

Path Render

Jian Cao

March 1, 2021

Purpose

1 Purpose

- Ray tracer
- Monte Carlo renderer
- Sampling light
- Ref

Ray tracer

- Design a path render that can do recursive Ray Tracing by using CPU and implicit Monte Carlo.
 - Rect, ball and triangle.
 - material: Lambertian, Metal, Diffuse and Dielectric
 - Modeling Light Scatter, Reflectance and Mirrored Light Reflection.
 - Mirrored Light Reflection
 - A Scene with Metal Spheres
 - Fuzzy Reflection
 - Ray-Triangle Intersection
 - Image texture mapping

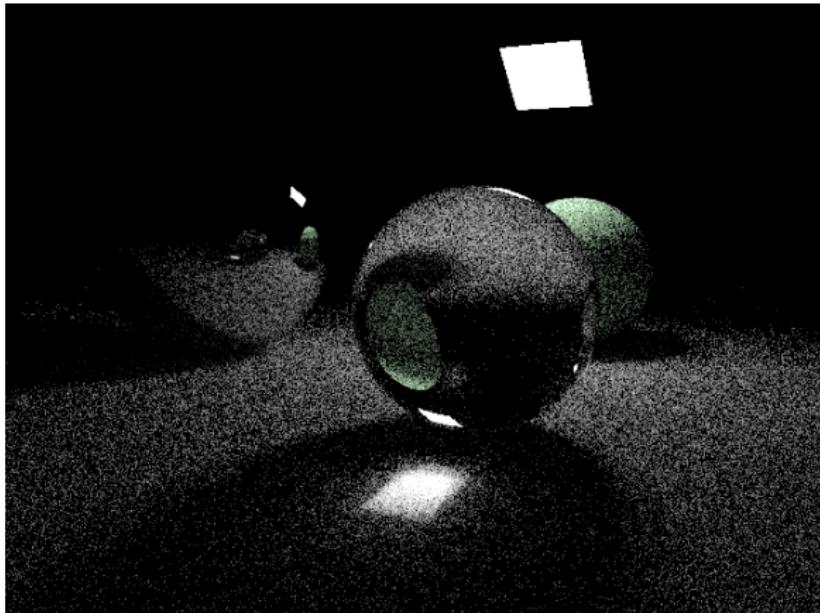


Figure: quad-light

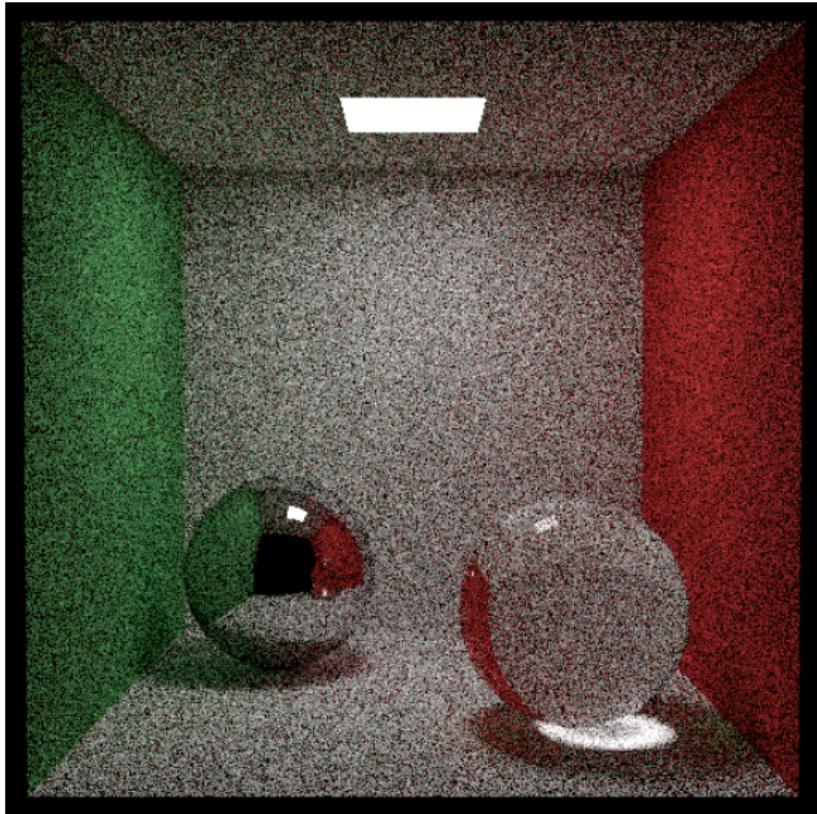


Figure: cornell-box

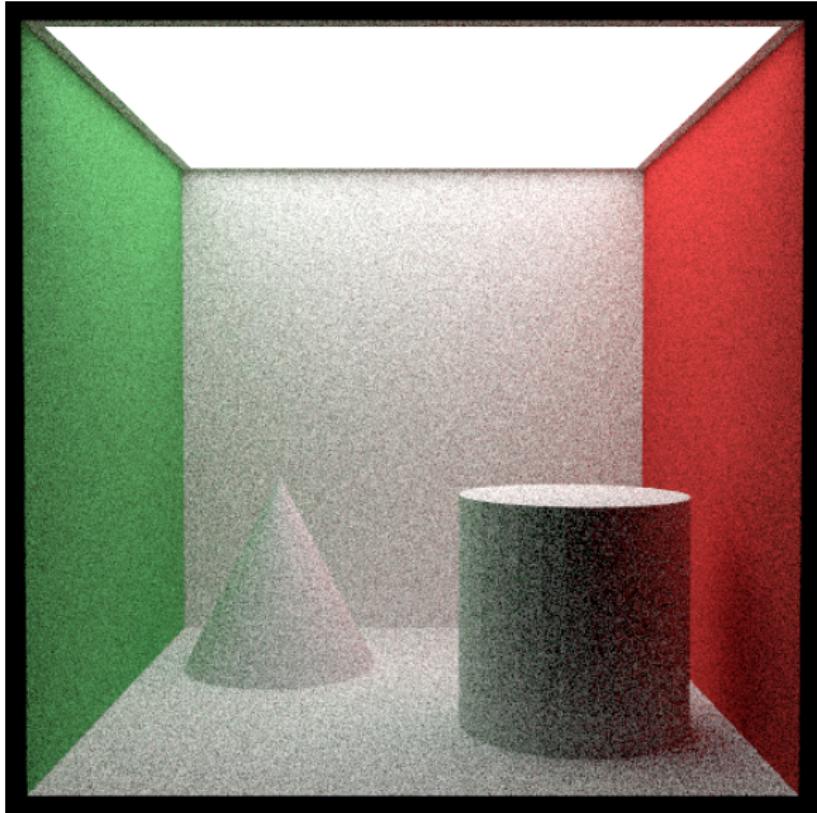


Figure: simple-geometry for Ray-Triangle Intersection

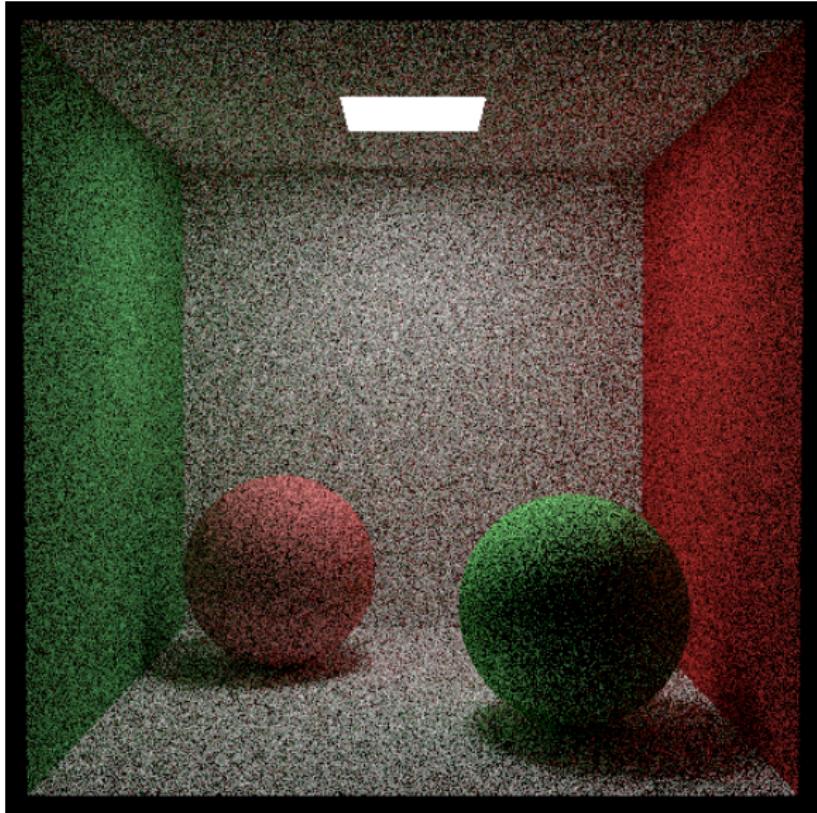


Figure: constant-cornell-box

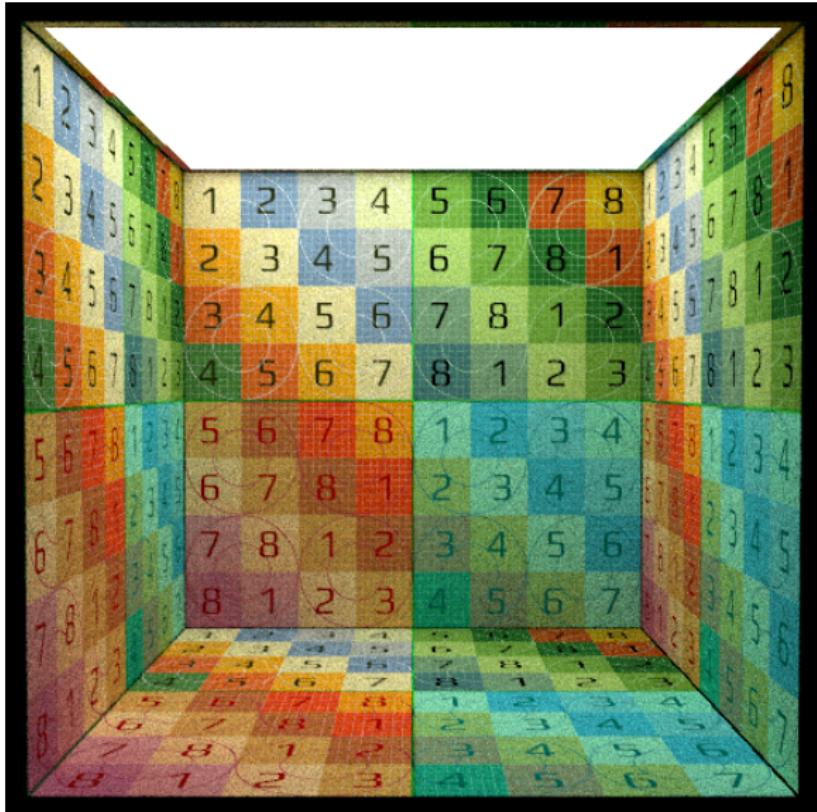


Figure: textured-box

- implement sampling routines for different distributions on the sphere.
- T
- implement Monte Carlo sampling techniques for your existing materials
- implement integrators for rendering
- implement Phong Material
- implement Normal Integrator that using shading normal as the output color.
- implement Ambient Occlusion Integrator
- implement Material Sampling Integrator

