalizzata è sicuramente meno efficace del campionamento della distribuzione di normali visibili [PJH23]. Qui ci limitiamo a campionare in modo proporzionale alla NDF per semplicità.

### Oren-Nayar BRDF

Piuttosto che considerare la microgeometry come composta da facce perfettamente speculari, la *Oren-Nayar BRDF* assume microstrutture *Lambertiane* affinchè possa modellare materiali che presentano riflessione diffusa con back-scattering (sabbia, carta, plastica, ...). Dunque  $f_\mu(\hat{m}, \hat{\omega}_o, \hat{\omega}_i) = 1/\pi$  Modellando la distribuzione di microstrutture come una Gaussiana, ed utilizzando la V-Cavity Function, si ottiene un integrale che, approssimato per tener conto solo degli scattering events di primo e secondo ordine, si può dimostrare che si ottengono le seguenti

BRDF, che descrivono rispettivamente scattering events di primo e secondo ordine

$$f_r^1 = \frac{1}{\pi} \left( 1 - c_1 + c_2 c'_2 \tan \beta + c_3 c'_3 \tan \left( \frac{\alpha + \beta}{2} \right) \right) \quad (6.90)$$

$$f_r^2 = \frac{1}{\pi} c_4 \left( 1 - \gamma \left( \frac{2\beta}{\pi} \right)^2 \right) \quad (6.91)$$

dove, sia  $\sigma$  parametro di roughness della superficie, [Peg16]

$$\begin{aligned} \alpha &= \max\{\theta_o, \theta_i\} & c'_2 &= \gamma \times \begin{cases} \sin \alpha & \text{se } \gamma \geq 0 \\ c''_2 & \text{altrimenti} \end{cases} \\ \beta &= \min\{\theta_o, \theta_i\} & c'_3 &= (1 - |\gamma|) \left( \frac{4\alpha\beta}{\pi^2} \right)^2 \\ \gamma &= \cos(\varphi_o - \varphi_i) & c''_2 &= \sin \alpha - \left( \frac{2\beta}{\pi} \right)^3 \\ c_1 &= \frac{0.5\sigma^2}{\sigma^2+0.33} & c_2 &= \frac{0.45\sigma^2}{\sigma^2+0.09} \\ c_3 &= \frac{0.125\sigma^2}{\sigma^2+0.09} & c_4 &= \frac{0.17\sigma^2}{\sigma^2+0.13} \end{aligned} \quad (6.92)$$

Il modello è infine definito come combinazione lineare delle distribuzioni di primo e secondo ordine pesate sul parametro albedo  $\rho$

$$f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) = \rho f_r^1(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) + \rho^2 f_r^2(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) \quad (6.93)$$

Il suo campionamento può far uso della stessa metodologia utilizzata per la Lambertian BRDF

## 6.2 Sorgenti Luminose

Gli oggetti presenti della scena sono illuminati da sorgenti luminose, che le rende visibili. In quanto ci si è prefissati come obiettivo Physically Based Rendering, i modelli che analizziamo sono specificati attraverso la radiometria (o fotometria). Ad esempio, una sorgente potrebbe essere specificata attraverso uno spettro di emissione, oppure da una temperatura, se si assume che tale sorgente sia un corpo nero, (Equazione 4.10). Si noti che tali modelli non permettono il pieno controllo artistico della scena come modificare il falloff rate, oppure posizionare una sorgente che non proietti ombre, per il principio di conservazione dell'energia.

In scene con molte sorgenti luminose, non possiamo tener conto di tutti i loro contributi, e utilizziamo un approccio stocastico, secondo il quale dati opportuni pesi, possiamo considerarne soltanto un piccolo sottoinsieme, rappresentante le sorgenti più significative, campionate con una PMF uniforme, oppure proporzionale al flusso radiante emesso dalla sorgente<sup>24</sup>.

Tipi differenti di sorgenti luminose devono essere gestite in modo differente, in modo tale che il campionamento di una direzione per la Rendering Equation (Equazione 6.30) sia gestito in modo appropriato. [PJH23] riconosce quattro tipi principali di sorgenti luminose

- *DeltaPosition*: Sorgenti luminose occupanti un solo punto nello spazio. Ciò significa che ogni altro punto dello spazio è illuminato da una sola direzione da tali sorgenti, e dunque sono descrivibili da una Distribuzione di Dirac
- *DeltaDirection*: Sorgenti luminose che emettono radianza lungo un'unica direzione<sup>25</sup>.
- *Area*: Sorgenti luminose che emettono radianza da una superficie geometrica

<sup>24</sup>altri metodi non citati includono il campionamento delle sorgenti luminose secondo la loro visibilità tramite strutture dati spaziali come una BVH (Bounding Volume Hierarchy) [PJH23], oppure tramite la tecnica di Portal Culling [AHH18]

- *Infinite*: Sorgenti luminose all’infinito, che non possiedono geometria ad esse associata e forniscono radianza ai raggi che fuggono dalla scena

<sup>26</sup>Nota che una sorgente luminosa del tipo DeltaDirection può anche essere contemporaneamente del tipo DeltaPosition. Si pensi ad un raggio collimato Talvolta, campionare una direzione secondo la distribuzione della sorgente luminosa considerata può essere più efficace per ottenere un tasso di convergenza più rapido, in quanto, data una BSDF, eg. Lambertiana, soltanto un piccolo cono di essa potrebbe essere illuminato da sorgenti luminose e dunque contribuire in modo significativo alla radianza computata. Vedi Capitolo 1 per Multiple Importance Sampling (Importance Sampling in cui i pesi sono spatially varying, e il campionamento è effettuato secondo multiple PDF, eg, BSDF e Light Directional Distribution).

