Path Tracing

Brandon Simpson, Angus Read

Goals

- Understand the difference between Ray Tracing and Path Tracing
- Understand the different directional approaches to path tracing as well as their tradeoffs
- Understand some of the probabilistic techniques used to generate rays

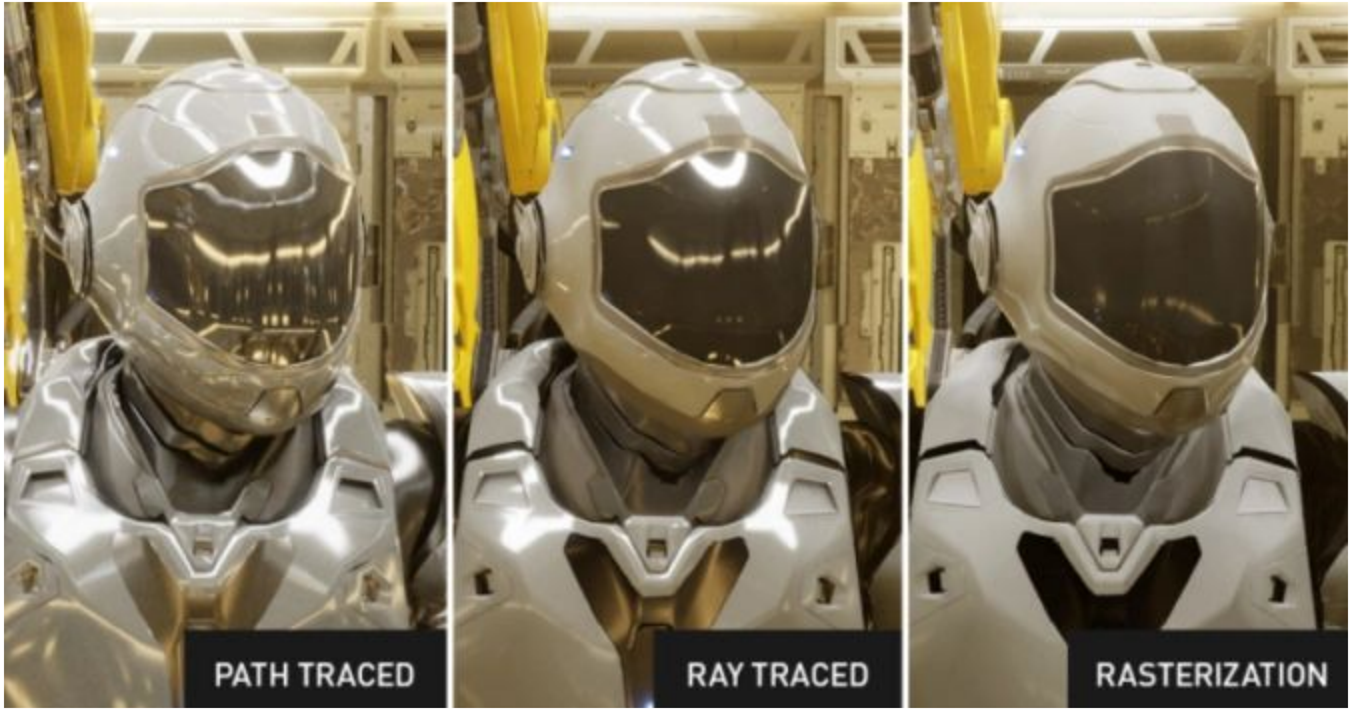
Ray Tracing Vs Path Tracing

Ray Tracing

- Traces a single ray as it reflects
- Deterministic Approach
- Ray origin at the camera eye

Path Tracing

- Generates N rays and gets the average color contribution
- Probabilistic Approach
- Ray origin at camera eye, light source, or both
- Models behavior of light rays to generate photorealistic renderings



