Distributed Renderer

CIS565 Final Project
by Sanchit Garg & Dome Pongmongkol

FINAL REPORT

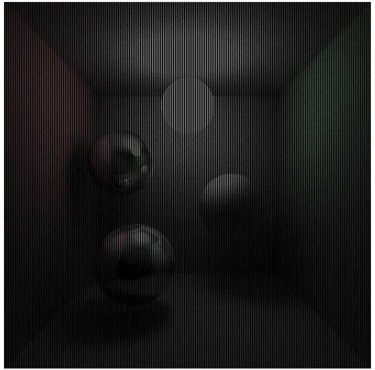
Project Status

- Networking and Render Division
  - TCP is used for sending and receiving data.
  - The scene definition is sent by the viewer to the Leader and the leader distributed is over to the other renderers.
  - When the renderers are done with n iterations, they send the image data to the leader. The leader accumulates all the data and returns it to the viewer.
- Rendering
  - We use Multiple Importance Sampling for the backend renderer.

Render Division

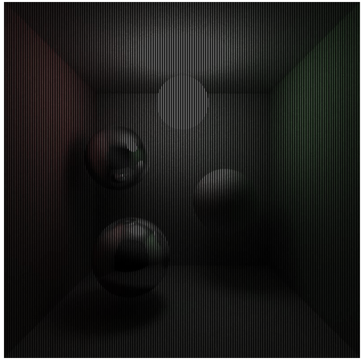
- Based on the number of rendering systems, divide the image to be rendered.
- Use render id as offset for starting the render
- Use total renders to find the pixels to be rendered.
- Example :
    - Let total renders = 4
    - Pixels rendered by Renderer id 0 : 0, 4, 8 ...
    - Pixels rendered by Renderer id 0 : 1, 5, 9 ...
    - Pixels rendered by Renderer id 0 : 2, 6, 10 ...
    - Pixels rendered by Renderer id 0 : 3, 7, 11 ...
- ***TODO*** : Accumulate the image from all the renders to get the final image.

Screen Shots (Render division)