Dunque, per ciascuna sorgente luminosa, specifichiamo anche la PDF proporzionale al suo spettro di emissione.

Infine, ci sono casi particolari da gestire che dipendono dal tipo di luce, ad esempio

- Se un raggio interseca una sorgente luminosa del tipo Area, la radianza uscente deve essere calcolata esclusivamente in base ai parametri della sorgente
- Per i raggi che fuggono dalla scena, tutte<sup>27</sup> le sorgenti del tipo Infinite devono essere considerate per potervi calcolare la radianza totale emessa data la direzione di provenienza del raggio

Si noti infine, che le luci di seguito specificate sono analizzate in un sistema di coordinate il quale posiziona la sorgente luminosa nell’origine con un orientamento particolare. Tale sistema di coordinate è ricavato dalla matrice di trasformazione fornita come parametro nella specifica della sorgente, in seguito tipicamente invertita per poter applicare i calcoli eseguiti nel Render Space.

### 6.2.1 Modelli di sorgenti luminose

Le sorgenti luminose analizzate sono:

- DeltaPosition: *Point Omni Lights, Spotlights*
- DeltaDirection: *Directional Lights*
- Area: *Diffuse Area Lights*
- Infinite: *Infinite Uniform Lights*

#### Point Omni Lights

Una Point Omni Light, o Isotropic Point Light, è una sorgente luminosa puntuale che emette radiazioni con intensità radiante  $I_e$  costante in tutte le direzioni. La radianza in questo caso non è appropriata per descrivere la sorgente luminosa, in quanto essa descrive flusso radiante per unità di angolo solido per unità di area. Ma tale luce puntuale non possiede area. Trascuriamo, per agio di implementazione, tale dettaglio fisico e lavoriamo con la radianza, la quale, data una isotropic point light in posizione  $\vec{p}$ , può essere calcolata come

$$L_e = \frac{I_e}{\|\vec{q} - \vec{p}\|^2} = \frac{I_e}{r^2} \quad (6.94)$$

Tale formula possiede due problemi, oltre all’inaccuratezza fisica citata prima: ha una singolarità in zero, e tende a zero per  $r \rightarrow \infty$  [AHH18]. Il primo problema può essere risolto aggiungendo un  $\epsilon$  piccolo, arbitrario, a denominatore, mentre il secondo può essere risolto

---

<sup>27</sup>o le più significative, come detto prima

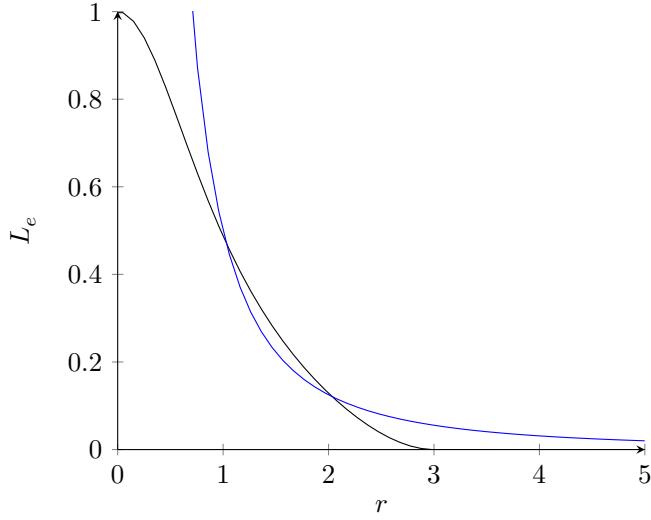


Figure 6.12: Confronto tra Normale Inverse Square Law e Formula con  $\epsilon$  aggiunto e moltiplicazione per windowing function

moltiplicando l'equazione risultante per una *windowing function*. Vedi Figura 6.12

$$f_{win}(r) = \left(1 - \left(\frac{r}{r_{max}}\right)^4\right)^{+2} \quad (6.95)$$

dove  $+2$  significa  $(\max(0, x))^2$

Se tale sorgente luminosa è campionata tramite PMF, la direzione che essa restituisce, in modo deterministico, in campionamento per la Rendering Equation è la direzione che punta verso la sua posizione, in quanto essa è di tipo DeltaPosition. Campionamento di direzione in ogni altro modo ritorna zero per via della distribuzione di Dirac [PJH23].

Al fine del campionamento delle sorgenti proporzionale alla potenza emessa, il flusso radiante emesso è pari a

$$\Phi_e = \int_{S^2} I_e d\hat{\omega} = 4\pi I_e \quad (6.96)$$

### Spotlight

Variazione di Point Lights (tipo DeltaPosition) le quali emettono radiazione in un cono di direzioni. Il loro Light Space possiede come origine il vertice del cono, e l'asse  $+z$  come direzione dell'asse. In particolare, una Spotlight è specificata geometricamente da due coni. Oggetti dentro il cono interno ricevono illuminazione massima, proporzionale al quadrato della distanza con stessi accorgimenti specificati per la Isotropic Point Light, oggetti al di fuori del cono più ampio non ricevono alcuna illuminazione, e oggetti nel volume tra i due coni ricevono illuminazione progressivamente minore man mano che ci si avvicina all'esterno del cono maggiore.

Tale coppia di coni è definita dagli angoli  $\theta_p$  (detto penumbra angle o falloff start) e  $\theta_u$  (detto umbra angle o falloff end).

In quanto della stessa categoria delle Isotropic Point Light, la loro densità di distribuzione di direzioni è una distribuzione di Dirac, dunque segue la stessa logica citata in precedenza.

