Introduction to Metropolis Light Transport

An Intuitive Approach

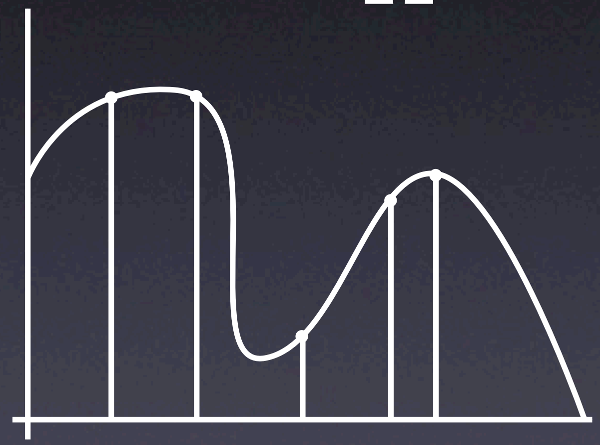
Mark Colbert

Overview

- Monte Carlo
  - Variance Reduction
  - Metropolis Sampling
- Path Tracing
- Metropolis Light Transport

Monte Carlo

We want to solve this: $\int_{\Omega} f(x)dx$



$$\int_{\Omega} f(u)du = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$$

Probability Density Functions

- Traditional Monte Carlo
  - Use Uniform Probability

$$p(x) = \frac{1}{b-a}$$

- Reduce variance by finding a density function that is close to the real function

$$p(x) \propto f(x)$$

Importance Sampling

- Extreme example illustrating how importance sampling helps

$$p(x) \propto f(x)$$

$$p(x) = cf(x)$$

Must normalize probability (something cannot happen more than 100% of the time)

$$c = \frac{1}{\int_\Omega f(u)du}$$

Importance Sampling

- Extreme example continued

$$\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} = \frac{f(X_i)}{cf(X_i)} = \frac{1}{c} = \int_{\Omega} f(u)du$$

Picking from a Distribution

- Choose random value, Xi, and solve for x.

$$\xi = \int_0^x p(u)du$$

(We call this a Cumulative Distribution Function)

Multi-Dimensional Monte Carlo

- Add integrals on top of integrals

$$\int_{\Omega_n} \cdots \int_{\Omega_2} \int_{\Omega_1} f(u_1, u_2, \ldots, u_n) du_1 du_2 \ldots du_n$$

$$\int_{\Omega} f(\mathbf{u}) d\mathbf{u}$$

Multi-Dimensional Monte Carlo

- Look at multi-dimensional variables as a vector of variables. Same techniques still applies!

$$\int_{\Omega} f(\mathbf{u})d\mathbf{u} = \frac{1}{N}\sum_{i}^{N}\frac{f(\mathbf{X_i})}{p(\mathbf{X_i})}$$