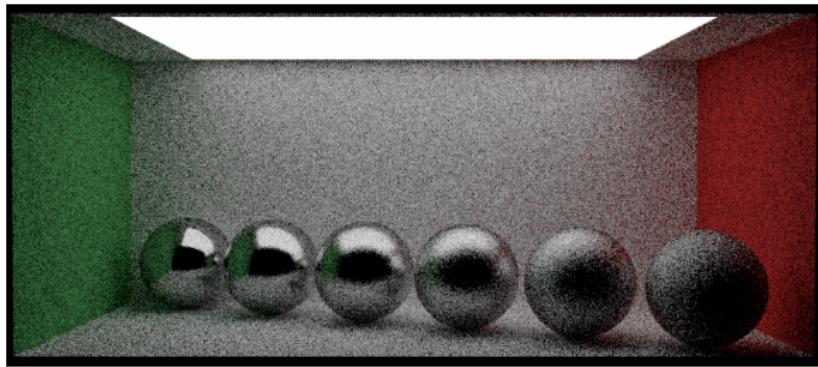


Figure: phong material



Figure: Ambient Occlusion

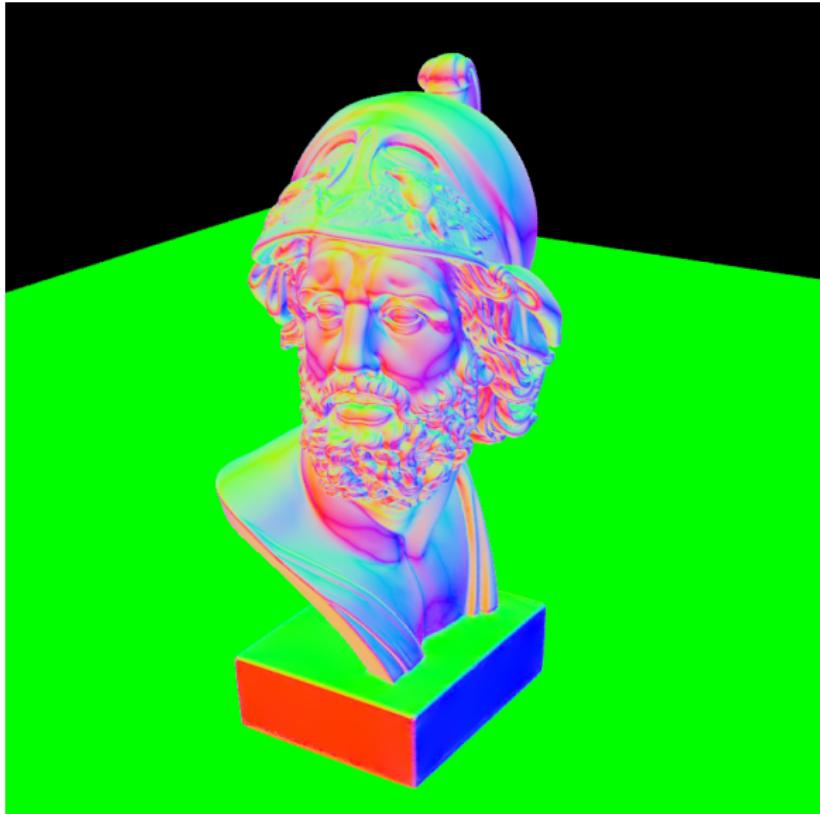


Figure: Normal Integrator

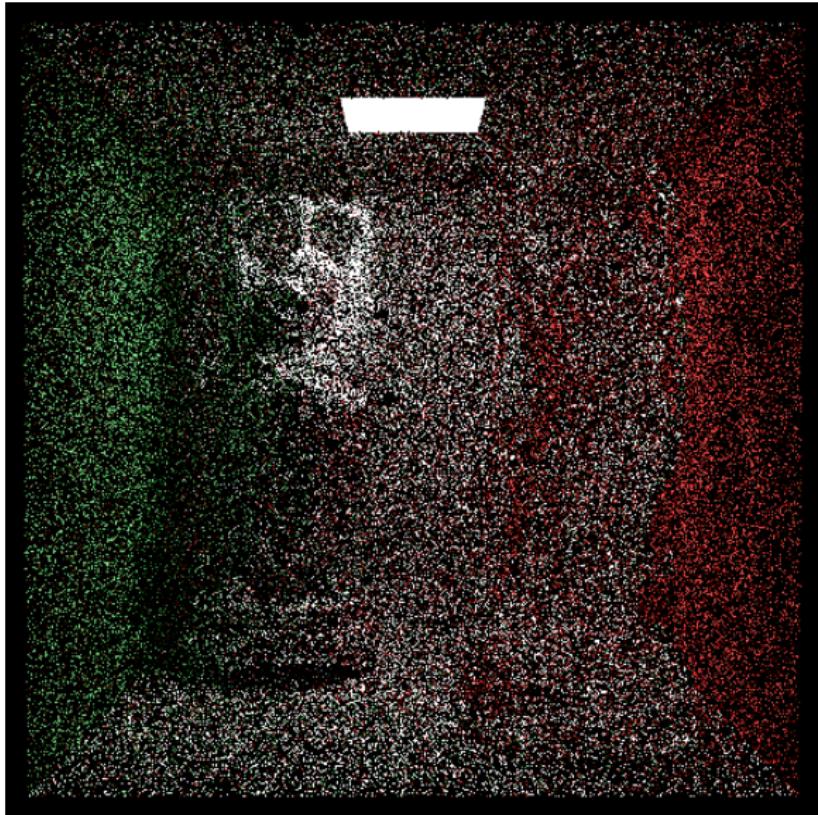


Figure: Material Sampling Integrator

Monte Carlo renderer

- implement sampling routines for different geometries like sphere, rect and triangle.
- calculate PDFs for your existing materials
- sample emissive geometry in the scene
- Direct Lighting Material Integrator
- Multiple Importance Sampling Integrator
- MIS Integrator With probability of 0.5, generate a direction by sampling the material; otherwise, generate a direction by sampling the lights.
- Full Path Tracing including direct and indirect lighting.