Al fine di calcolare la radianza per la spotlight, si segue lo stesso procedimento precedente, con la differenza che l'intensità radiante non è più costante, in particolare

$$I_e(\hat{\omega}) = \begin{cases} I_{e0} & \text{se } \langle \hat{\omega}, \hat{z} \rangle \geq \cos \theta_p \\ \text{smoothstep}(\langle \hat{\omega}, \hat{z} \rangle, \cos \theta_u, \cos \theta_p) & \text{se } \langle \hat{\omega}, \hat{z} \rangle \in (\cos \theta_u, \cos \theta_p) \\ 0 & \text{se } \langle \hat{\omega}, \hat{z} \rangle \leq \cos \theta_u \end{cases} \quad (6.97)$$

Dove una funzione smoothstep è una famiglia di funzioni atta a compiere interpolazione sigmoid-like tra 0 e 1, dipendente dai parametri input  $x$ , left edge  $a$ , right edge  $b$ . Di seguito è mostrata una funzione smoothstep implementata con interpolazione di Hermite cubica (si rimanda a [Peg16])

$$\text{smoothstep}(x, a, b) = \begin{cases} 0 & \text{se } x \leq a \\ y^2(3 - 2y) & \text{se } x \in (a, b), \text{ con } y = \frac{x - a}{b - a} \\ 1 & \text{se } x \geq b \end{cases} \quad (6.98)$$

Il calcolo della potenza emessa dalla sorgente, integriamo l'intensità radiante nel dominio con contributi non nulli. Si supponga  $I_{e0} = 1\text{W}/\text{sr}$

$$\begin{aligned} \Phi_e &= \int_{\Omega} I_e(\hat{\omega}) d\hat{\omega} \\ &= \int_0^{2\pi} d\varphi \left( \int_0^{\theta_p} \sin \theta d\theta - \int_{\theta_p}^{\theta_u} \text{smoothstep}(\cos \theta, \cos \theta_u, \cos \theta_p) \sin \theta d\theta \right) \\ &= 2\pi \left( 1 - \cos \theta_p + \frac{\cos \theta_p - \cos \theta_u}{2} \right) \end{aligned} \quad (6.99)$$

### Directional Light

Modella una point light posizionata all'infinito, dunque illuminante ogni punto della scena da una direzione prefissata. La radianza incidente in un punto  $\vec{p}$  per via di una directional light può essere descritta dalla distribuzione di Dirac

$$L_i(\vec{p}, \hat{\omega}) = L_e \delta(\hat{\omega} - \hat{\omega}_l) \quad (6.100)$$

dove  $\hat{\omega}_l$  direzione della luce.

Tale sorgente luminosa emette flusso radiante proporzionale alle dimensioni della scena. Dunque, sia la scena racchiusa da una sfera di raggio  $R$ , e sia la radianza emessa dalla sorgente luminosa  $L_e$ , la potenza emessa  $\Phi_e$  è pari a alla potenza irradiata nel disco di raggio  $R$ , superficie perpendicolare irradiata dalla luce emessa

$$\Phi_e = \int_A \int_{\Omega} L_e \delta(\hat{\omega} - \hat{\omega}_l) d\hat{\omega} dA = \pi R^2 L_e \quad (6.101)$$

### Diffuse Area Light

Sorgente Luminosa definita da una superficie ed una distribuzione di direzioni in cui essa emette radianza. Il calcolo della radianza emessa generalmente il calcolo di un integrale su un angolo solido di direzioni rispetto al punto di osservazione, il quale spesso non può essere risolto in modo analitico, dovendo dunque ricorrere al Metodo di Monte Carlo.

Assumo che la distribuzione di direzioni in cui la sorgente luminosa emette sia uniforme nella emisfera, per ogni punto della superficie.

Ciò vuol dire che, dato un punto di riferimento da cui la sorgente luminosa è osservata, la superficie luminosa sarà visibile solo da un cono di direzioni. Una direzione è campionata con distribuzione uniforme da tale cono, e la radianza emessa è stimata come essere pari alla radianza definita per il punto di intersezione tra direzione campionata e superficie, lungo la direzione campionata.

La PDF per ogni data direzione in cui la luce è incontrata è pari, supponendo sorgente luminosa circolare, all'inverso dell'area del disco

Supponendo radianza costante in ogni punto della superficie per ogni direzione dell'emisfera, la potenza emessa dalla sorgente luminosa può essere computata come

$$\Phi_e = \pi A L_e \quad (6.102)$$

### Infinite Area Light

Una volta che un raggio fugge dalla scena e "raggiunge l' $\infty$ ", bisogna contare i contributi di tutte le sorgenti luminose all'infinito, che contribuiscono a definire il colore di background. Dunque ciò che si modella è una area light sferica (o cubica) il cui raggio tende all' $\infty$ , la quale emette radianza all'interno della sfera da tutte le direzioni, secondo una densità di distribuzione che può essere uniforme o variabile, nell'ultimo caso guidata da un'immagine fornita in input, la quale è poi proiettata sulla sfera o cubo (esempi: Equi-Rectangular Map, Cube Map).

Supponendo invece radianza emessa costante da tutte le direzioni  $L_e(\vec{p}, \hat{\omega}) = L_e$ , ciò significa che il Direct Lighting Integral<sup>28</sup>

$$\begin{aligned} L_o(\vec{p}, \hat{\omega}_o) &= \int_{S^2} f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) L_i(\vec{p}, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \\ &= L_e \int_{S^2} f_r(\vec{p}, \hat{\omega}_o, \hat{\omega}_i) |\langle \hat{n}, \hat{\omega}_i \rangle| d\hat{\omega}_i \end{aligned}$$

Dunque il campionamento di tale sorgente luminosa non è efficace per Multiple Importance Sampling, in quanto in questo caso la cosine-weighted BSDF è il termine che determina se il contributo della funzione integranda in un dato punto del dominio è significativo o meno.

