AM207: Stochastic Optimization Methods

Monte Carlo methods for improved rendering

Tegan Brennan, Stephen Merity, Taiyo Wilson

Summary

Stochastic ray tracing is one of the fundamental algorithms in computer graphics for generating photo-realistic images. We implement an unbiased Monte Carlo renderer as an experimental testbed for evaluating improved sampling strategies. Our results show that the improved sampling methods we use for rendering can give comparable image quality over twenty times faster than naive Monte Carlo methods.

Approach

- Rendering is characterized by the light transport equation [Kajiya, 1986] which states the radiance at a point is the sum of the self-emitted radiance and the reflected radiance.

$$\begin{aligned} L_{out}(x,\theta_o) &= L_{emit}(x,\theta_o) + L_{reflected}(x,\theta_o) \\ &= \text{emitted from obj} + \text{reflected onto obj} \end{aligned}$$

- As the integral $L_{reflected}(x,\theta_o)$ is over all possible reflected light rays, it is impossible to compute exactly. We approximate this using Monte Carlo methods.
- For a high definition 1080p image, the radiance of over 2 million pixels needs to be computed, requiring computing potentially billions of light paths. Ray traced frames in Pixar's *Cars* took 15-24 hours to render [Christensen et al., 2006].

Our goal: Can we minimize noise and improve speed through intelligent sampling methods?

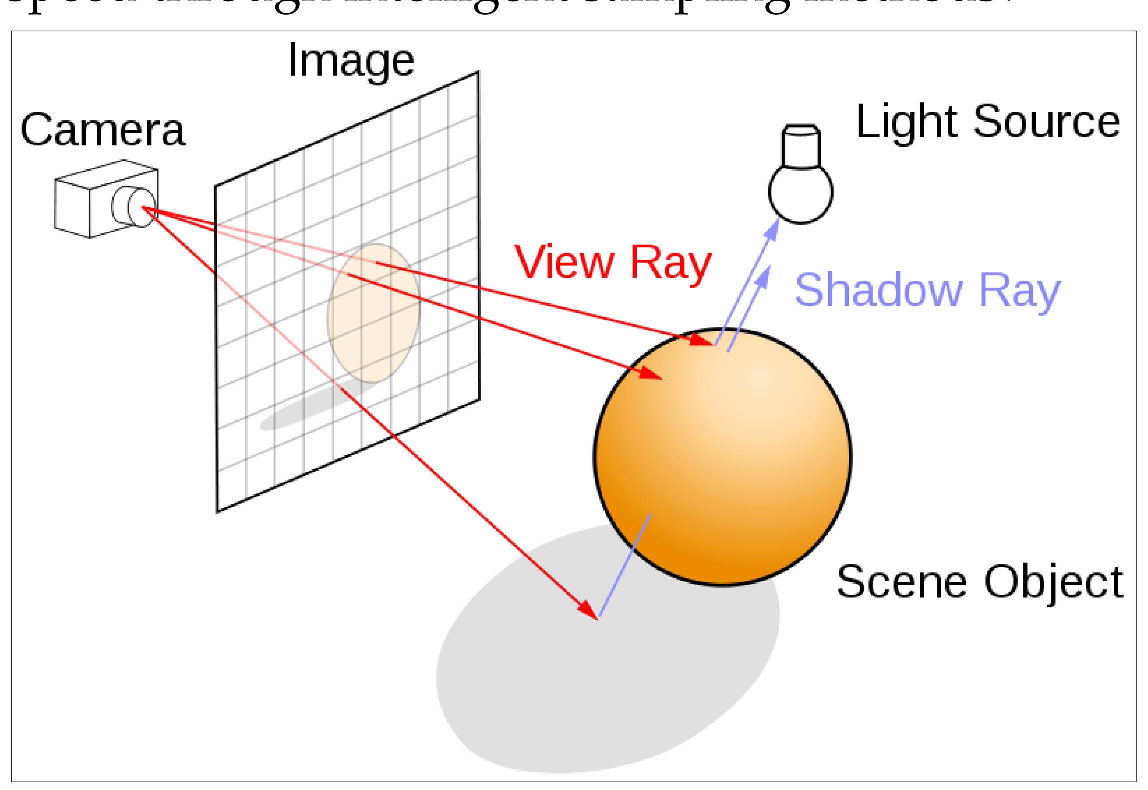

Figure 1: Illustration of ray tracing

Methods

BRDF sampling

- When light hits a surface, it bounces away from it unless completely absorbed. Working out the probability that it bounces in a given direction is a core question in accurate rendering.
- Perfect mirrors have highly defined reflection directions, but diffuse surfaces (such as paper) have far more widely distributed reflection directions.
- Importance sampling uses the bidirectional reflectance distribution function (BRDF) of the material rather than scaling by probability.