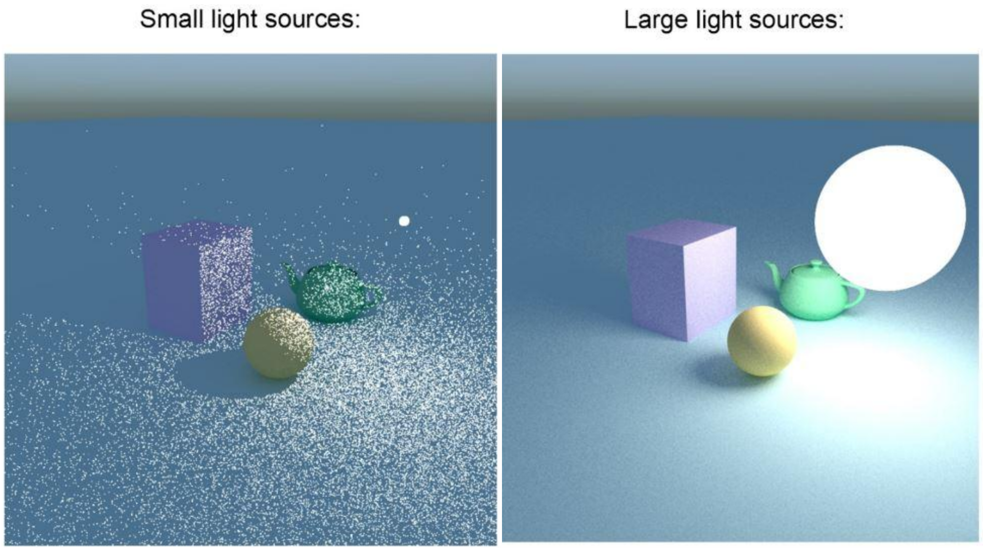
https://blogs.nvidia.com/blog/2022/03/23/what-is-path-tracing/


classic OpenGL Quake 2
Q2VKPT (Vulkan path-tracing)

Backwards Path Tracing

Backwards Path Tracing

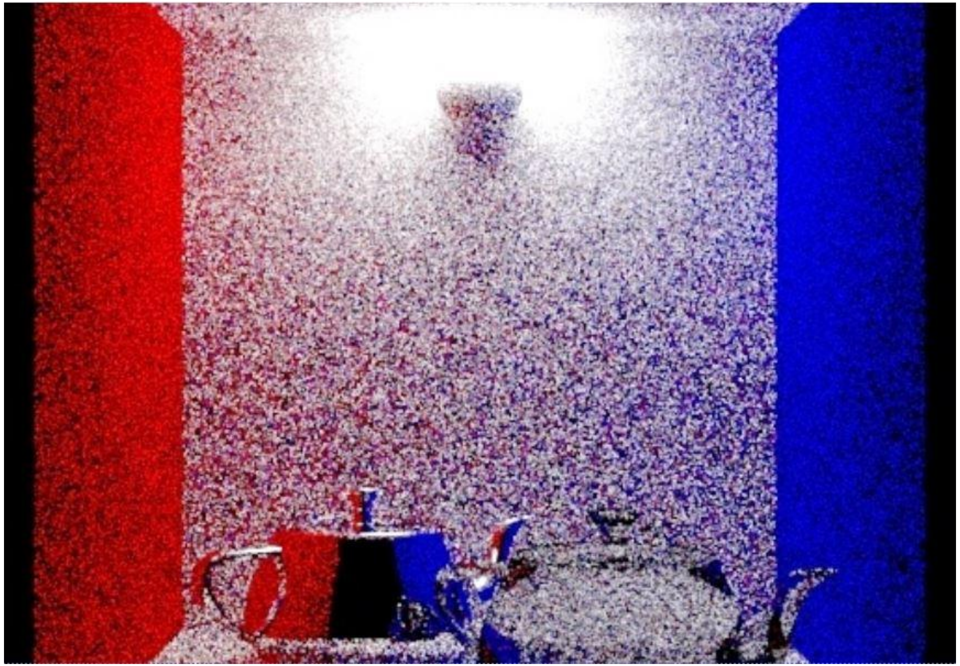
- Ray originates at the camera eye
- Reflects around scene until it encounters light source
- Ensures that all rays will reach the eye
- **Problem**: what if the light source is small?



Forward Path Tracing

Forward Path Tracing

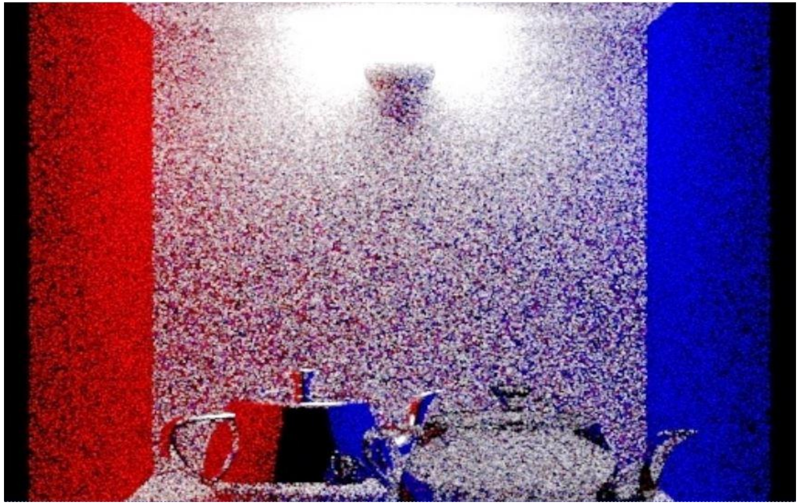
- Ray originates at a light source
- Reflects around scene until it encounters the camera eye
- Produces the most photorealistic renderings
- **Problem**: wastes rays — not all rays will reach the camera eye



Forward Path Tracing, 2048 samples per pixel

Q: How can we capture the efficiency of Backward Tracing along with the accuracy of Forward Tracing?