Se risulta necessario campionare in ogni caso tale sorgente, ciò può essere compiuto campionando uniformemente una direzione dalla sfera unitaria di direzioni (vedi Figura 6.13): Sia  $A = r^2(\varphi_h - \varphi_l)(\cos \theta_l - \cos \theta_h)$  area della porzione di sfera definita da  $[\theta_l, \theta_h] \times [\varphi_l, \varphi_h]$ , in cui ciascun punto ha PDF  $p(\theta, \varphi) = 1/A$ . Inverse Transform Sampling può essere compiuto

---

<sup>28</sup>Reflectance Equation della superficie per la quale si incontra, al campionamento di un nuovo raggio, una sorgente luminosa

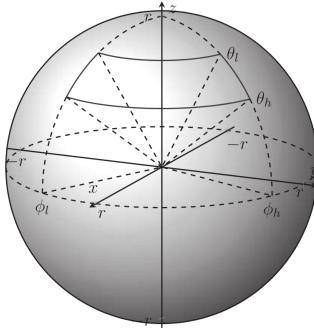


Figure 6.13: Illustrazione dei termini geometrici per parametrizzare la sfera. Immagine da [Peg16]

considerando che la CDF così ottenuta è separabile

$$\begin{aligned}
 P(\theta_s, \varphi_s) &= \int_{\varphi_l}^{\varphi_s} \int_{\theta_l}^{\theta_s} \frac{1}{r^2(\varphi_h - \varphi_l)(\cos \theta_l - \cos \theta_h)} r^2 \sin \theta d\theta d\varphi \\
 &= \int_{\varphi_l}^{\varphi_s} \frac{1}{\varphi_h - \varphi_l} d\varphi \int_{\theta_l}^{\theta_s} \frac{\sin \theta}{\cos \theta_l - \cos \theta_h} d\theta \\
 &= \left[ \frac{\varphi}{\varphi_h - \varphi_l} \right]_{\varphi=\varphi_l}^{\varphi_s} \left[ \frac{-\cos \theta}{\cos \theta_l - \cos \theta_h} \right]_{\theta=\theta_l}^{\theta_s} \\
 &= \frac{\varphi_s - \varphi_l}{\varphi_h - \varphi_l} \frac{\cos \theta_l - \cos \theta_s}{\cos \theta_l - \cos \theta_h}
 \end{aligned} \tag{6.103}$$

Da cui

$$\xi_\varphi = \frac{\varphi_s - \varphi_l}{\varphi_h - \varphi_l} \rightarrow \varphi_s = \text{lerp}(\xi_\varphi, \varphi_l, \varphi_h) \tag{6.104}$$

$$\xi_\theta = \frac{\cos \theta_l - \cos \theta_s}{\cos \theta_l - \cos \theta_h} \rightarrow \theta_s = \arccos(\text{lerp}(\xi_\theta, \cos \theta_l, \cos \theta_h)) \tag{6.105}$$

con  $\text{lerp}(x, a, b) = (1 - x)a + xb$ . Ponendo  $\varphi_l = 0, \varphi_h = 2\pi, \theta_l = 0, \theta_h = \pi$ , la direzione è campionata nell'intera sfera unitaria.

La potenza totale da una Infinite Light può essere trovata integrando la radianza emessa in tutte le direzioni e nell'area della sfera che racchiude la scena.<sup>29</sup>

$$\Phi_e = L_e \int_{S^2} d\hat{\omega} \int_A dA = 4\pi^2 R^2 L_e \tag{6.106}$$

### 6.2.2 Light Sampling

Per via dell'assunzione di linearità della Radiometria, l'illuminazione in un punto in una scena con multiple sorgenti luminose può essere calcolata sommando contributi indipendenti di ogni

<sup>29</sup>Nota come nella Equazione 6.106 non è presente il termine coseno per proiettare la radianza sull'area effettivamente irradiata. Questo perché in quanto la sfera è all' $\infty$ , si approssima il suo centro al punto  $\vec{p}$  della superficie

sorgente luminosa. In scene con migliaia di sorgenti luminose, considerarle tutte comporta un costo eccessivo. Dunque possiamo applicare un metodo stocastico corrispondente all'analogo dell'integrazione di Monte Carlo nel discreto [PJH23]

$$\sum_i^n f_i \approx \frac{f_j}{p(j)} \quad (6.107)$$

dove stiamo rimpiazzando la somma dei contributi di tutte le sorgenti luminose con una o un sottoinsieme limitato di esse, con  $p(j)$  PMF di una distribuzione  $\mathcal{P}$ , e  $j \sim \mathcal{P}$ .

Ciascuna contributo di ogni sorgente luminosa scelta è pesato nel MIS con la probabilità di essere selezionata.

Il campionamento di sorgenti luminose ideale è quello che tiene conto della potenza emessa dalle sorgenti luminose, la loro distanza dal punto di osservazione, se sono occluse o meno, la normale alla superficie<sup>30</sup>, la BSDF, tutti i fattori che contribuiscono a sottrarre importanza ad una sorgente.