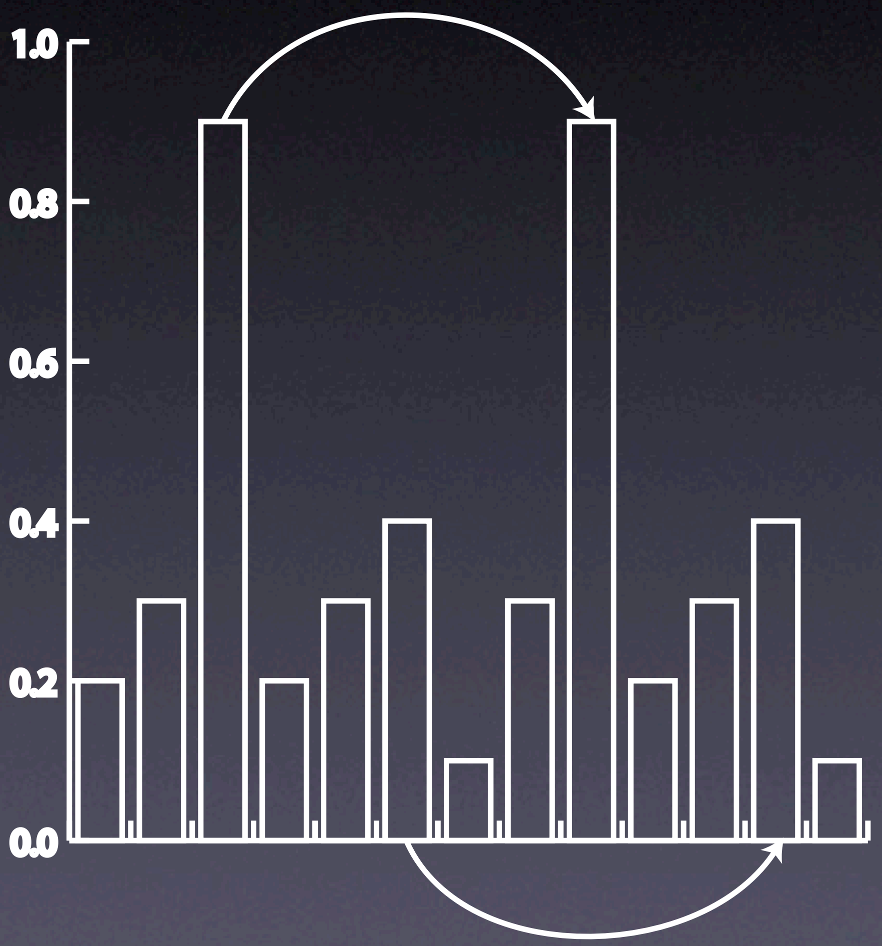
Variance Reduction

- Metropolis Sampling
  - Simple idea ~ Similar integrals should share samples

$$\Psi_i = \int_\Omega f_i(\mathbf{u}) d\mathbf{u}$$

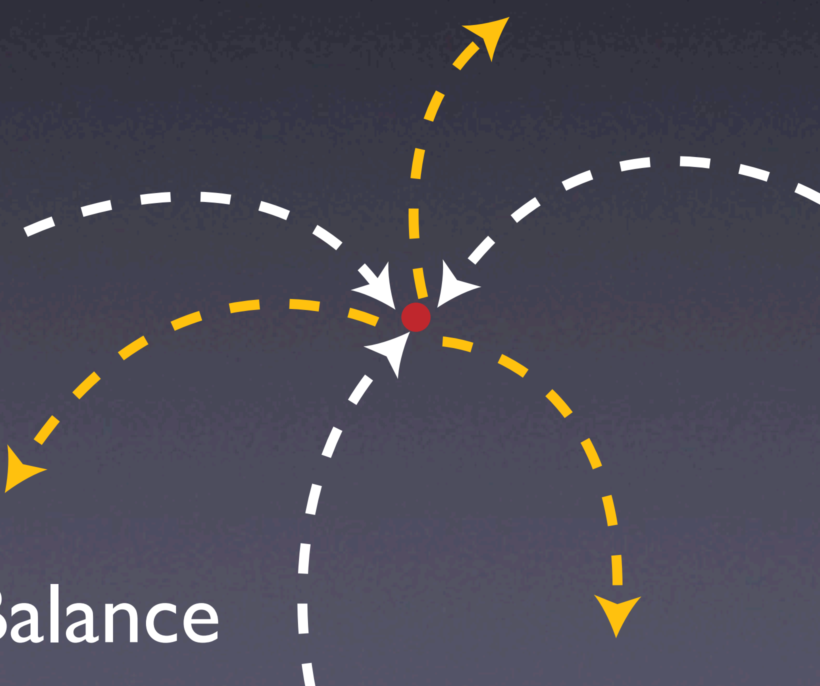
Metropolis Sampling

- Histogram view
- Move samples around for similar integrals



Metropolis Sampling

- How do we share samples without biasing (un-randomizing) result?
- We create a general balance between samples



General Balance

Maintaining Balance

- Transfer Probability

$$T(y \rightarrow x)$$

- Most difficult to choose transfer probability
- As long as general balance is held within the limit, the function is suited for Metropolis
- Want to find function that produces the best result, with the fewest samples.

Metropolis Sampling Algorithm

```
choose starting sample x

for i = 0 to mutations
    y = mutate(x)
    acceptance = y/x T(y › x)/T(x › y)
    if (randomReal(0,1) < acceptance)
        x = y

    deposit sample x
end for
```

Neat Example

```
choose random starting pos (x,y)
T(y › x)/T(x › y) = 1

for i = 0 to mutations
    npos = random pos
    acceptance = val[npos]/val[pos]
    if (randomReal(0,1) < acceptance)
        pos = npos

    add 1 at pos
end for
```

