

ALMA MATER STUDIORUM · UNIVERSITÀ DI  
BOLOGNA

---

**Corso di Laurea Magistrale in  
Ingegneria Informatica**

# **Ray Tracing**

## **LAB 4**

Fondamenti di Computer Graphics M

**Cristian Davide Conte 0001034932**

**Anno Accademico: 2022-2023**

# Contents

<b>1</b>	<b>Gestione shadow rays per la generazione degli hard shadows</b>	<b>2</b>
<b>2</b>	<b>Gestione ricorsiva dei reflection rays</b>	<b>4</b>
<b>3</b>	<b>Gestione soft shadows mediante risorse luminose ad area</b>	<b>5</b>

# Chapter 1

## Gestione shadow rays per la generazione degli hard shadows

La tecnica del raytracing consiste nel tracciare dei raggi (detti "primari") attraverso i pixel della scena (uno per pixel nel raytracing classico, molteplici con la tecnica del path-tracing) osservando l'intersezione con uno o più oggetti. In caso di mancata intersezione il colore del pixel corrispondente può direttamente essere settato al colore dello sfondo. Nel caso in cui invece si colpisce un oggetto, altri raggi (detti "secondari") potranno essere generati (ripetendo il procedimento un numero di volte pari al massimo numero di rimbalzi consentito). Al fine di creare le ombre tramite l'algoritmo del raytracing, non appena un oggetto viene colpito da un raggio (primario o secondario), è possibile castare degli Shadow Rays verso le sorgenti luminose: se questi intersecano altri oggetti in scena, allora l'oggetto dal quale sono partiti è in ombra. L'implementazione dell'algoritmo per la generazione delle hard shadows tramite shadow rays è all'interno del metodo "TraceRay" del file "raytracer\_student.cpp".

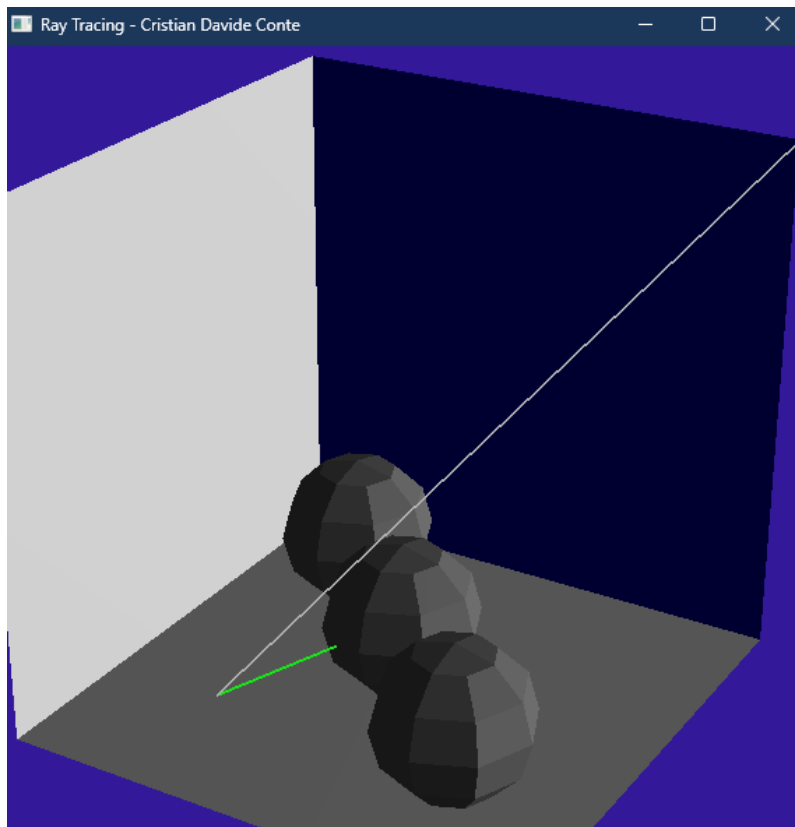


Figure 1.1: Hard Shadows - Shadow Ray (in verde)