A: Combine both approaches: Bidirectional Path Tracing



Forward Path Tracing, 2048 samples per pixel

Bidirectional Path Tracing, 600 samples per pixel

Bidirectional Path Tracing

- Combines forwards and backwards path tracing
- Retains the efficiency of backwards tracing by not wasting rays as well as the accuracy of forward tracing by modeling the path of light particles
- Each sample is forward **and** backward traced
- Intermediate vertices on each path are connected
- Converges faster than unidirectional but requires more work

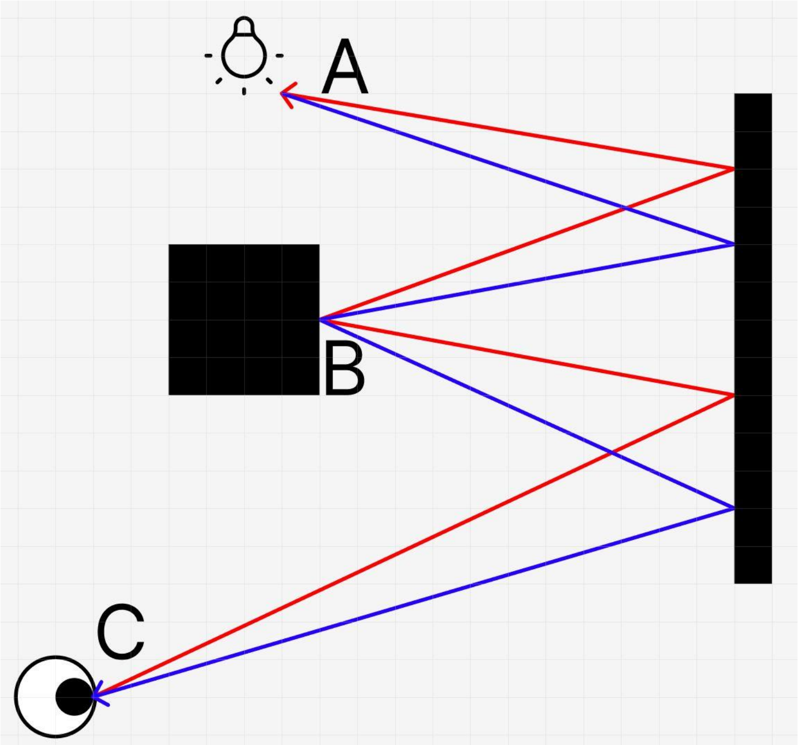
Path Tracing Approaches

Forwards Tracing Path

Backwards Tracing Path

Bidirectional Path Tracing would stop reflecting after intersecting intermediate vertex **B**

Backwards/Forwards Tracing are special cases of this; the intermediate point is the first vertex in the path – vertices **A** and **C** respectively



Q: What is the problem with backwards tracing scenes with small lights?

A: There is no problem – rays originate from the light.

B: The rays are less likely to intersect the light.

C: It wastes rays that do not intersect with the eye.

D: ¯_(ツ)_/¯ IDK.

Limitations

- Interior scenes
  - Rays take longer to escape the environment
- Many Rays are needed

Solution

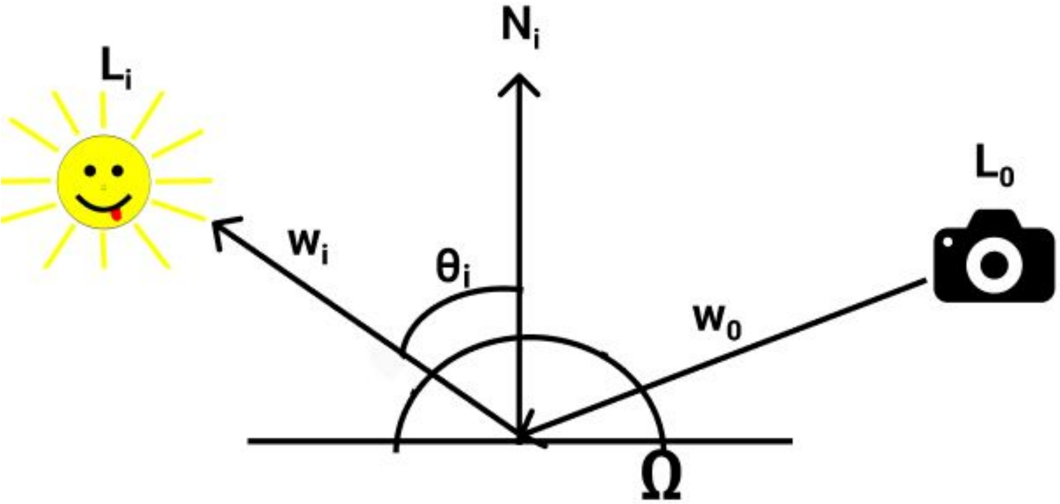
- Russian Roulette:
- Terminate path with probability inverse to contribution
  - Paths that aren't likely to contribute to the scene terminate sooner
- Boost non-terminated paths by their probability to be terminated