Originally Presented by Cline & Egbert

Neat Example

- Equal probability of transfer
- For mutation we just choose another pixel

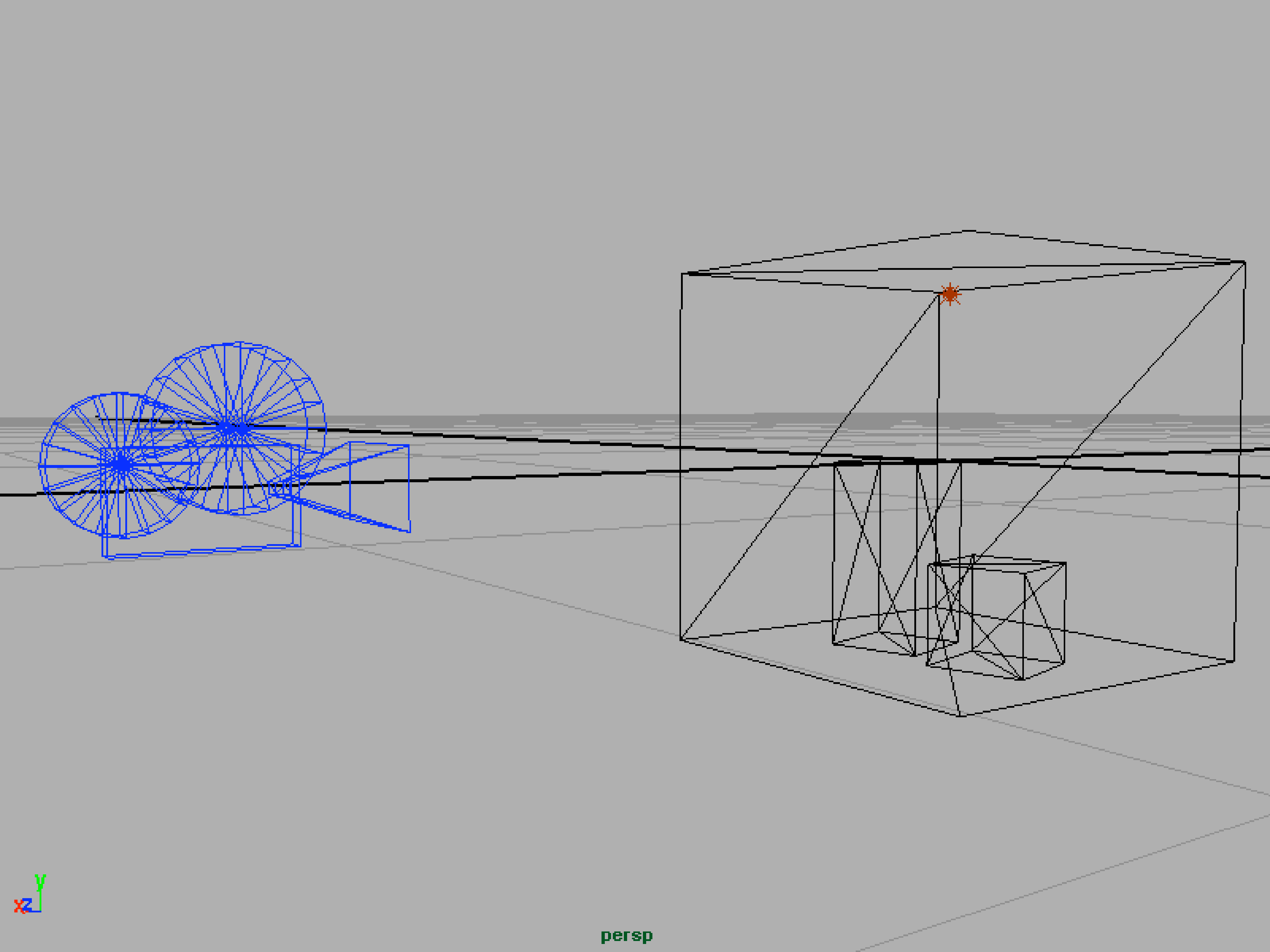
8 mpp

256 mpp

Neat Example

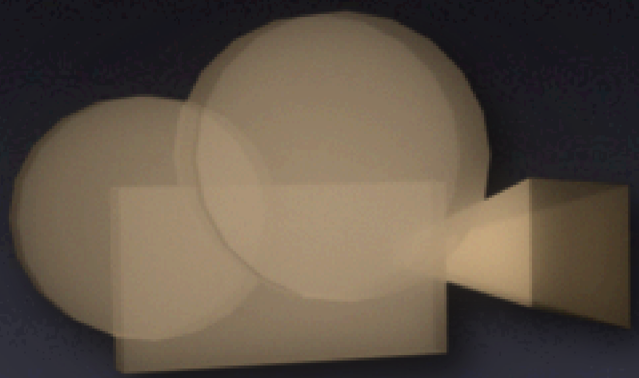
- We see that the more samples we transfer between the integrands (pixels) the closer we come to the real solution
- Shows MS is unbiased, as in the limit, it will properly distribute the samples across the integrands