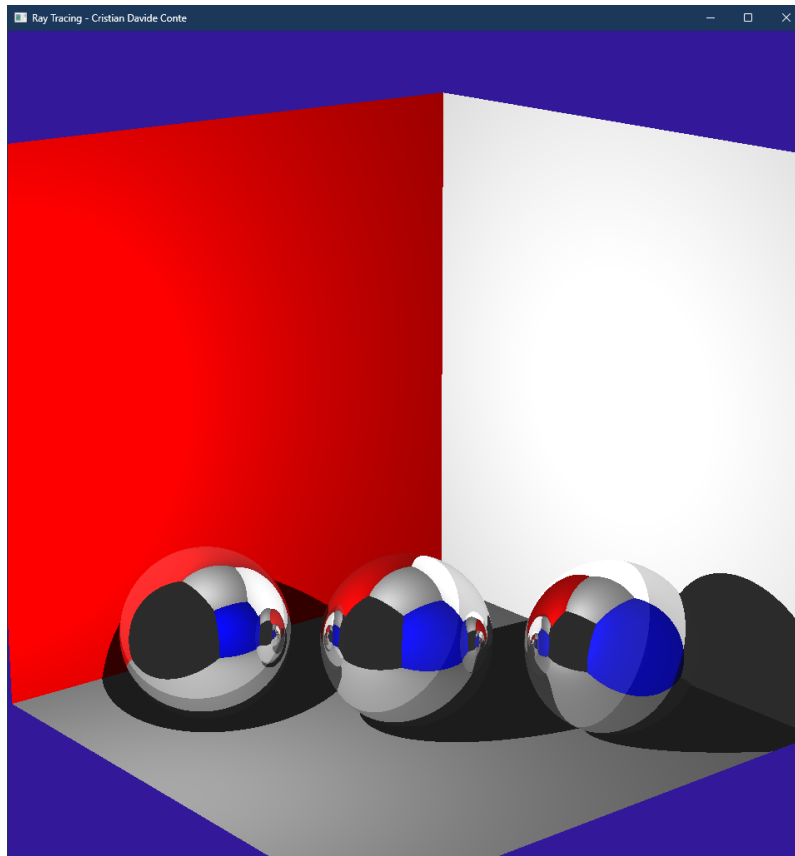


Figure 1.2: Hard Shadows

## Chapter 2

### Gestione ricorsiva dei reflection rays

Al fine di implementare la gestione dei reflection rays, si va a controllare che le superfici colpite dai raggi siano riflesse (e che il numero di rimbalzi rimanenti sia  $> 0$ ) generando così dei Reflection Rays che hanno una direzione data dall'angolo (coseno) d'incidenza relativo alla superficie (calcolato tramite il dot-product tra la normale alla superficie e il raggio in ingresso). Anche questo algoritmo è implementato tramite la funzione "TraceRay" che, richiamando se stessa e decrementando "bounce\_count", realizza la parte ricorsiva dell'algoritmo e permette la propagazione dei Reflection Rays sulle altre superfici.