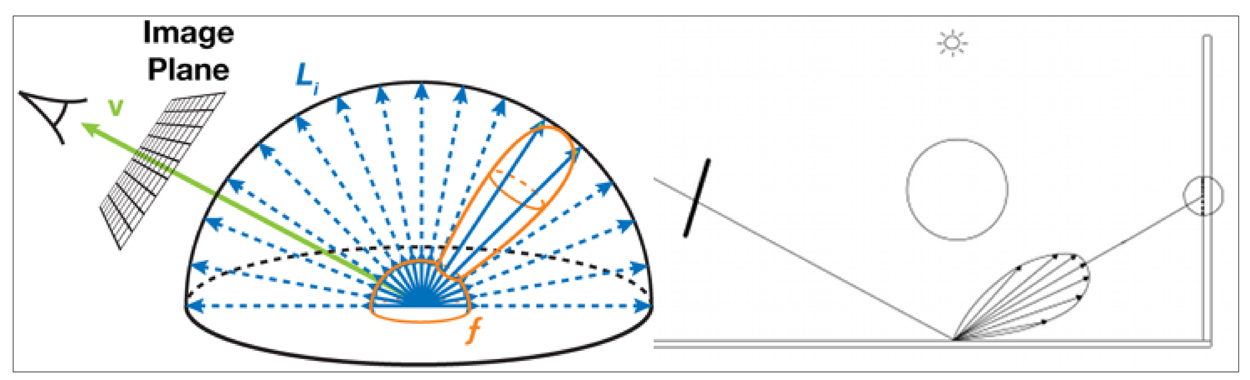

Figure 2: Uniform sampling of the potential bounce directions (blue) versus BRDF sampling (orange).

Explicit light sampling

- When the light source is small, the probability that a random light path will hit it is unlikely, requiring a larger number of light paths to be sampled.
- By exploiting our knowledge of the scene, specifically the location of these luminaires, we can perform importance sampling towards the location of the light sources.

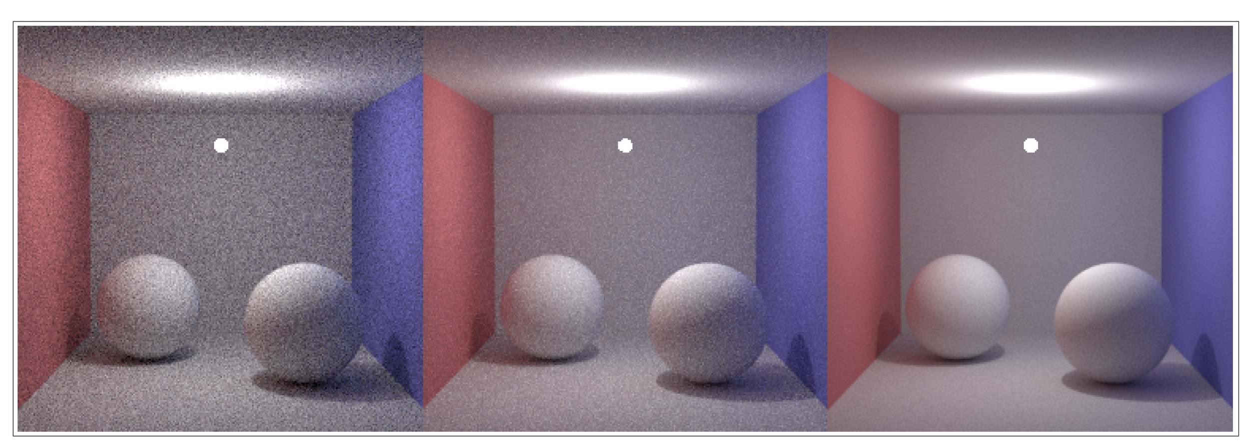

Figure 3: The image on the left used implicit emitter sampling and took 2,583 seconds. The images on the middle and right used explicit emitter sampling and took 9 and 101 seconds respectively.

Rejection sampling via Russian roulette path termination

- Bright photons in real life may bounce millions of times, but computing such a long path is intensive and contributes little to the final result.
- Stopping after a finite number of bounces can result in significant bias however.
- Russian roulette path termination fixes this by introducing a path termination probability q at each step, where the probability depends on the reflectance properties of the material currently being sampled.