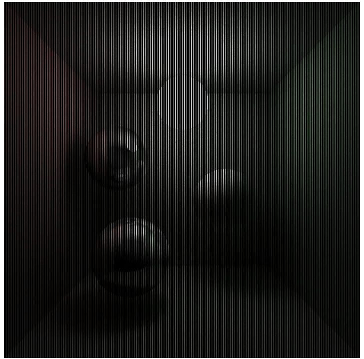


Render ID 0

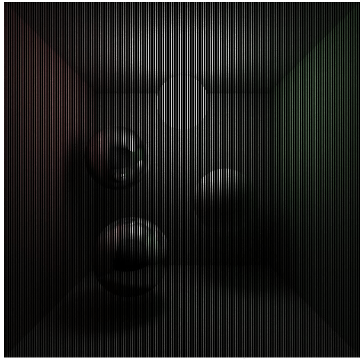
+



Render ID 1

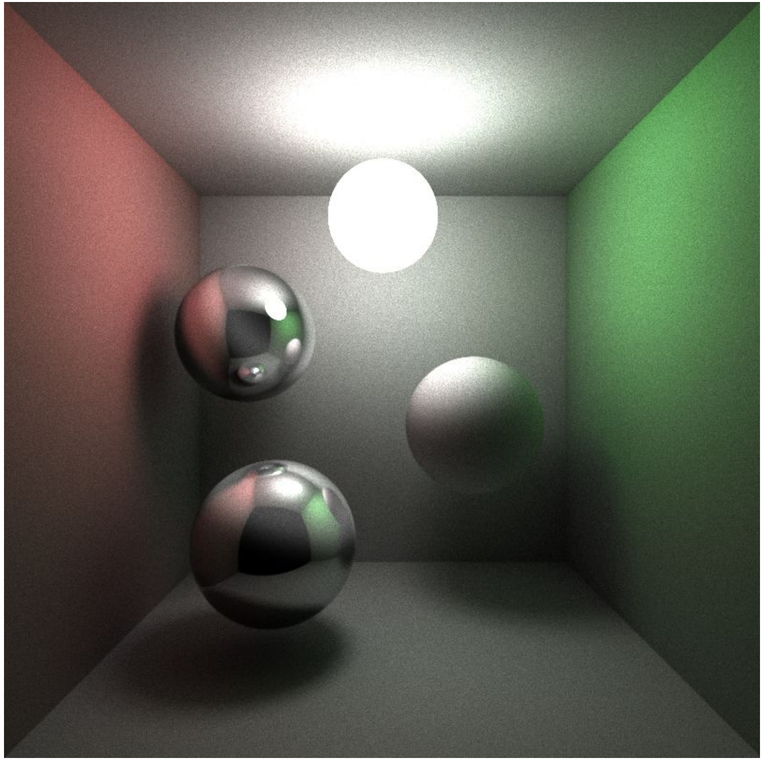


Render ID 2



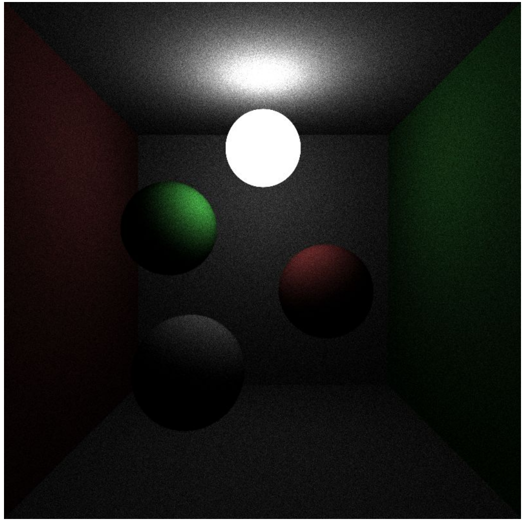
Render ID 3

=



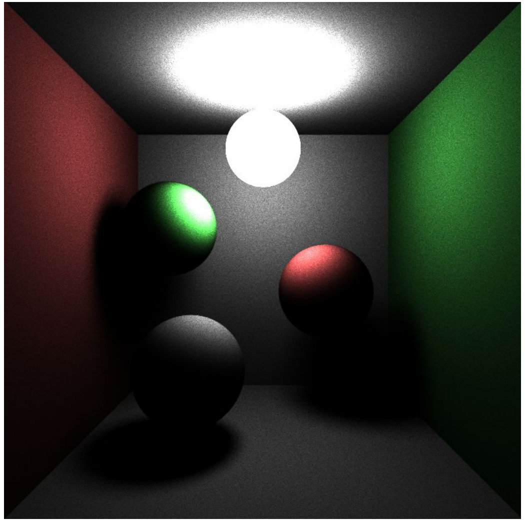
Full render

Screen Shots (BIS + LIS = MIS)



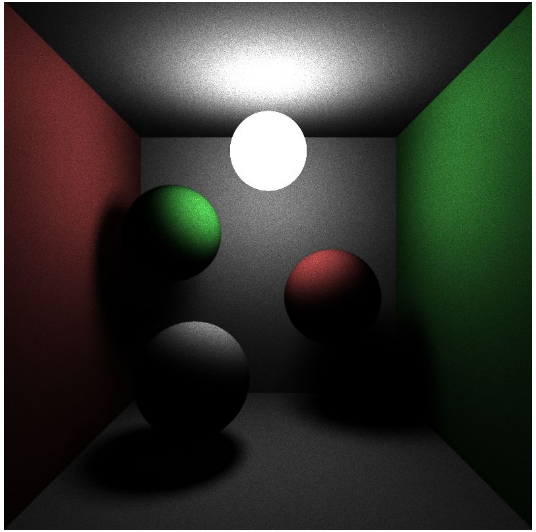
BRDF Importance sampling (FIXED)

+



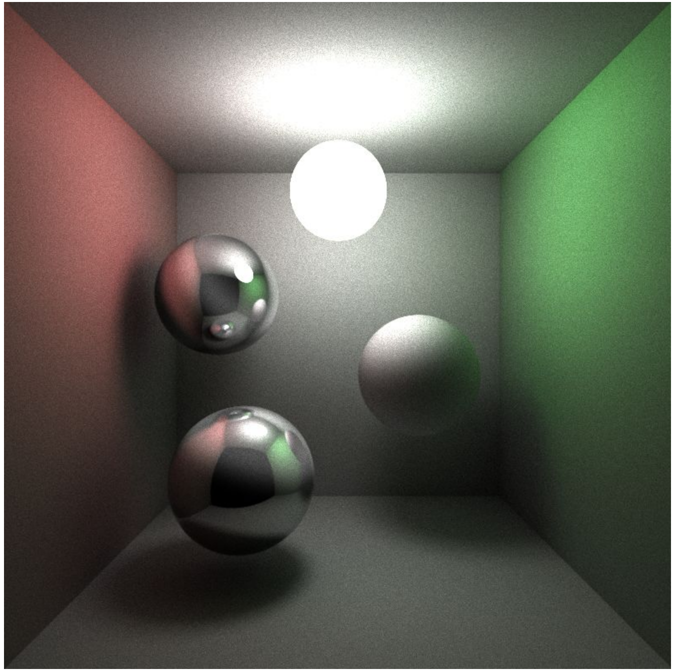
Light Importance sampling (FIXED)

=

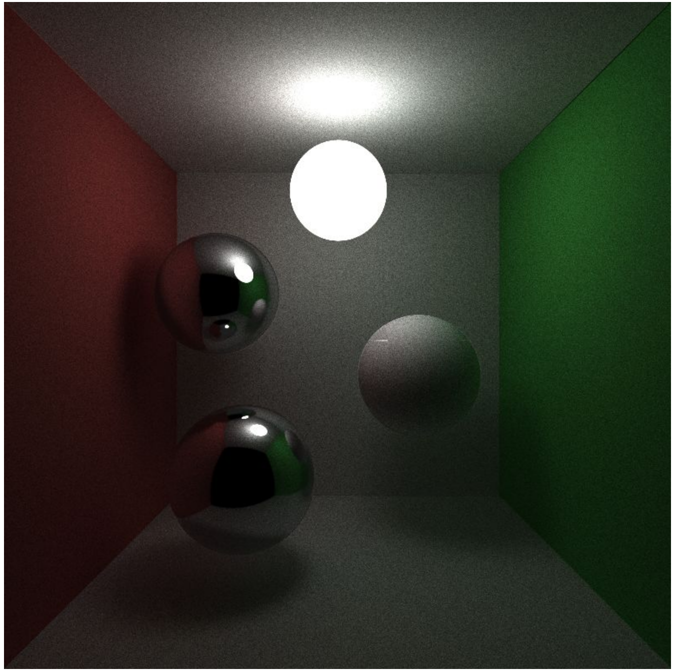


Multiple Importance sampling
[Direct Illumination]

Screen Shots (Comparison MIS vs CIS565 HW)



MIS Path Tracer
(Trace Depth 3, 1000 samples)



CIS 565 Path Tracer with Direct Illumination
(Trace Depth 3, 1000 samples)