Rendering Equation

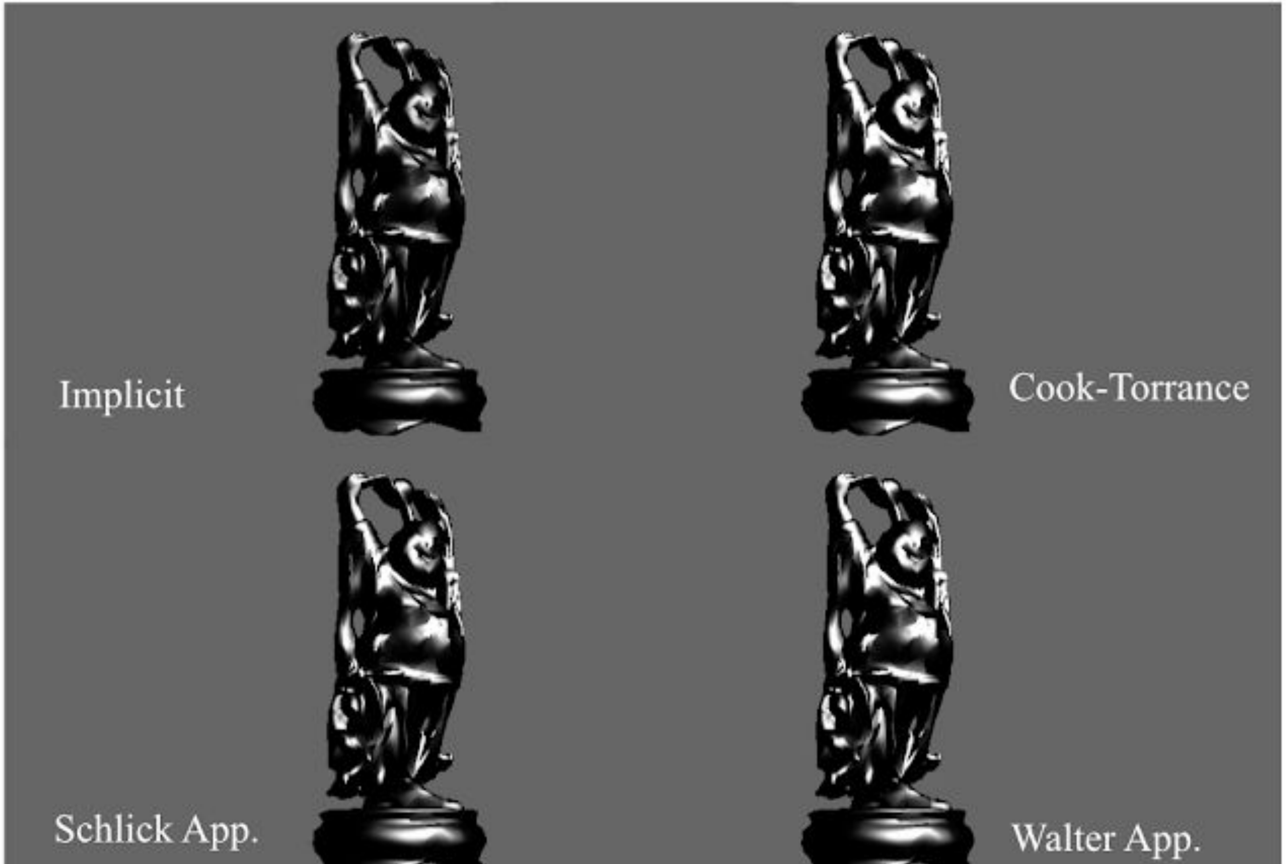
$$L_0(w_0) = \int_{\forall\Omega} L_i(w_i)cos(\theta_i) f_r(w_i, w_0)dw_i$$

Geometric Terms

BRDF



Different Geometry Terms



Bidirectional Reflectance Distribution Function (BRDF)

- Function of incoming/outgoing directions
- Models how light interacts with material
- Should uphold conservation of energy
    - Outgoing radiance <= incoming radiance
- Rotationally symmetric about the normal
- BRDF is the same if incident and reflected light are reversed

$$f_r(w_i, w_0)$$

Q: What does it mean for a BRDF to obey energy conservation?