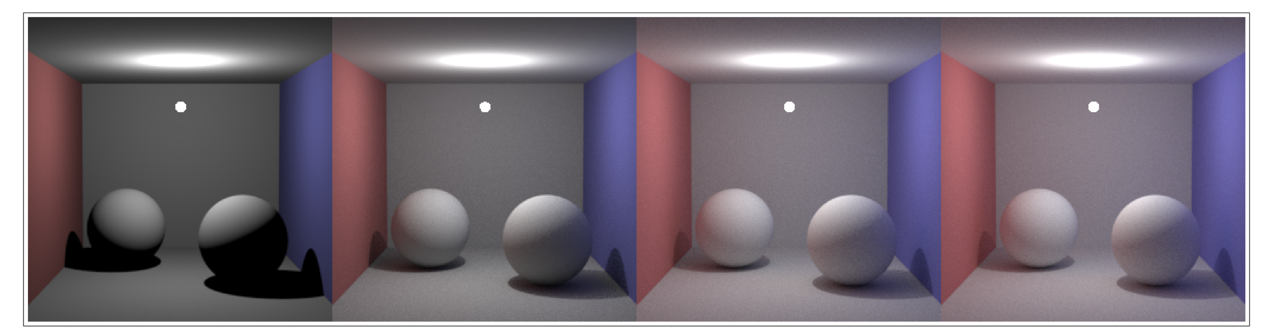

Figure 4: Finite path termination of 1, 2, and 3 bounces leads to bias compared to Russian roulette sampling.

Physical phenomena

Monte Carlo methods allow easy emulation of physical phenomena such as camera lenses, including aperture size and focal length.

- Small apertures result in highly collimated (parallel) rays sent into the scene, resulting in a "flat" image. Large apertures send spread out rays into the scene, resulting in a large field of view.
- For each pixel of our image, rays are sent at angles so that they converge at the focal length. Parts of the image not in this focal plane will be blurred, while those that are will appear sharper.

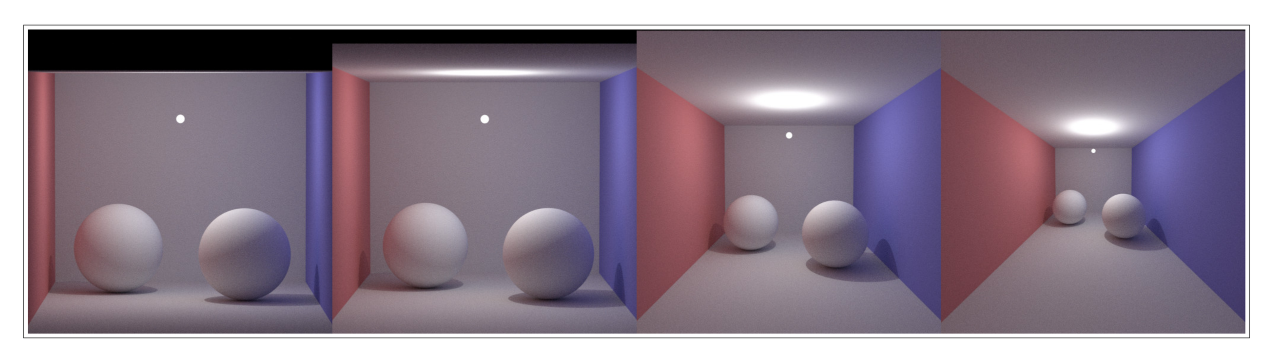

Figure 5: Aperture size affects the field of view. From left to right: $\frac{1}{4}$, $\frac{1}{2}$, 2 and 4 times normal aperture size.

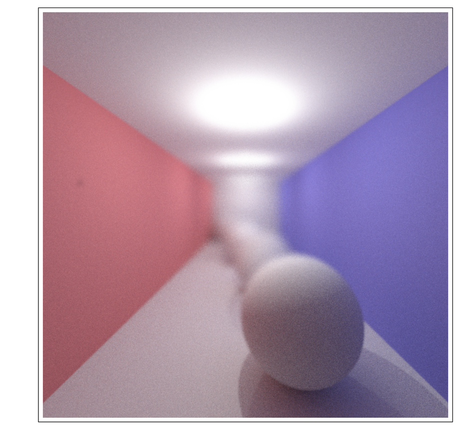

Figure 6: Shallow depth of field and short focal length result in only the front of the first ball in focus while the rest of the scene is blurred increasingly past this point. Both physical phenomena are intuitive modifications to the Monte Carlo rendering algorithm.

Future work

- Adaptive sampling of pixels based upon variance reduction techniques from the Monte Carlo literature.
- Engineering optimizations such as octrees for improving the speed of ray intersection calculations.
- More complex materials, such as mirrors, metallic surfaces and dielectric surfaces such as glass.

Conclusion

- By modifying the Monte Carlo ray tracer to use importance sampling and rejection sampling, we were able to substantially improve both the speed and quality of our rendered images without restricting the scenes that can be tackled.
- Our results show that these improved sampling methods give equivalent quality over twenty times faster than naive Monte Carlo methods.

References

Per H Christensen, Julian Fong, David M Laur, and Dana Batali. Ray tracing for the movie Cars. In *Interactive Ray Tracing 2006, IEEE Symposium on*, pages 1–6. IEEE, 2006.

James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.