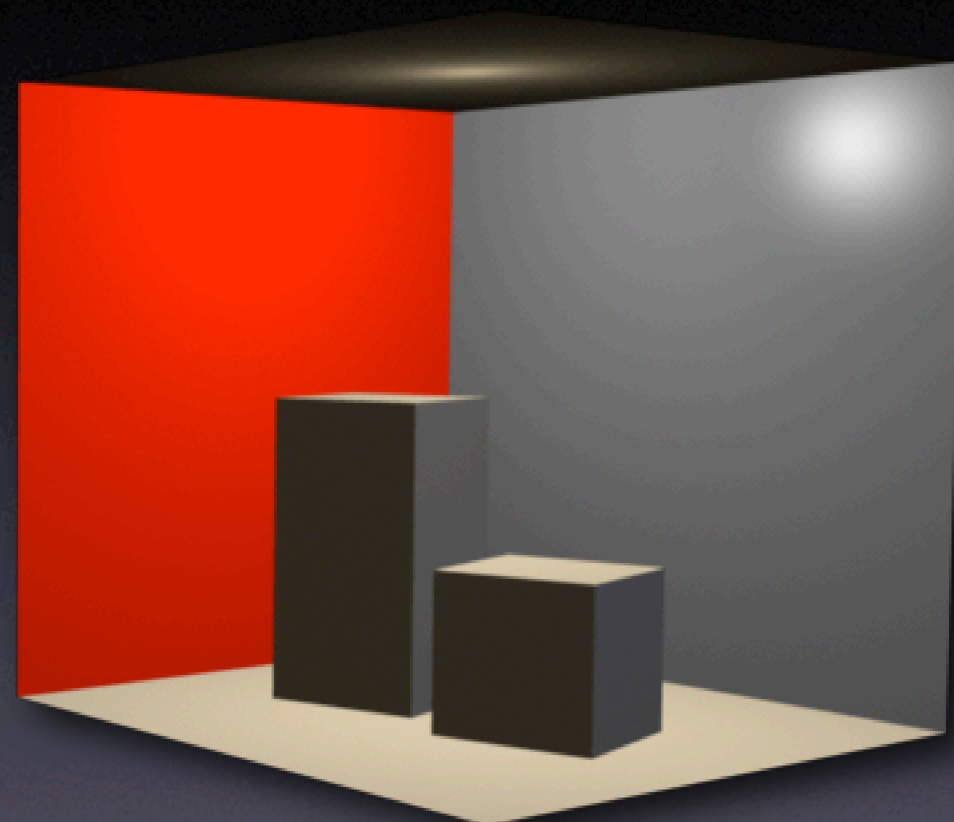
Path Tracing



persp

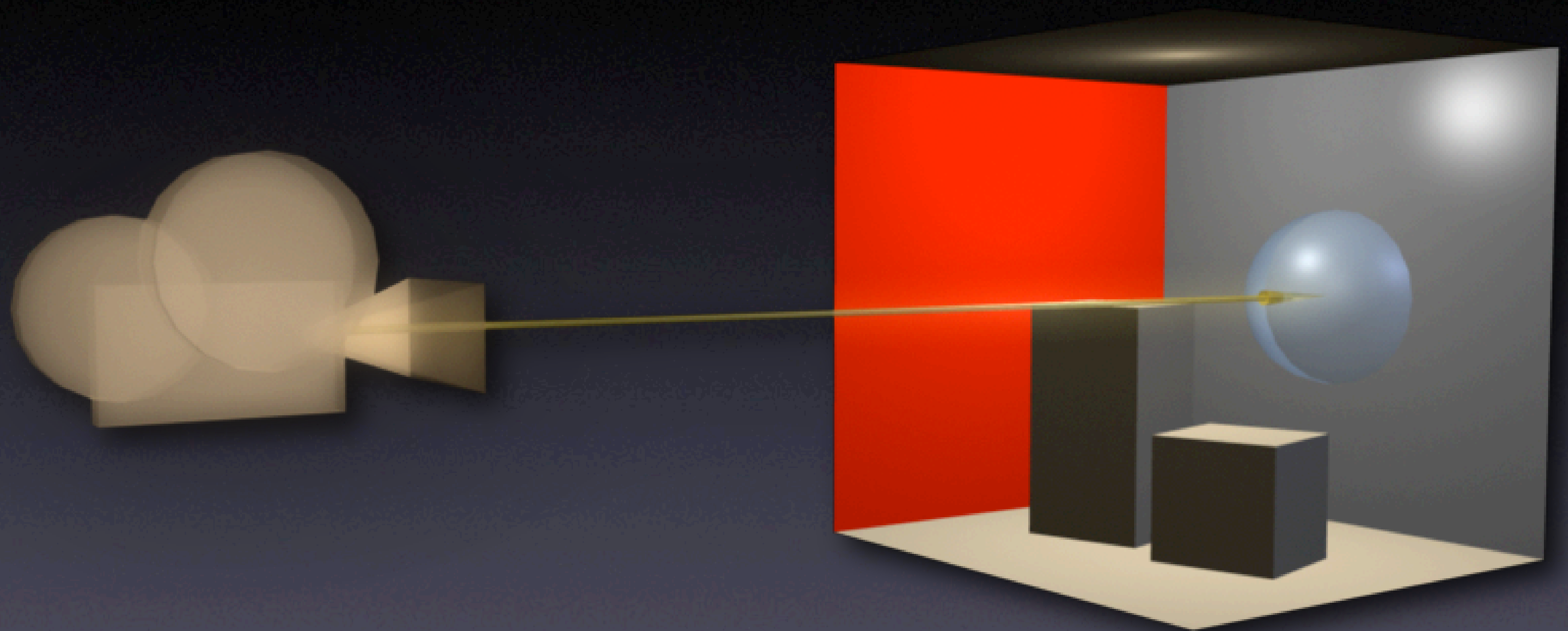
Pixel





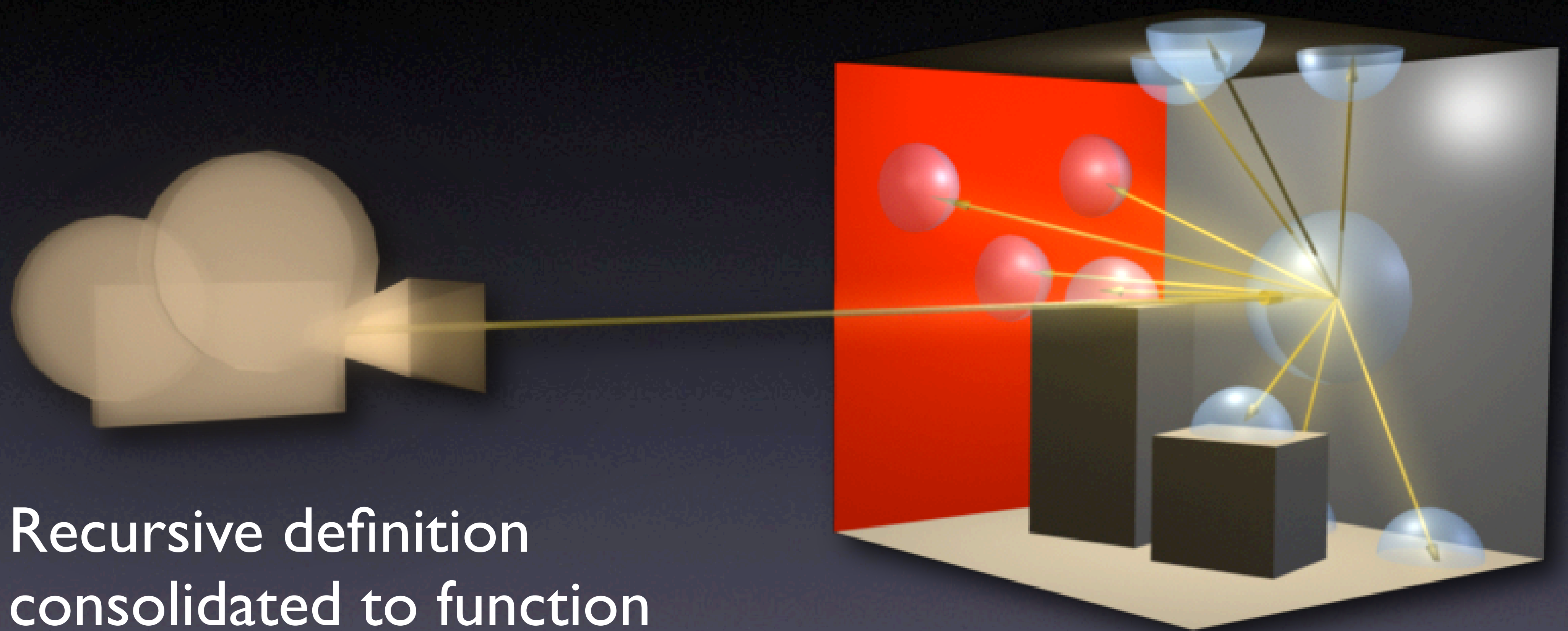
$$I_{i,j} = \int_A L_i(w_o)\,d\omega_o$$

Primary Ray



$$L_i(\omega_o) = L_e + \int_\Omega L_i(\omega_i) f(\omega_i, \omega_o) \cos\theta_i d\omega_i$$

Secondary Ray



Recursive definition consolidated to function of finite dimension, g

$$I_{i,j} = \int_A \int_{\Omega_1} \int_{\Omega_2} \cdots \int_{\Omega_n} g(\omega_o, \omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_n}) d\omega_{i_n} \ldots d\omega_{i_2} d\omega_{i_1} d\omega_o$$

Converting Paths to Monte Carlo

$$I_{i,j} = \int_A \int_{\Omega_1} \int_{\Omega_2} \dots \int_{\Omega_n} g(\omega_o, \omega_{i_1}, \omega_{i_2}, \dots, \omega_{i_n}) d\omega_{i_n} \dots d\omega_{i_2} d\omega_{i_1} d\omega_o$$

$$I_{i,j} = \int g(\mathbf{u}) d\mathbf{u}$$

$$I_{i,j} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{g(\mathbf{X}_i)}{p(\mathbf{X}_i)}$$