Le due strategie qui analizzate sono *Campionamento Uniforme* e *Campionamento proporzionale alla potenza emessa*

### Campionamento Uniforme di Sorgenti Luminose

Strategia più semplice: campiona ogni sorgente luminosa con equa probabilità, dunque sia  $n_l$  numero di luci presenti nella scena

$$p(j) = \frac{1}{n_l} \quad (6.108)$$

### Campionamento proporzionale alla Potenza di Sorgenti Luminose

La probabilità di campionare una particolare sorgente luminosa è proporzionale alla potenza emessa. Tale approccio risulta essere più efficace del campionamento uniforme, ma non considera la posizione dello spazio della luce rispetto al punto in cui si stima la radianza uscente, limitandone l'efficacia. La potenza emessa considerata è la media nello spettro delle lunghezze d'onda visibili del flusso radiante spettrale, campionato in un prefissate lunghezza d'onda

$$p(j) = \frac{\bar{\Phi}_e^j}{\sum_{i=1}^{n_l} \bar{\Phi}_e^i} \quad (6.109)$$

---

<sup>30</sup>Si veda la legge del coseno di Lambert

# **Part III**

# **Appendici**

## Appendix A

# Numeri Pseudocasuali e Algoritmi Vari

### A.1 Generazione di numeri Pseudorandom

Nell'intero testo più volte si è fatta allusione a variabili aleatorie uniformi standard, senza effettivamente poi specificare come nella pratica esse vengono implementate. Generare numeri realmente casuali è ovviamente impossibile, dunque lo scopo di questa categoria di algoritmi è generare una sequenza parametrizzata da un seed, tipicamente settato ad un valore univoco (come un valore di hash, o una data), la quale, nonostante sia perfettamente deterministica, appaia casuale. Tale sequenza è purtroppo periodica, dunque bisogna cercare di parametrizzare e tale RNG in modo tale da rendere il periodo sufficientemente grande da non renderlo evidente, e ovviamente, tale generatore deve rispettare il prerequisito di dover approssimare il più accuratamente possibile la densità di distribuzione  $\mathcal{U}(0, 1)$ .

Il metodo più semplice per generare una tale sequenza è il *Generatore Lineare Congruenziale* (LCG): Sia  $x_t \in \{0, \dots, m - 1\}$  stato del generatore e  $m > 0$  modulo,  $a, c$  parametri del generatore, allora

$$\begin{aligned} x_t &= (ax_{t-1} + c) \bmod m, \quad t = 1, 2, \dots \\ \xi_t &= \frac{x_t}{m} \end{aligned} \tag{A.1}$$

Una scelta standard per questi parametri [KTB11] è  $a = 7^5$ ,  $c = 0$ ,  $m = 2^{31} - 1$

Un'altra categoria di RNG è il *Generatore Ricorsivo-Multiplo* di ordine<sup>1</sup>  $k$ , con stato  $\vec{x}_t = [x_{t-k+1}, \dots, x_t]^T \in \{0, \dots, m - 1\}^k$ , le cui transizioni di stato sono definite dalla generazione di un nuovo componente. Lo spazio output dell'algoritmo è un vettore distribuito con PDF  $\mathcal{U}(0, 1)^k$ .

$$x_t = (a_1 x_{t-1} + \dots + a_k x_{t-k}) \bmod m, \quad t = k, k + 1, \dots \tag{A.2}$$

Di solito, [KTB11],  $m$  è scelto come numero primo maggiore di  $2^{18} - 2$ . L'operazione sopracitata può essere interpretata come moltiplicazione matrice vettore, giungendo ad una forma detta *Generatore Matriciale Congruenziale* (MRG)

$$\vec{x}_t = (A \vec{x}_{t-1}) \bmod m, \quad t = 1, 2, \dots \tag{A.3}$$

---

<sup>1</sup>mantiene in "memoria" gli ultimi  $k$  stati, da cui il nome

dove

$$A = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & \cdots & a_1 \end{bmatrix} \text{ e } \vec{x}_t = \begin{bmatrix} x_t \\ x_{t+1} \\ \vdots \\ x_{t+k-1} \end{bmatrix} \quad (\text{A.4})$$

# Appendix B

## Codice

### B.1 Owen Scrambling di punti della Sequenza di Halton

```
// baseIndex <- base dell'Inverso Radicale
// a          <- cifre dell'inverso radicale d1...dm
// hash       <- valore casuale
Float OwenScrambledRadicalInverse(int baseIndex, uint64_t a,
                                    uint32_t hash)
{
    int base = Primes[baseIndex];
    Float invBase = 1.f / base, invBaseM = 1;

    // bits Scrambled finora
    uint64_t reversedDigits = 0;
    int digitIndex = 0;

    // finche non si esauriscono le cifre di a da
    // mettere nel floating point
    while (1 - invBaseM < 1)
    {
        // estrai il valore della <digitIndex> cifra.
        // Equivalente a un right shift del numero
        // in base <base>
        uint64_t next = a / base;
        int digitValue = a - next * base;

        // calcola un seed unico per permutare casualmente il valore
        // estratto nota che tale seed dipende da tutti i valori
        // precedentemente permutati
        uint32_t digitHash = MixBits(hash ^ reversedDigits);

        // seleziona la <digitValue>-esima permutazione
        // pseudocasuale basata sul seed <digitHash>,
        // di <base> numeri casuali
```

```
digitValue = PermutationElement(digitValue, base, digitHash);

// aggiungi la cifra a quelle calcolate finora
reversedDigits = reversedDigits * base + digitValue;

invBaseM *= invBase;
++digitIndex;
a = next;
}
return std::min(invBaseM * reversedDigits, OneMinusEpsilon);
}
```