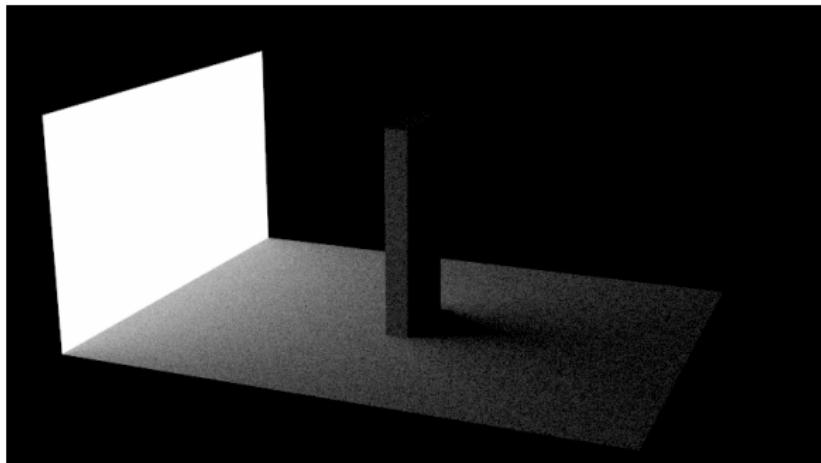


Figure: Material Sampling odyssey

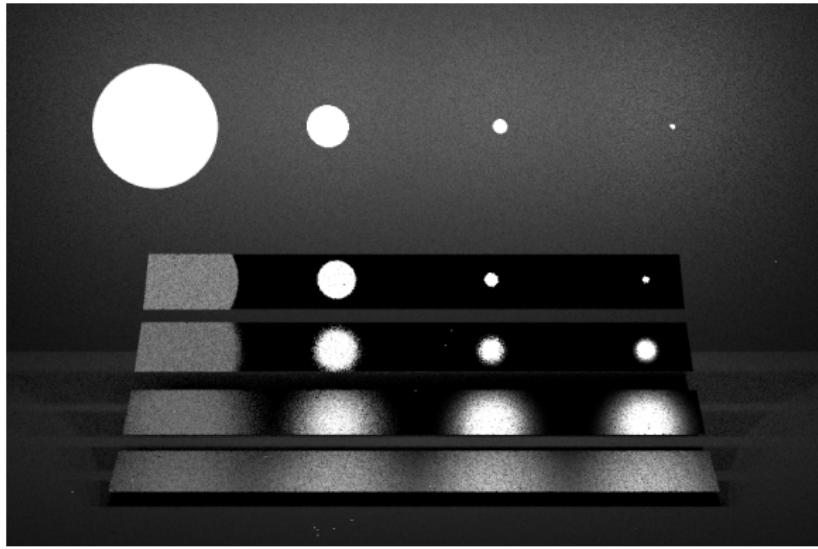


Figure: MIS Integrator

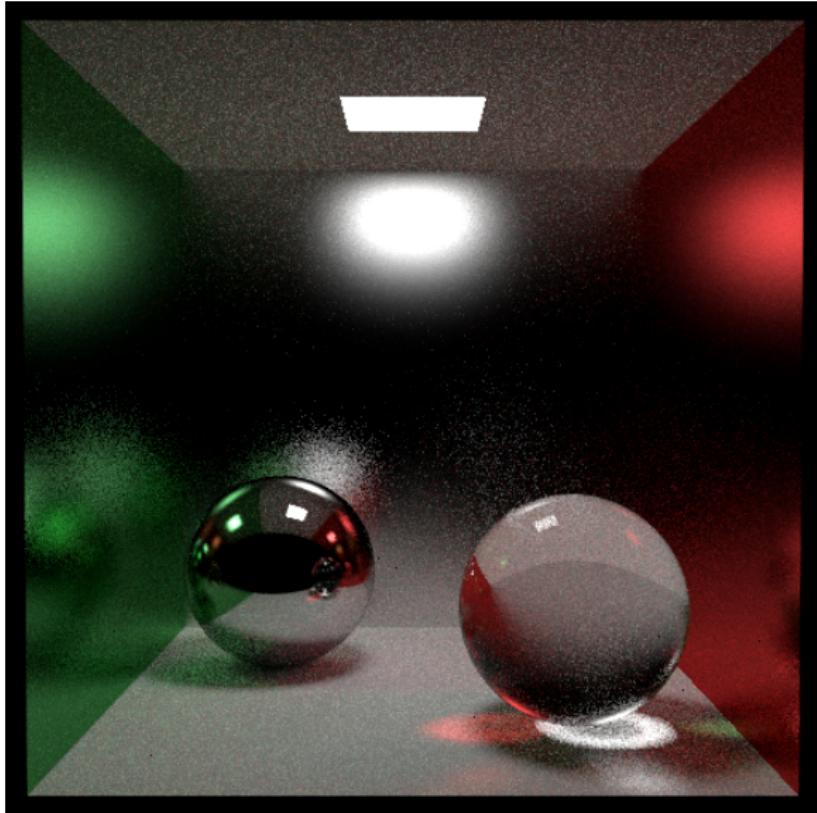


Figure: Full Path Tracing

References



Prof. Wojciech Jarosz

dartmouth Rendering Algorithms



Peter Shirley

Ray Tracing in One Weekend

<https://raytracing.github.io/books/RayTracingInOneWeekend.html>

The End