Path Tracing

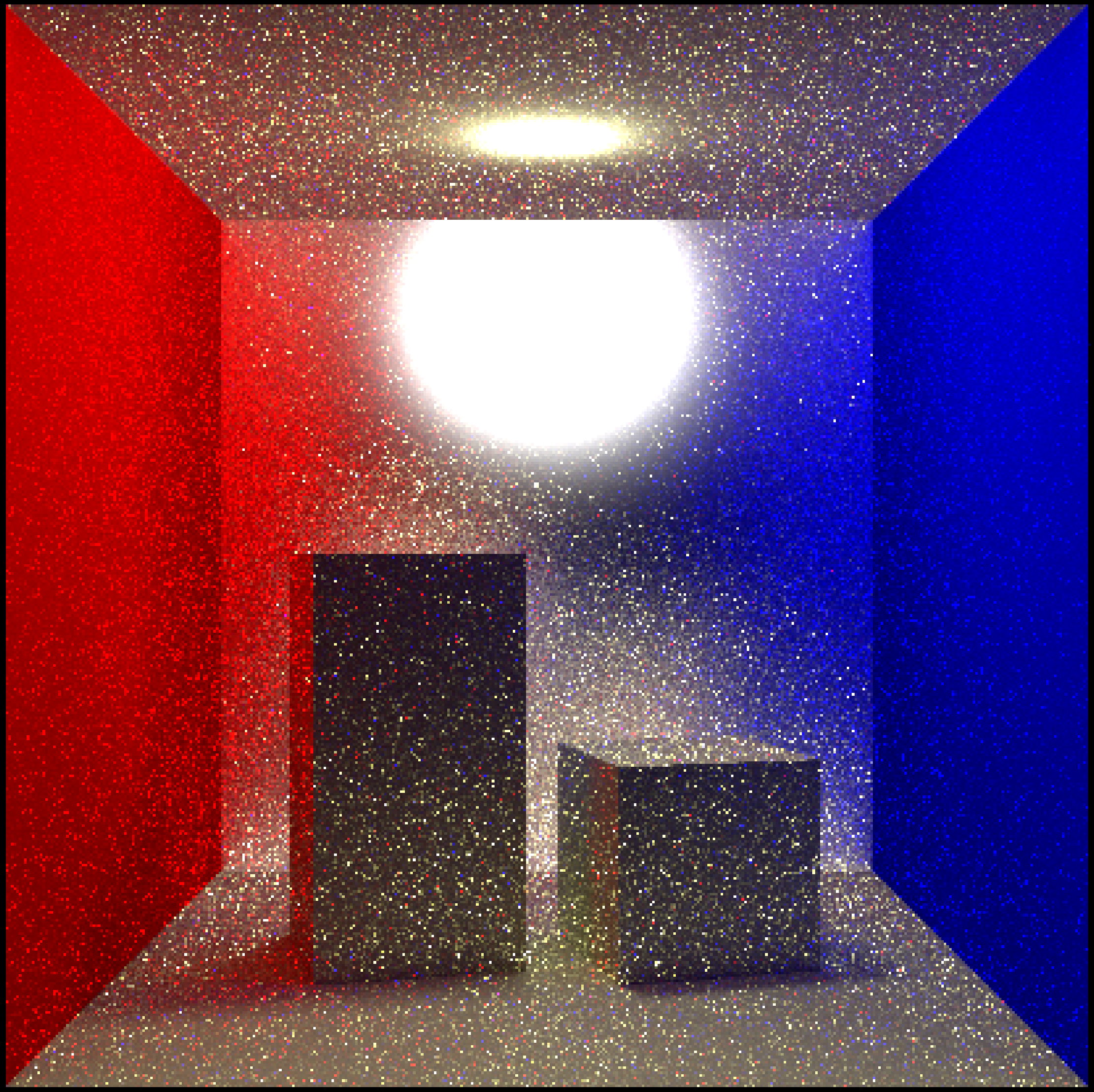
- PDFs are chosen based upon:
  - Material Properties
  - Choice of Light
- Russian Roulette
  - Path length (how many integrals are we going to solve)

Path Tracing

$$X_f = \frac{f_0(\omega_{i_0}, \omega_{o_0}) \cos\theta_{i_0}}{p_0(\omega_{i_0}, \omega_{o_0})} \cdot \frac{f_1(\omega_{i_1}, \omega_{o_1}) \cos\theta_{i_1}}{p_1(\omega_{i_1}, \omega_{o_1})} \cdot \ldots \frac{f_n(\omega_{i_n}, \omega_{o_n}) \cos\theta_{i_n}}{\pi d^2} \cdot \frac{V(x_n, x_{light}) L_e A}{p_{light}\, p_{len}(n)}$$

- Initially, solve for cosine weighted BRDF divide by probability of sample
- When connecting light, the probability of any direction is uniform, but varies depending upon the distance the path must travel
- Divide by the probability of the light and probability of the length of the path

Results



500 samples per pixel

Improvements to Path Tracing

- Path per wavelength of light
  - Birefringence
- Bi-directional Path Tracing
  - Start generating a path from both the eye and the light
  - Connect them in the middle
  - Further reduces variance

Metropolis Light Transport

- Surprisingly similar to Metropolis Sampling
- Now, the samples are paths
- Hard Part
  - Determining Transfer Probability
    - Changes in Path Density
  - Mutation Strategy