A: The integral of the BRDF sums to 1

B: Incoming Radiance <= Outgoing Radiance

C: Incoming Radiance == Outgoing Radiance

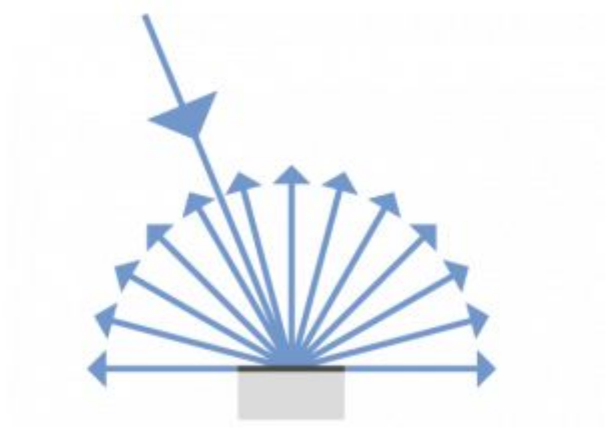
D: Outgoing Radiance <= Incoming Radiance

Lambertian BRDF

- Radiance is proportional to the cosine of the angle between the illuminating source and the normal
- Radiance is uniform over the surface
- Each scatter direction is equally likely

<u>Calculation</u>

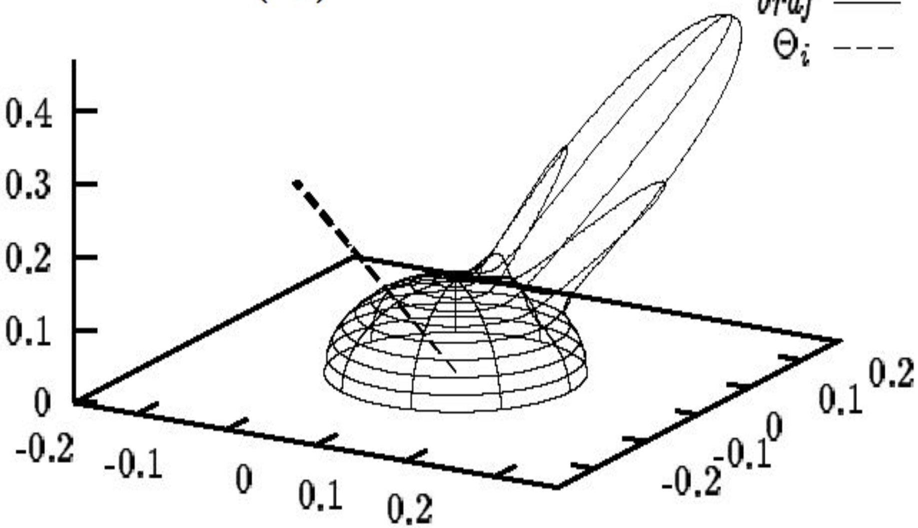
- Create an orthonormal basis using surface normal
- Generate a sample over the hemisphere
- Convert to world space



Blinn-Phong BRDF

$$f_r(w_i, w_o) = k_d + k_s \cdot \frac{cos^n(\alpha)}{cos(\theta_i)}$$

- α = angle between perfect specular reflection and outgoing ray
- n=specular exponent
- Kd = amount of energy reflected diffusely
- Ks = fraction of perpendicular energy reflected specularly
- Energy Conservation -> kd+ks<1