## B.2 Acceptance-Rejection Sampling di un disco unitario

```
import numpy as np

def target_pdf(x, y):
    return (4 / np.pi) if (x**2 + y**2 <= 1 and x >= 0 and y >= 0)
    else 0

def instrumental_pdf(x, y):
    return 1 if (0 <= x <= 1 and 0 <= y <= 1) else 0

def acceptance_probability(x, y, b):
    return

def rejection_sampling(num_samples):
    samples = []
    accepted_samples = 0
    b = 4 / np.pi

    while accepted_samples < num_samples:
        x = np.random.uniform(0, 1)
        y = np.random.uniform(0, 1)
        if np.random.uniform(0, 1)*b*target_pdf(x, y)
        <= instrumental_pdf(x, y):
            samples.append((x, y))
            accepted_samples += 1

    return samples

def alt_rejection_sampling(num_samples):
    samples = []
    accepted_samples = 0
    attempts = 0
    while accepted_samples < num_samples:
        attempts++
        x = np.random.uniform(0, 1)
```

```

y = np.random.uniform(0, 1)
if x**2+y**2 <= 1:
    samples.append((x, y))
    accepted_samples += 1

pi_estimate = accepted_samples / attempts
return (samples, pi_estimate)

```

### B.3 Metropolis-Hastings Sampling per campionare $\mathcal{N}(0, 1)$

```

import math
import numpy as np
from numpy.random import default_rng

_rng : Generator = default_rng()

def normPdf(value, mean = 0.0, stddev = 1.0):
    rSqrt2Pi = 0.39894228
    rStddev = 1 / stddev
    return rSqrt2Pi * rStddev * math.exp(-0.5*((value - mean)* rStddev)**2)

def metropolisHastings(samples, N):
    for i in np.arange(N-1):
        u = _rng.uniform()
        proposedSample = _rng.normal(samples[i], 0.05)

        acceptance =
            np.clip(min(1, normPdf(proposedSample) / normPdf(samples[i])), 0., 1.)
        if u < acceptance:
            samples[i+1] = proposedSample
        else:
            samples[i+1] = samples[i]

    return

```

### B.4 Monte Carlo Integration per un integrale semplice

Ripreso dal codice della repository del progetto:

```

import math
import numpy as np
import numpy.typing as npt
import sys
from numpy.random import default_rng
from typing import Generator, Annotated, Callable
from annotated_types import Gt

_rng : Generator = default_rng()

```

```
def integrand(x : float):
    arg : float = math.pi*math.sqrt(x*0.1)
    return (math.sin(arg)/arg)*math.e**arg;

# suppone il calcolo dell'integrale definito e' nell'intervallo [0,10]
def trueIntegralValue():
    return (10*math.e**math.pi+10)/(math.pi**2)

def monteCarloIntegration(integrandFunction : Callable[[float], float],
                           domainSize : float, nSamples : Annotated[int, Gt(0)]) :
    sum : float = 0.;
    for i in np.arange(0, nSamples):
        u : float = _rng.uniform()
        x_i = 10. * u + sys.float_info.epsilon * (u == 0)
        sum += integrandFunction(x_i)
    return domainSize * sum / nSamples
```