Mutation and Transfer

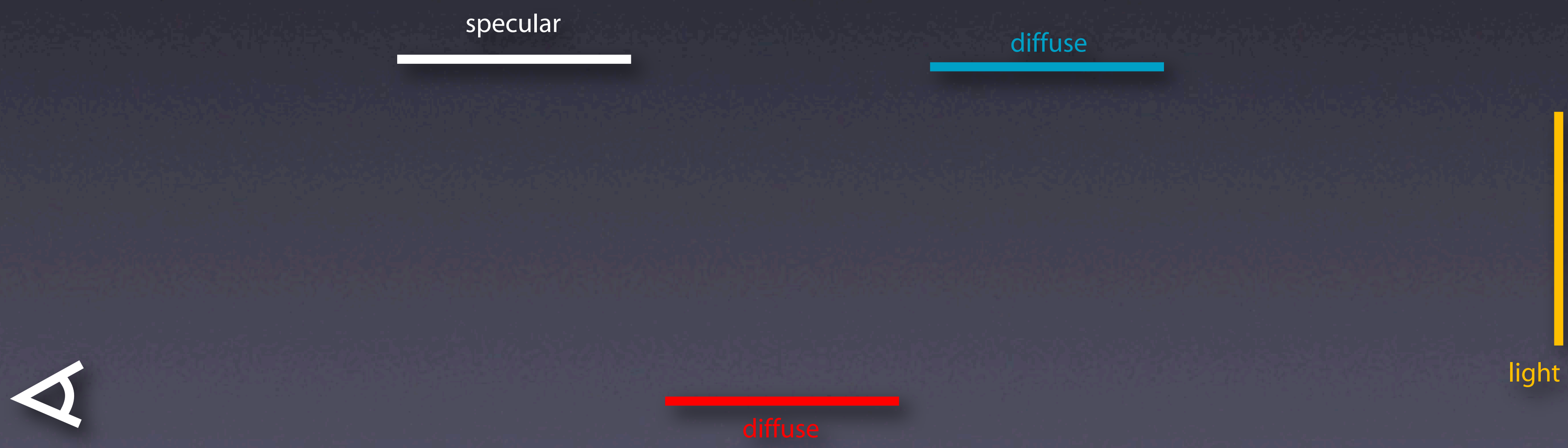
- Several mutation strategies
  - Lens Perturbation*
  - Caustic Perturbation
  - Random Path Mutation
- Many rules for how a mutation changes the path density

Lens Perturbation

- Take a path and start it in a new location on the image plane
- With the new location on the image plane, we now have a chain reaction of mutations

Chain Mutation

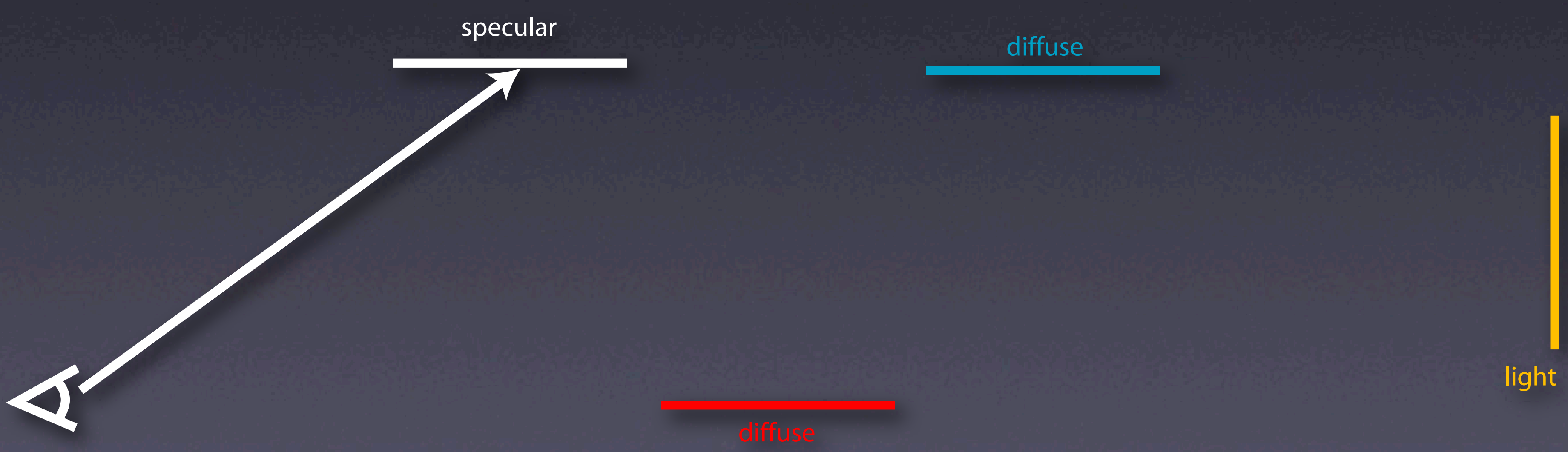
- Lets walk through the mutation of the following path
- Path X: LDDSE