Choosing Directions Probabilistically

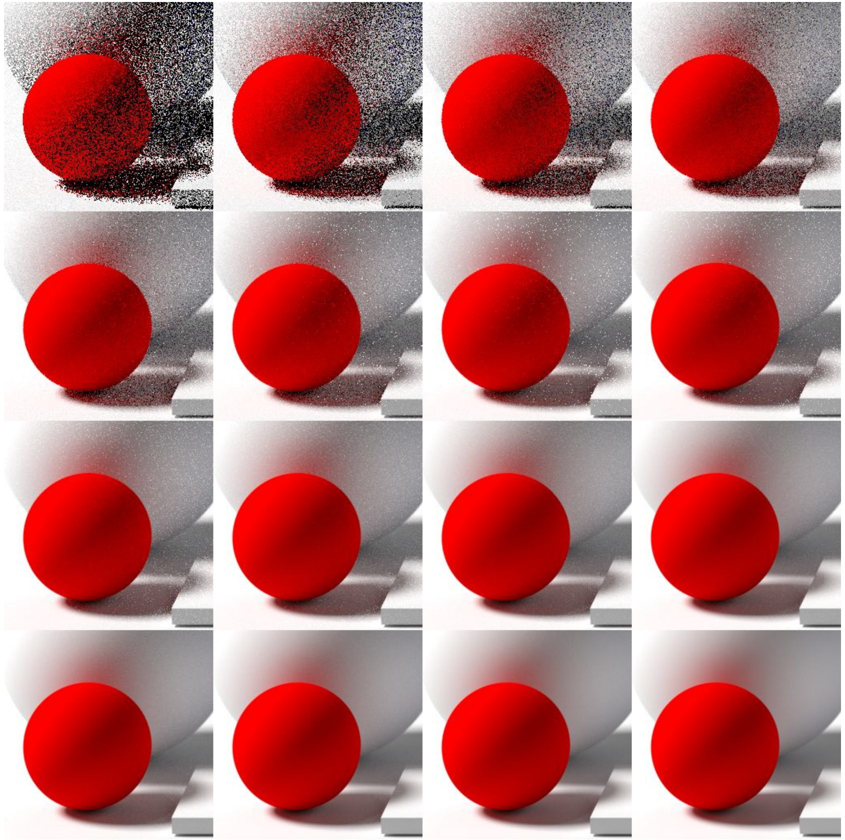
Naive Method

- Sample Uniformly
  - Lots of wasted calculations
- Importance Sampling
- BRDF tells us how incoming light will scatter
- Can we sample proportional to BRDF?

Inverse Transform Method

1. Calculate the PDF
2. Calculate CDF
3. Invert CDF
4. Sample std uniform
5. Plug sample into inverted CDF
6. Convert To World Coordinates

Sample Size Matters!



Using Less Samples

- Sampling Better Directions
- More Advanced Monte Carlo Methods
- Reducing noise to photorealistic levels can take a lot of samples

Post Processing

- Can be faster and utilize information known after sampling
- Denoising
- Machine Learning

Low sample count results in noisy images

Coherency

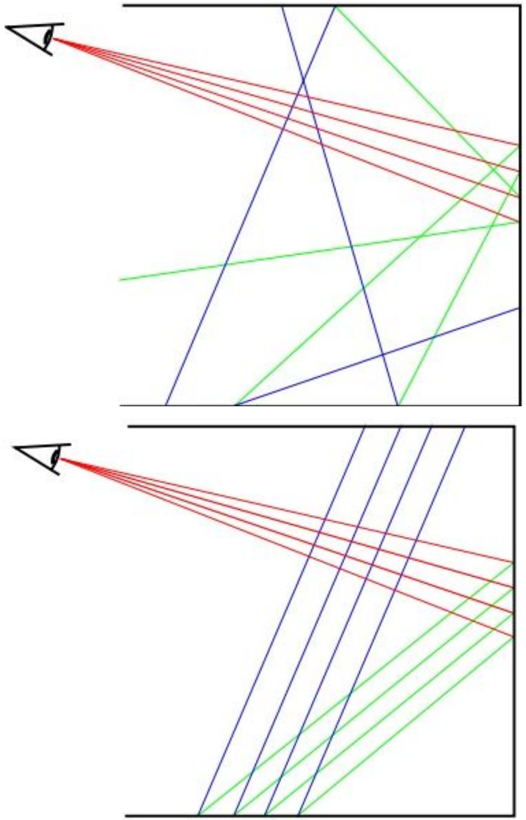
- Classify rays based on their position and orientation
- Spatially grouped random sample

