Distributed Renderer

CIS565 Final Project
by Sanchit Garg & Dome Pongmongkol

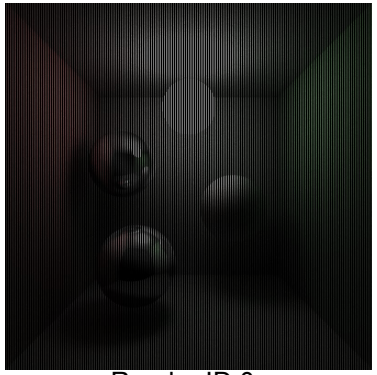
Progress

- Render Division
  - Render a part of the image based on total number of renderers.
  - Rendering every nth pixel based on number of renderers and the index of the renderer.
  - Image transfer through TCP protocol and accumulated on the viewer.
- Multiple Importance Sampling
  - Found bugs in the renderer and fixed them
  - Implemented Indirect illumination. Now can render full global illumination results.
  - Reflective materials added. Refractive material implemented but needs debugging.

Render Division

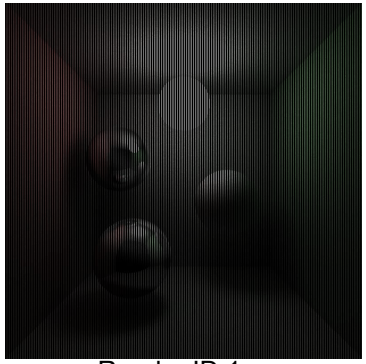
- Based on the number of rendering systems, divide the image to be rendered.
- Use render id as offset for starting the render
- Use total renders to find the pixels to be rendered.
- Example :
  - Let total renders = 4
  - Pixels rendered by Renderer id 0 : 0, 4, 8 ...
  - Pixels rendered by Renderer 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)