Lens Perturbation

Equal probability that a new pixel is chosen

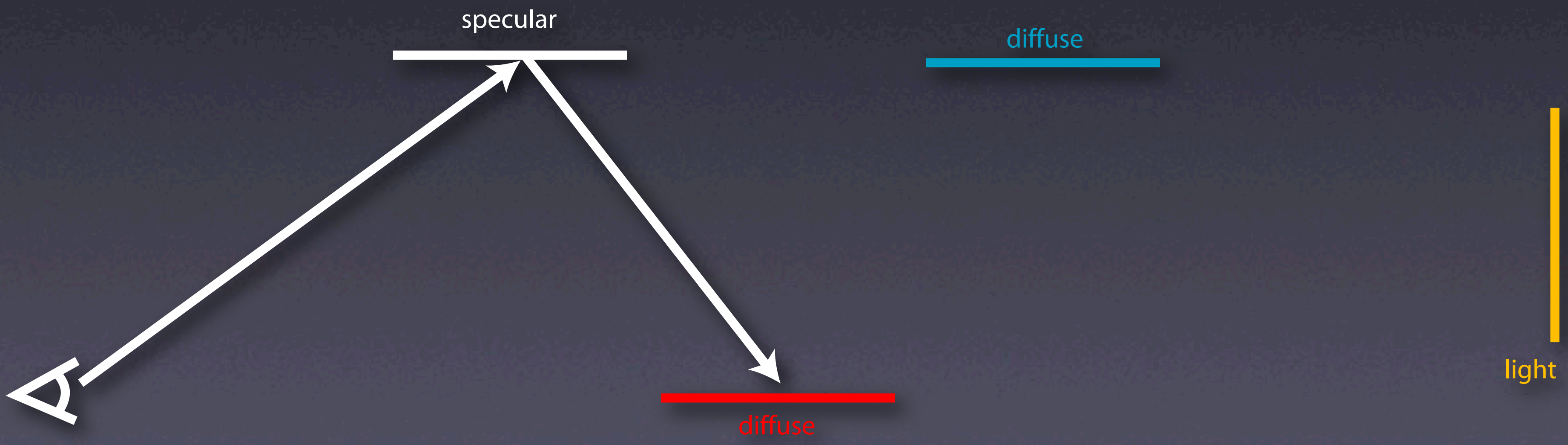
$$\frac{T(y \to x)}{T(x \to y)} = 1$$



Specular › (Specular | Diffuse)

Travel in the same incoming direction as path x at specular surface

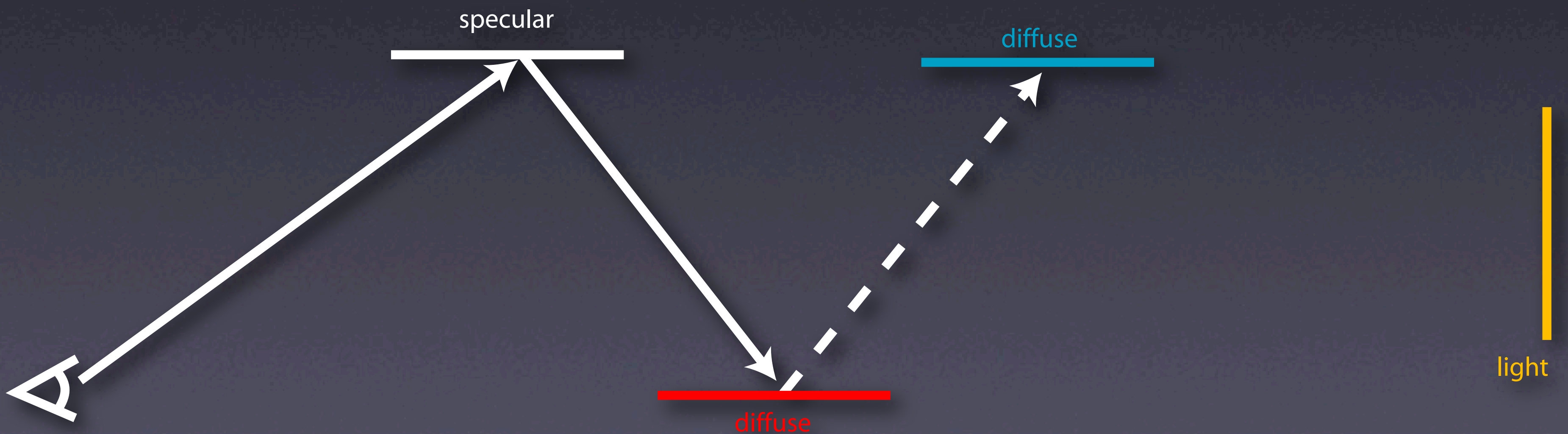
$$\frac{T(y \to x)}{T(x \to y)} = 1$$



Diffuse › Diffuse

If two diffuse surfaces, just connect the paths

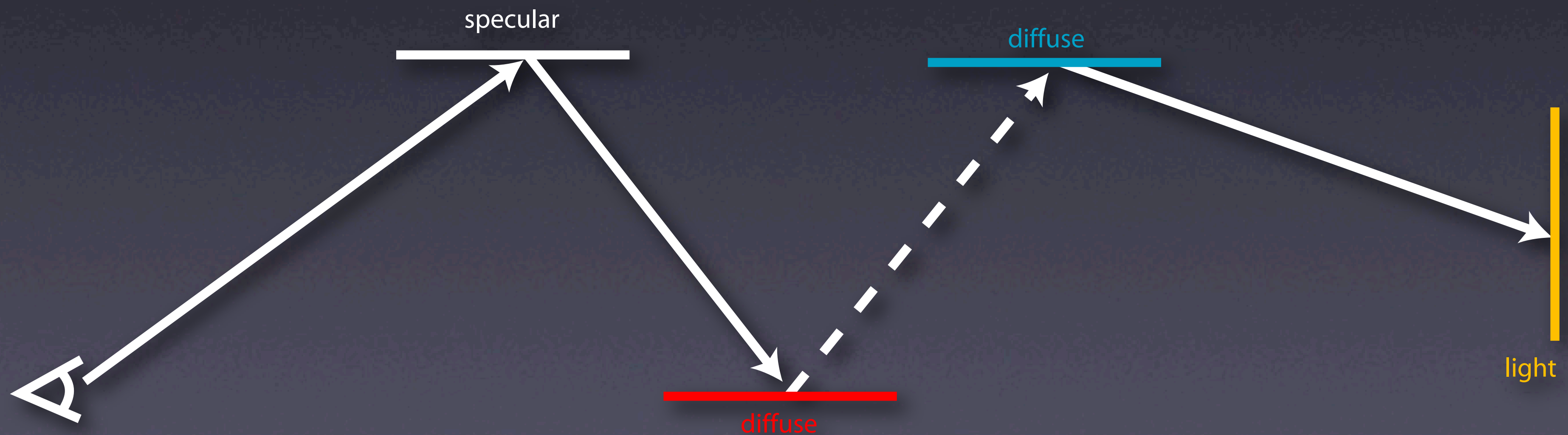
$$\frac{T(y \rightarrow x)}{T(x \rightarrow y)} = \frac{\cos\theta_{1_x}\cos\theta_{2_x}}{d_x^2}\frac{d_y^2}{\cos\theta_{1_y}\cos\theta_{2_y}}$$