<u>Pros</u>

- More likely to hit similar objects and textures
- Better cache locality

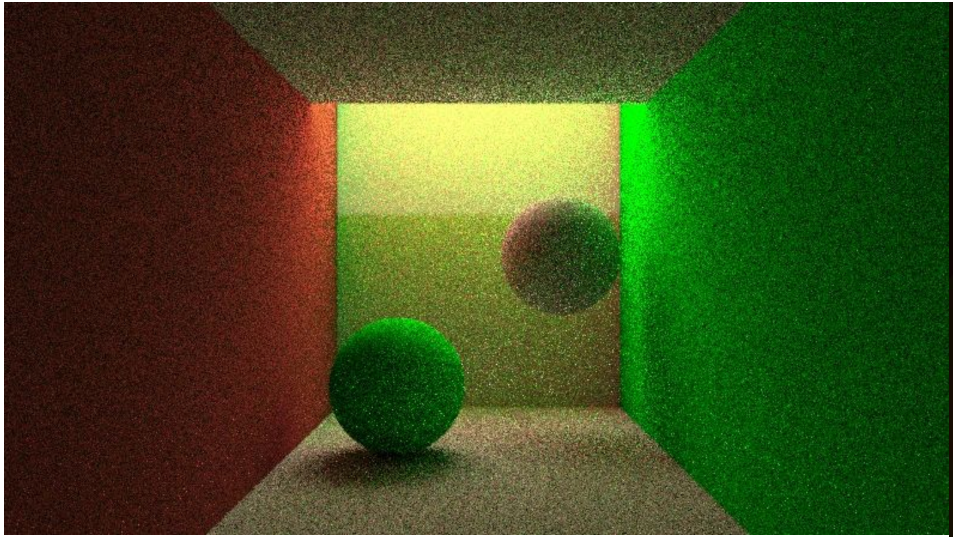
<u>Cons</u>

- Difficult for dynamic scenes

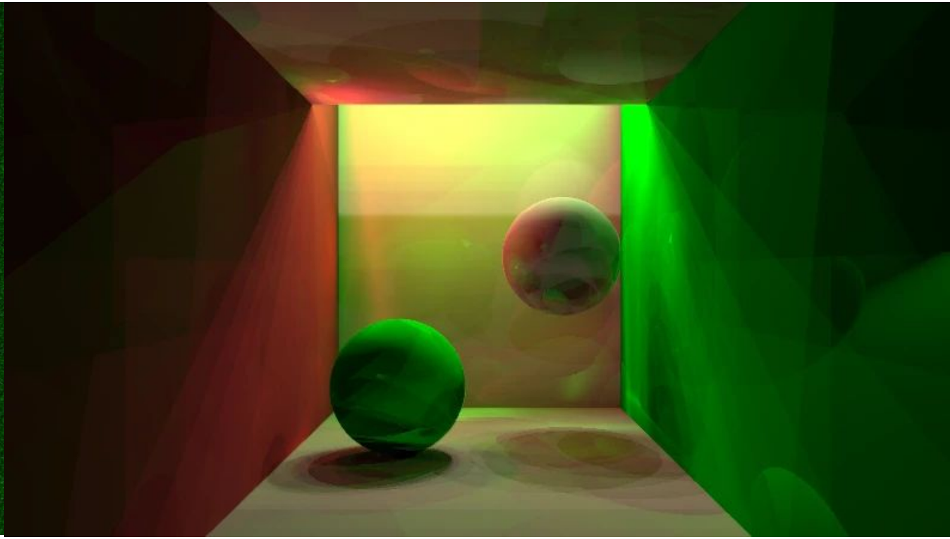


Coherency

Incoherent=random noise



Coherent=structural noise

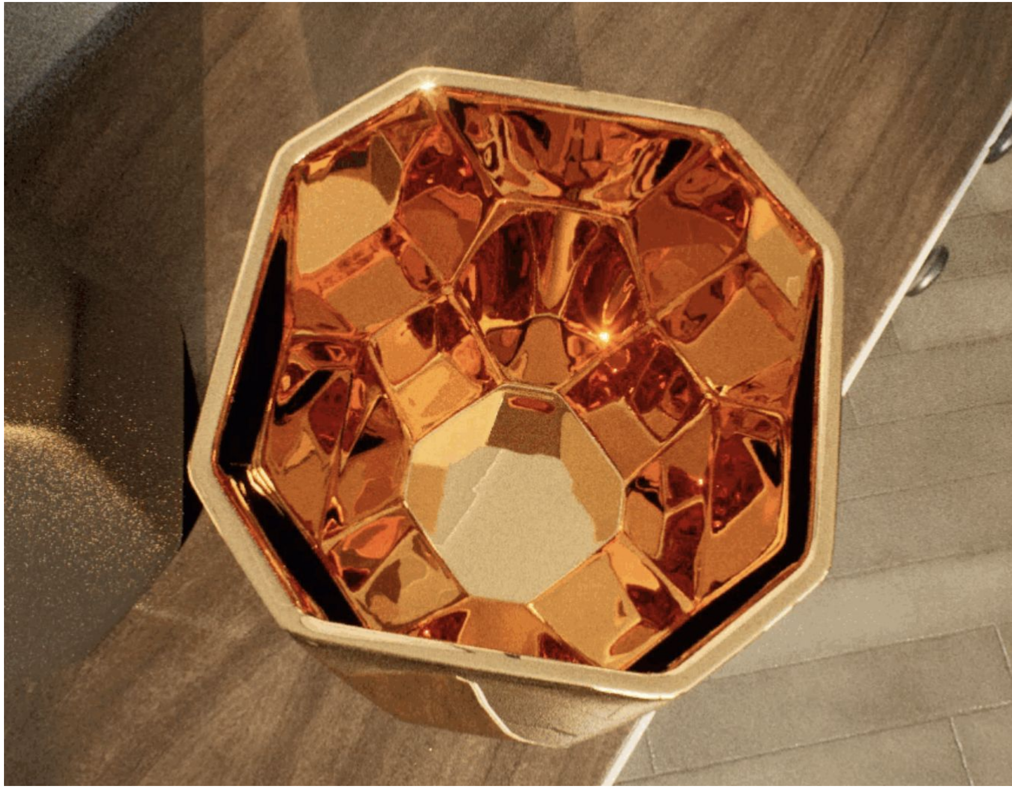


Fireflies

- Anonymously bright single pixels
- Result from numerical instability solving the rendering equation

<u>Mitigation</u>

- Clamp Radiance
- Reject high variance samples
- Fix it in post



More Advanced Methods

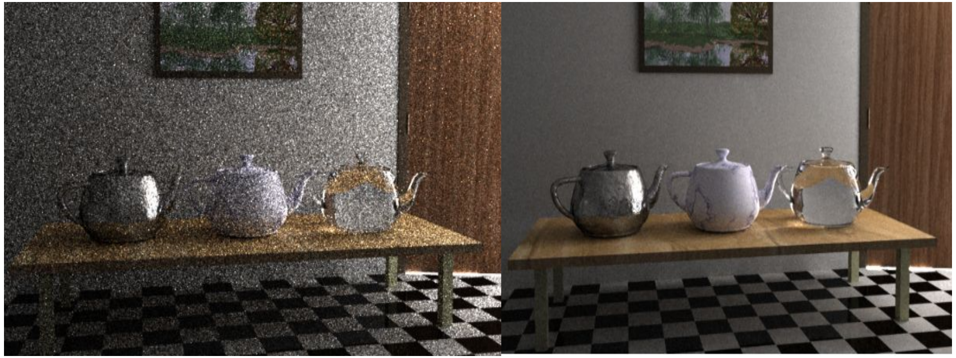
[Metropolis Light Transport]

- Uses bidirectional paths from eye to light