## B.5 Semplice Path Tracer

Segue una semplice implementazione dell'Algoritmo del Path Tracing, riferito a <http://www.kevinbeason.com/smallpt/>

```
#include <math.h>
#include <stdlib.h>
#include <stdio.h>

struct Vec {
    double x, y, z;
    Vec(double x_=0, double y_=0, double z_=0){ x=x_; y=y_; z=z_; }
    Vec operator+(const Vec &b) const { return Vec(x+b.x,y+b.y,z+b.z); }
    Vec operator-(const Vec &b) const { return Vec(x-b.x,y-b.y,z-b.z); }
    Vec operator*(double b) const { return Vec(x*b,y*b,z*b); }
    Vec mult(const Vec &b) const { return Vec(x*b.x,y*b.y,z*b.z); }
    Vec& norm(){ return *this = *this * (1/sqrt(x*x+y*y+z*z)); }
    double dot(const Vec &b) const { return x*b.x+y*b.y+z*b.z; }
    Vec operator%(Vec&b){return Vec(y*b.z-z*b.y,z*b.x-x*b.z,x*b.y-y*b.x);}
};

struct Ray { Vec o, d; Ray(Vec o_, Vec d_) : o(o_), d(d_) {} };
enum Refl_t { DIFF, SPEC, REFR };
struct Sphere {
    double rad;
    Vec p, e, c;
    Refl_t refl;
    Sphere(double rad_, Vec p_, Vec e_, Vec c_, Refl_t refl_):
        rad(rad_), p(p_), e(e_), c(c_), refl(refl_) {}
    double intersect(const Ray &r) const {
        Vec op = p-r.o;
```

```

        double t, eps=1e-4, b=op.dot(r.d), det=b*b-op.dot(op)+rad*rad;
        if (det<0) return 0; else det=sqrt(det);
        return (t=b-det)>eps ? t : ((t=b+det)>eps ? t : 0);
    }
};

Sphere spheres[] = {
    Sphere(1e5, Vec( 1e5+1,40.8,81.6), Vec(),Vec(.75,.25,.25),DIFF),
    Sphere(1e5, Vec(-1e5+99,40.8,81.6),Vec(),Vec(.25,.25,.75),DIFF),
    Sphere(1e5, Vec(50,40.8, 1e5), Vec(),Vec(.75,.75,.75),DIFF),
    Sphere(1e5, Vec(50,40.8,-1e5+170), Vec(),Vec(), DIFF),
    Sphere(1e5, Vec(50, 1e5, 81.6), Vec(),Vec(.75,.75,.75),DIFF),
    Sphere(1e5, Vec(50,-1e5+81.6,81.6),Vec(),Vec(.75,.75,.75),DIFF),
    Sphere(16.5,Vec(27,16.5,47), Vec(),Vec(1,1,1)*.999, SPEC),
    Sphere(16.5,Vec(73,16.5,78), Vec(),Vec(1,1,1)*.999, REFR),
    Sphere(600, Vec(50,681.6-.27,81.6),Vec(12,12,12), Vec(), DIFF)
};

inline double clamp(double x){ return x<0 ? 0 : x>1 ? 1 : x; }
inline int toInt(double x){ return int(pow(clamp(x),1/2.2)*255+.5); }
inline bool intersect(const Ray &r, double &t, int &id){
    double n=sizeof(spheres)/sizeof(Sphere), d, inf=t=1e20;
    for(int i=int(n);i--;) if((d=spheres[i].intersect(r))&&d<t){t=d;id=i;}
    return t<inf;
}
Vec radiance(const Ray &r, int depth, unsigned short *Xi){
    double t;
    int id=0;
    if (!intersect(r, t, id)) return Vec();
    const Sphere &obj = spheres[id];
    Vec x=r.o+r.d*t, n=(x-obj.p).norm(), nl=n.dot(r.d)<0?n:n*-1, f=obj.c;
    double p = f.x>f.y && f.x>f.z ? f.x : f.y>f.z ? f.y : f.z;
    if (++depth>5) if (erand48(Xi)<p) f=f*(1/p); else return obj.e;
    if (obj.refl == DIFF){
        double r1=2*M_PI*erand48(Xi), r2=erand48(Xi), r2s=sqrt(r2);
        Vec w=nl, u=((fabs(w.x)>.1?Vec(0,1):Vec(1)%w).norm(), v=w*u;
        Vec d = (u*cos(r1)*r2s + v*sin(r1)*r2s + w*sqrt(1-r2)).norm();
        return obj.e + f.mult(radiance(Ray(x,d),depth,Xi));
    } else if (obj.refl == SPEC)
        return obj.e + f.mult(radiance(Ray(x,r.d-n*2*n.dot(r.d)),depth,Xi));
    Ray reflRay(x, r.d-n*2*n.dot(r.d));
    bool into = n.dot(nl)>0;
    double nc=1, nt=1.5, nnt=into?nc/nt:nt/nc, ddn=r.d.dot(nl), cos2t;
    if ((cos2t=1-nnt*nnt*(1-ddn*ddn))<0)
        return obj.e + f.mult(radiance(reflRay,depth,Xi));
    Vec tdir = (r.d*nnt - n*((into?1:-1)*(ddn*nnt+sqrt(cos2t))).norm();
    double a=nt-nc, b=nt+nc, R0=a*a/(b*b), c = 1-(into?-ddn:ddn:tdir.dot(n));
    double Re=R0+(1-R0)*c*c*c*c*c, Tr=1-Re, P=.25+.5*Re, RP=Re/P, TP=Tr/(1-P);
    return obj.e + f.mult(depth>2 ? (erand48(Xi)<P ?
        radiance(reflRay,depth,Xi)*RP:radiance(Ray(x,tdir),depth,Xi)*TP) :
        radiance(reflRay,depth,Xi)*Re+radiance(Ray(x,tdir),depth,Xi)*Tr);
}

```

```

}

int main(int argc, char *argv[]){
    int w=1024, h=768, samps = argc==2 ? atoi(argv[1])/4 : 1;
    Ray cam(Vec(50,52,295.6), Vec(0,-0.042612,-1).norm());
    Vec cx=Vec(w*.5135/h), cy=(cx/cam.d).norm()*.5135, r, *c=new Vec[w*h];
#pragma omp parallel for schedule(dynamic, 1) private(r)
    for (int y=0; y<h; y++){
        fprintf(stderr, "\rRendering (%d spp) %5.2f%%", samps*4, 100.*y/(h-1));
        for (unsigned short x=0, Xi[3]={0,0,y*y*y}; x<w; x++)
            for (int sy=0, i=(h-y-1)*w+x; sy<2; sy++)
                for (int sx=0; sx<2; sx++, r=Vec()){
                    for (int s=0; s<samps; s++){
                        double r1=erand48(Xi), dx=1-sqrt(1-r1);
                        double r2=erand48(Xi), dy=1-sqrt(1-r2);
                        Vec d = cx*( ( (sx+.5 + dx)/2 + x)/w - .5) +
                                cy*( ( (sy+.5 + dy)/2 + y)/h - .5) + cam.d;
                        r = r + radiance(Ray(cam.o+d*140,d.norm()),0,Xi)*(1./samps);
                    }
                    c[i] = c[i] + Vec(clamp(r.x),clamp(r.y),clamp(r.z))*.25;
                }
        }
    FILE *f = fopen("image.ppm", "w");
    fprintf(f, "P3\n%d %d\n%d\n", w, h, 255);
    for (int i=0; i<w*h; i++)
        fprintf(f,"%d %d %d ",.toInt(c[i].x), toInt(c[i].y), toInt(c[i].z));
}

```

Comandi per la compilazione e resa (linux):

```

g++ -O3 -fopenmp smallpt.cpp -o smallpt
time ./smallpt 5000
display image.ppm

```

## B.6 Calcolo della varianza di un'immagine

```

from PIL import Image
import numpy as np

def compute_image_variances(image_path):
    # Open the image
    image = Image.open(image_path)

    # Convert the image to a numpy array
    image_array = np.array(image)

    # Convert the image to the linear RGB color space
    image_linear_rgb = image_array.astype(np.float32) / 255.0

    # Compute the variances for each color channel