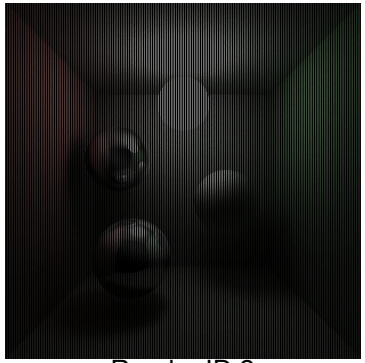


Render ID 0

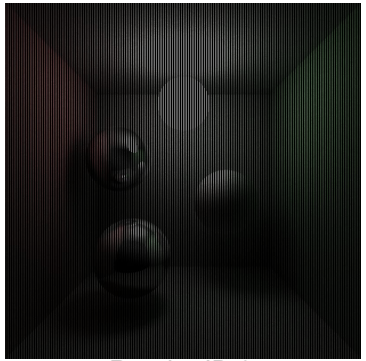
+



Render ID 1

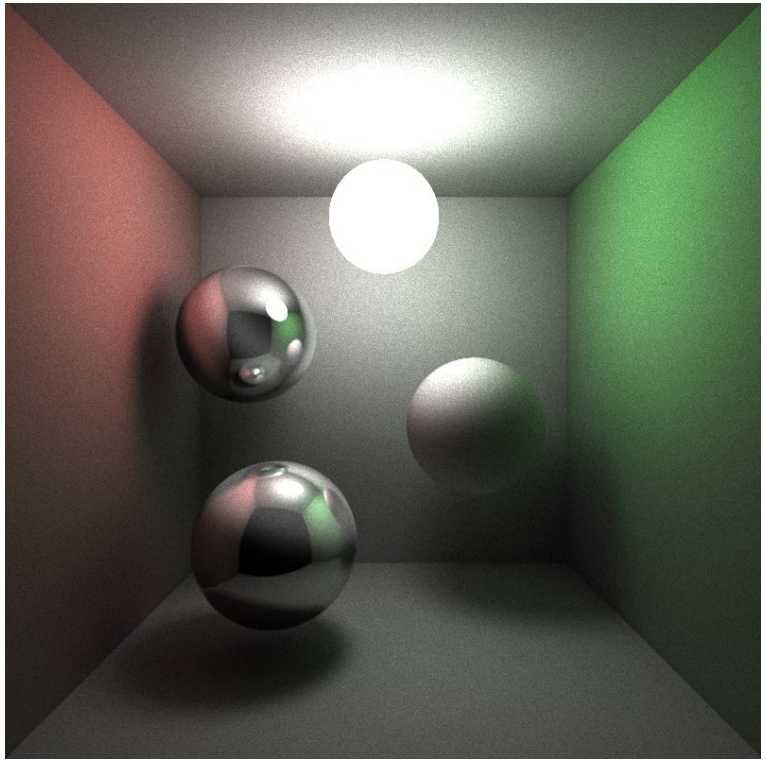


Render ID 2



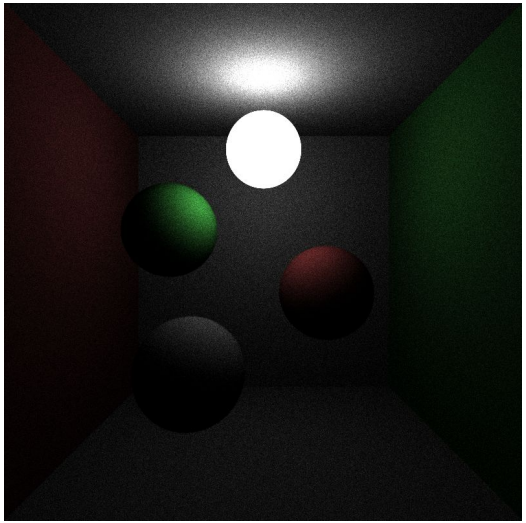
Render ID 3

=



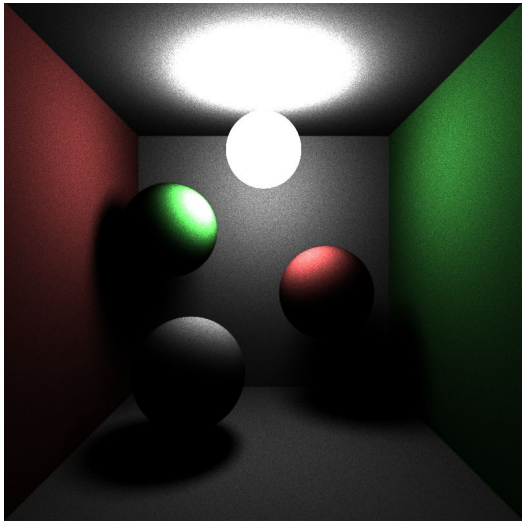
Full render

Screen Shots (BIS + LIS = MIS)



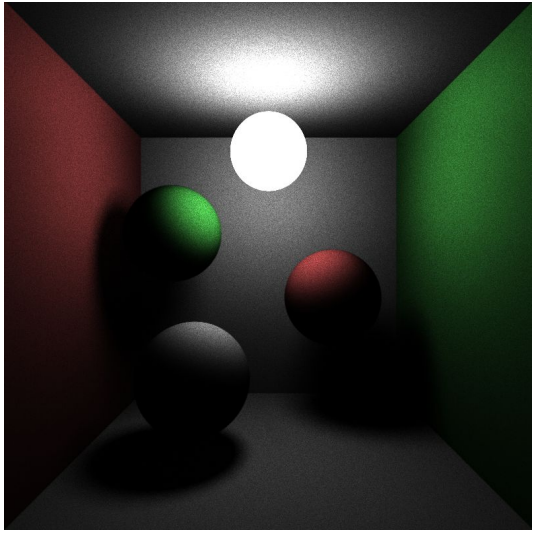
BRDF Importance sampling (FIXED)

+



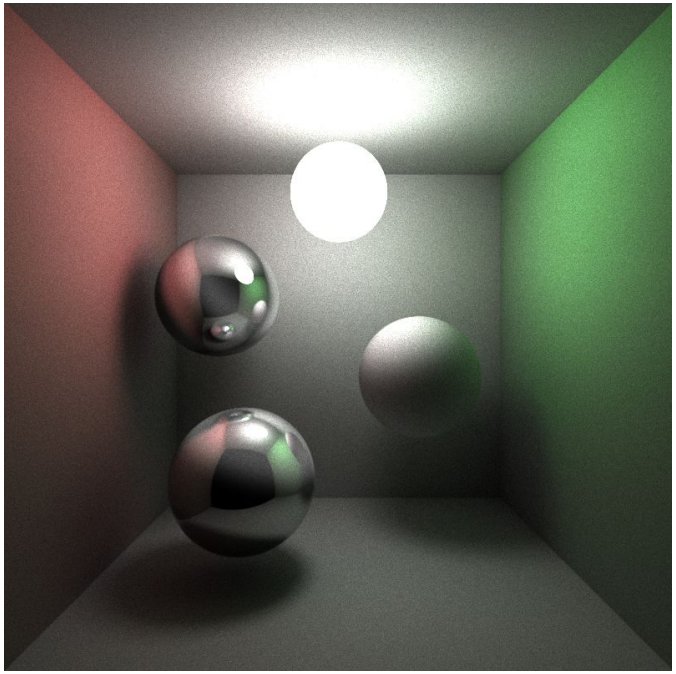
Light Importance sampling (FIXED)

=

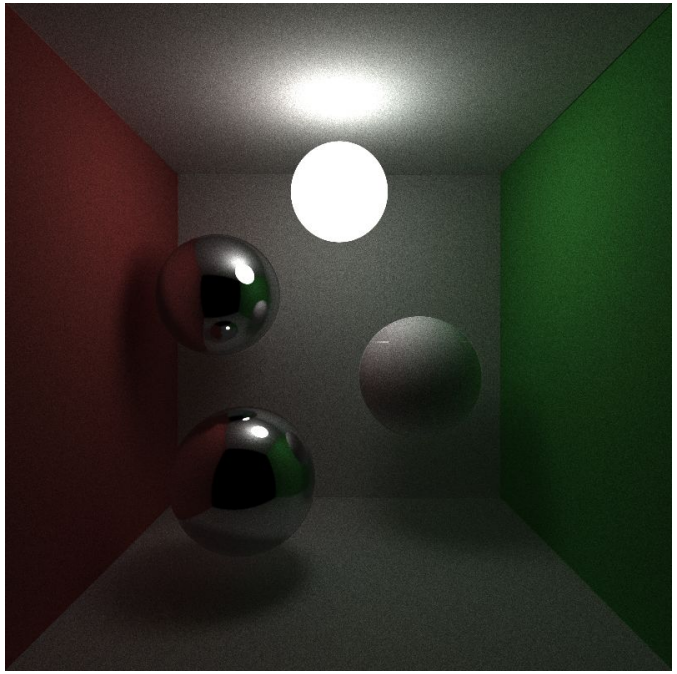


Multiple Importance sampling [Direct Illumination]

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)

Next

- Scene file transfer from viewer to leader to renderer.
- Camera navigation on viewer end
  - Adding camera control on the viewer.
- Material
  - Debug Refractive material implementation.
- Scene Setup
  - Setup more interesting scenes for the renderer.
- Performance analysis
  - Network delay testing vs one machine renderer.
  - Try and find the optimal number of iterations to send the image data.