- Once path is found explore nearby with slight path perturbations

[Energy Redistribution Path Tracing]

- Uses path mutations to redistribute the energy of the samples over the image plane.

Bidirectional, n=40 MLT



Machine Learning Approaches

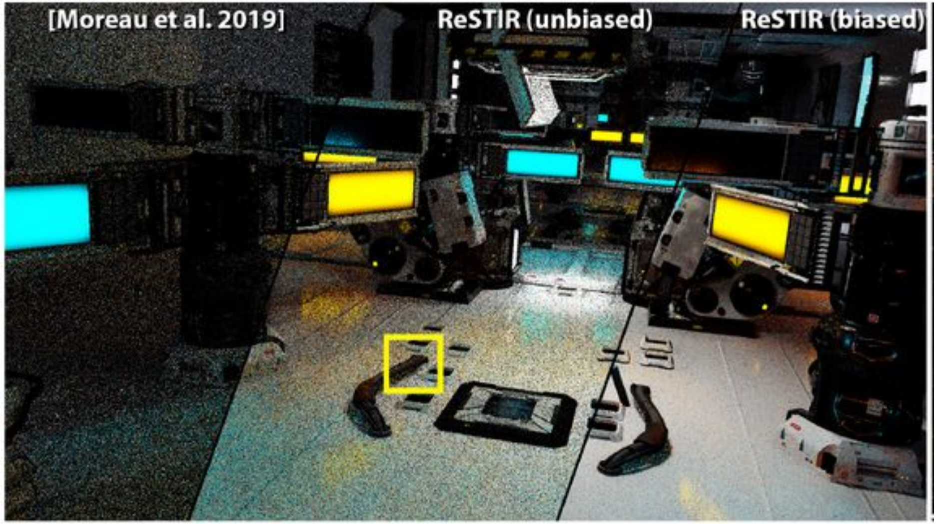
Neural Temporal Adaptive Sampling

- Learn to place more samples in
  - Non-occluded regions
  - Specular highlights

Adversarial Monte Carlo Denoising

- GANs learn to distribution from monte carlo path traced images

Spatiotemporal Reservoir Resampling Restir



https://cs.dartmouth.edu/wjarosz/publications/bitterli20spatiotemporal.html

Questions?