```

## B.6. CALCOLO DELLA VARIANZA DI UN'IMMAGINE

---

```
channel_variances = np.var(image_linear_rgb, axis=(0, 1))

# Compute the luminance channel, using RGB->Luma from Photometric/digital ITU BT.709:
luminance = 0.2126 * image_linear_rgb[:, :, 0] + 0.7152 * image_linear_rgb[:, :, 1] + 0.0722 *

# Compute the variance of the luminance channel
luminance_variance = np.var(luminance)

return channel_variances, luminance_variance
```

# Bibliography

- [al77] Nicodemus F. E. et al. *Geometrical considerations and nomenclature for reflectance*. Tech. rep. Final Report National Bureau of Standards, Washington, DC. Inst. for Basic Standards, 1977.
- [Sch94] Christophe Schlick. “An Inexpensive BRDF Model for Physically-Based Rendering”. In: *Proceedings of Eurographics*. 1994.
- [LS05] István Lazányi and László Szirmay-Kalos. “Fresnel Term Approximation for Metals”. In: *Proceedings of WSCG*. 2005.
- [Sch07] Janos Schanda. *Colorimetry. Understanding The CIE System*. Wiley Interscience, 2007.
- [Jam09] Barbara G. Grant James M. Palmer. *The Art of Radiometry*. SPIE Press, 2009.
- [KTB11] Dirk P. Kroese, Thomas Talmre, and Zdravko I. Botev. *Handbook of Monte Carlo Methods*. 1st ed. Wiley, 2011.
- [Ike14] Katsushi Ikeuchi. *Computer Vision*. 1st ed. Springer New York, NY, 2014.
- [Peg16] Vincent Pegoraro. *Handbook of Digital Image Synthesis: Scientific Foundations of Rendering*. 1st ed. A K Peters/CRC Press, 2016.
- [AHH18] Tomas Akenine-Möller, Eric Haines, and Naty Hoffman. *Real-Time Rendering, Fourth Edition*. 4th ed. A K Peters/CRC Press, 2018.
- [Wen19] Johannes Hanika et al. Wenzel Jakob. “A Low-Dimensional Function Space for Efficient Spectral Upsampling”. In: *Eurographics Symposium on Rendering 2019*. 2019.
- [al21] Alban Fichet et al. “An OpenEXR Layout for Spectral Images”. In: *Journal of Computer Graphics Techniques* 10.3 (2021).
- [al22] Jingang Zhang et al. “A survey on computational spectral reconstruction methods from RGB to hyperspectral imaging”. In: *Science Reports* 12 (2022).
- [PJH23] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. 4th ed. Morgan Kaufmann, 2023.

# Glossary

**Absorptance** Rapporto tra flusso radiante o luminoso assorbito e flusso radiante o luminoso incidente in date condizioni (come la lunghezza del percorso nel mezzo considerato). 58

**Bidirectional Reflectance Distribution Function** Esprime la proporzione tra irradianza (spettrale) incidente nel punto  $\vec{p}$  da un angolo solido centrato in  $\hat{\omega}_i$  e radianza (spettrale) riflessa in direzione  $\hat{\omega}_o$ . Misurata in  $\text{sr}^{-1}$ . 58, 63

**Emittanza Luminosa** Flusso luminoso emesso da una superficie per unità di area, emittanza radiante spettrale pesata sulla percezione umana. Si misura in lx. 45

**Emittanza Radiante** Flusso Radiante emesso per unità di area. Misurata in  $\text{W/m}^2$ . 41

**Energia Luminosa** Energia trasportata dalla luce percepita da un osservatore umano. Misurata in  $\text{lm} \cdot \text{s}$ . 45

**Energia Radiante** Energia aggregata di tutti i fotoni in una radiazione elettromagnetica. Misurata in J. 41

**Flusso Luminoso** Misura la potenza della luce percepita da un osservatore umano. Misurata in lm. 45

**Flusso Radiante** Energia Radiante *emessa, riflessa, trasmessa, incidente*, per unità di tempo. Misurata in W. 41

**Illuminanza** Flusso luminoso incidente una superficie per unità di area, irradianza spettrale pesata sulla percezione umana. Si misura in lx. 45

**Intensità Luminosa** Misura della intensità radiante spettrale pesata sulla percezione umana. Misurata in cd. 45

**Intensità Radiante** Flusso Radiante emesso, riflesso, trasmesso, per unità di angolo solido. Misurata in  $\text{W/sr}$ . 41

**Irradianza** Flusso Radiante incidente per unità di area. Misurata in  $\text{W/m}^2$ . 41

**Luminanza** misura dell'intensità luminosa per unità di area di luce che è emessa, incidente, trasmessa, riflessa, passa attraverso una particolare direzione. Misurata in  $\text{cd/m}^2$ . 45

**Radianza** Flusso Radiante emesso, riflesso, trasmesso, incidente da/su una superficie, per unità di angolo solido per unità di *area proiettata*. Misurata in  $\text{W}/(\text{m}^2 \cdot \text{sr})$ . 41

## Glossary

---

**Radiosità** Flusso Radiante *uscente* (emesso, riflesso e trasmesso) per unità di area. Misurata in  $\text{W/m}^2$ . 41

**Reflectance** Per una data radiazione con data SPD, polarizzazione, geometria: Rapporto tra flusso radiante o luminoso riflesso e flusso radiante o riflesso incidente. Nel primo caso, le grandezze possono essere spettrali o cumulative. Misura adimensionale. 58

**Trasmittance** Per una data radiazione con data SPD, polarizzazione, geometria: Rapporto tra flusso radiante o luminoso trasmesso e flusso radiante o riflesso incidente, nelle date condizioni (come la lunghezza del percorso della radiazione nell mezzo che si considera). Nel primo caso, le grandezze possono essere spettrali o cumulative. Misura adimensionale. 58