Remaining Connections

Copy remaining path, possibly perturb if diffuse vertex on path

$$\frac{T(y \to x)}{T(x \to y)} = 1$$



Putting it All Together

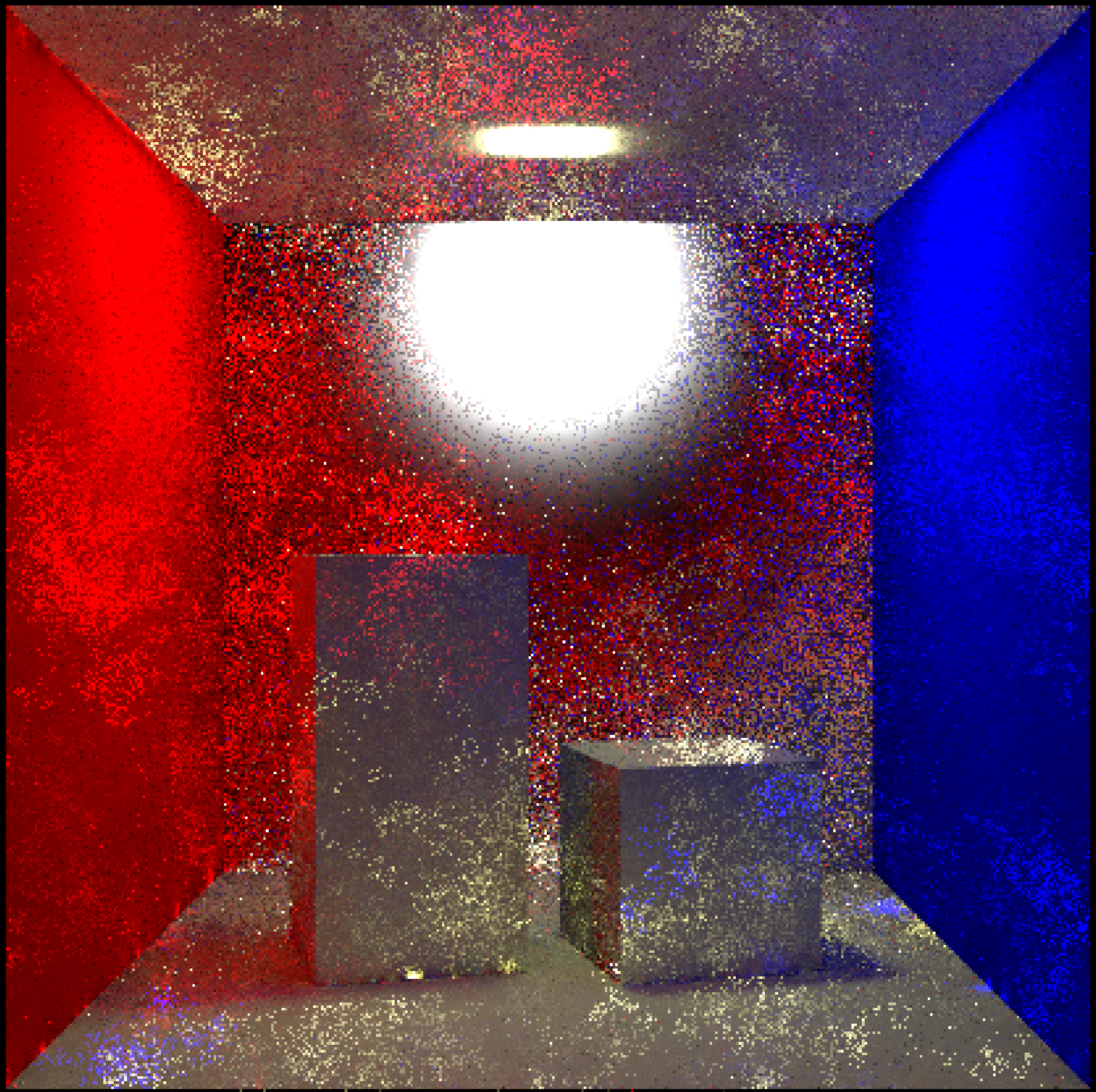
```
start with random path x

for i = 0 to mutations
    y = mutate(x)
    acceptance = y/x T(y › x)/T(x › y)
    if (randomReal(0,1) < acceptance)
        x = y

    deposit path x at corresponding pixel
end for
```

- we just went over transfer probabilities for paths
- y and x are represented by luminance value of sample when calculating acceptance

Up Scaling

- Undetermined amount of samples per pixel
- Scale up by average number of samples per pixel

Results



200 MC samples 500 mutations

Results

- Difficult to compare in sheer terms of samples
  - Mutations are much faster than MC
    - Path reuse
    - Cache coherence
- Possible specular bug
- Splotchiness?

Problems with MLT

- Start-up Bias
  - If we just mutate one randomly generated path, in the limit, we will eventually get a correct solution if we apply several mutation strategies
  - However, we do not want to be close to the limit, we want to be done with a few samples
  - Thereby, we generate several random paths and mutate them

To Do...

- Transfer probabilities and mutations are the key
  - Several transfer probabilities I did not even describe in this presentation
- Find a better estimation for changes in path density, you could find a better solution
  - Possibly publish another SIGGRAPH paper

Established Improvements

- Importance Sample Mutations
- Energy Redistribution
  - Improved Balance Algorithm
  - Post-process filtering

References

- Pharr, M., & Humphreys, G. PBRT
- Cline, D., & Egbert, P., A Practical Introduction to Metropolis Light Transport. Tech Report, Brigham Young University
- Cline, D., Talbot, J., & Egbert, P., Energy Redistribution Path Tracing, SIGGRAPH '05
- Veach, E., Robust Monte Carlo Methods for Light Transport Simulation. Dissertation, Stanford '97