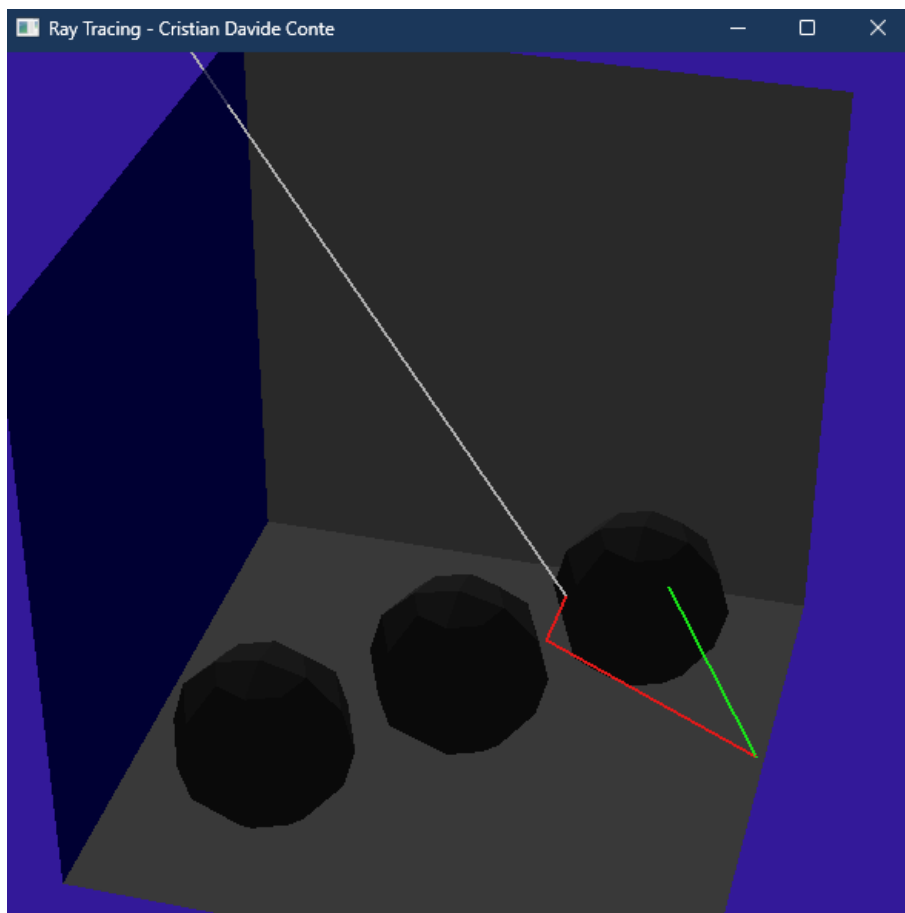


Figure 2.1: Reflection Rays (in rosso)

## Chapter 3

# Gestione soft shadows mediante risorse luminose ad area

La generazione delle hard shadows del capitolo 1 avveniva perchè, a seguito di una collisione raggio-oggetto, un singolo Shadow Ray veniva castato verso le sorgenti di luce (uno per sorgente luminosa). Al fine di generare le zone di penombra e di conseguenza le soft-shadows, per ogni luce andiamo a castare molteplici Shadow Ray (in pratica stiamo considerando la luce come "ad area" invece che "puntiforme"). La strategia più semplice d'implementazione consiste nel prendere punti a caso sulla superficie della luce: questo potrebbe (e spesso accade) generare alcuni artefatti in fase di rendering. Per sistemare questo problema si è deciso di dividere la superficie della luce in maniera uniforme seguendo una divisione basata sulla variabile "increment\_factor" (come se sopra vi fosse applicata una griglia di  $(1 / \text{increment\_factor})$  righe e  $(1 / \text{increment\_factor})$  colonne).

Nota: l'opzione "num\_shadow\_samples" all'interno dello script "run\_LAB\_04.bat" permettere di scegliere il numero di Shadow Ray castati (per luce).

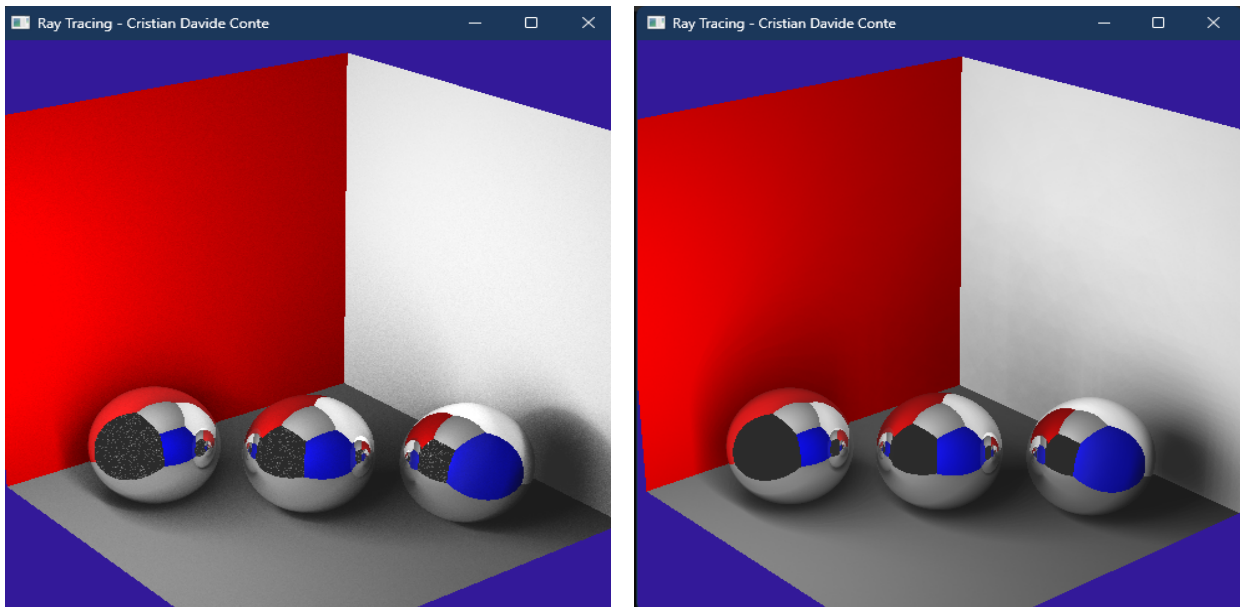


Figure 3.1: Soft Shadows - Random Sampling (sx) vs Grid Sampling (dx)