Gradient-Domain Metropolis Light Transport

Jaakko Lehtinen
NVIDIA
Aalto University

Tero Karras
NVIDIA

Samuli Laine
NVIDIA

Miika Aittala
NVIDIA
Aalto University

Frédo Durand
MIT CSAIL

Timo Aila
NVIDIA

NVIDIA

Aalto University
School of Science

CSAIL

Image Processing 101

I(x,y)

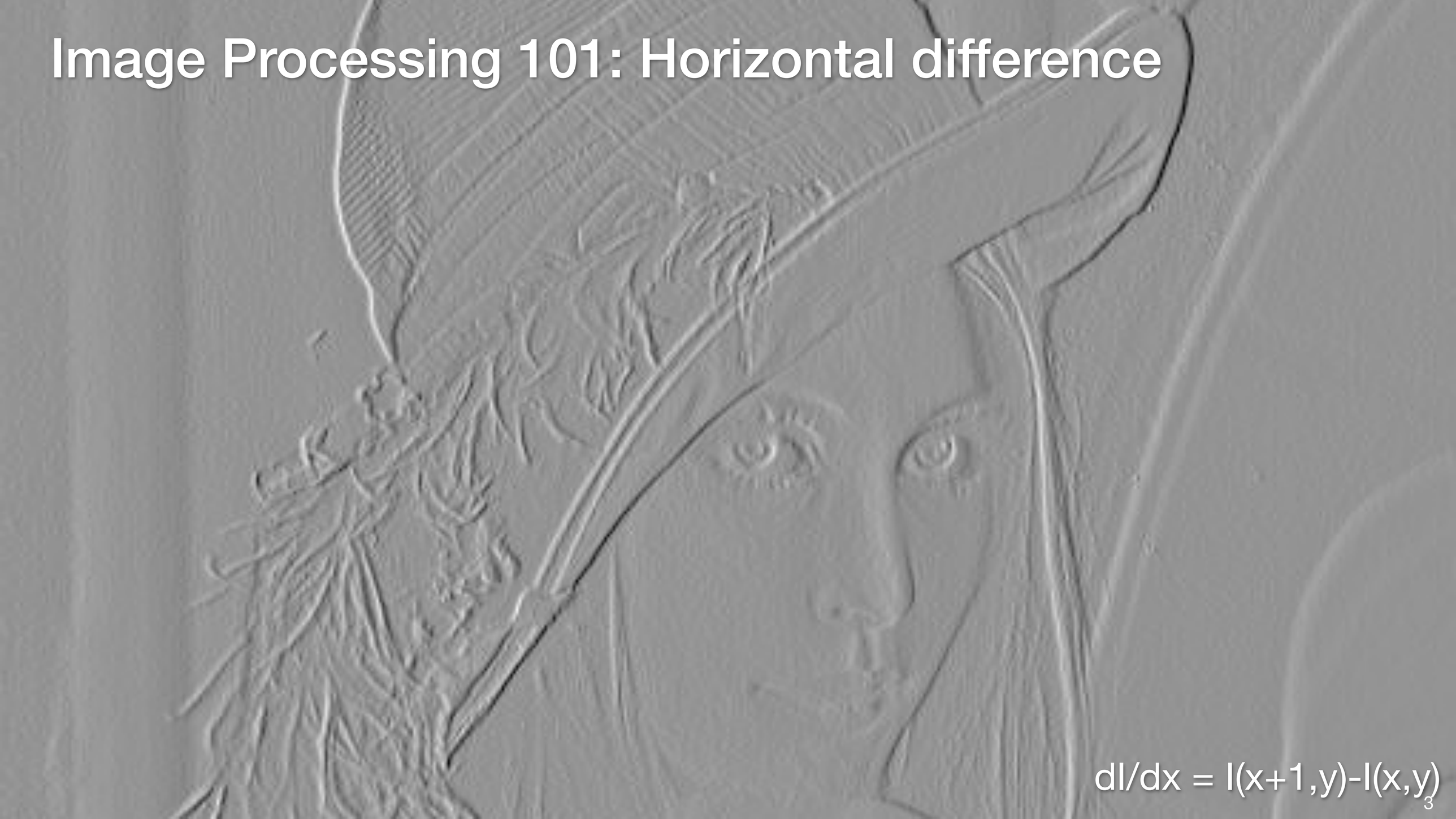
Image Processing 101

I(x,y)

Image Processing 101

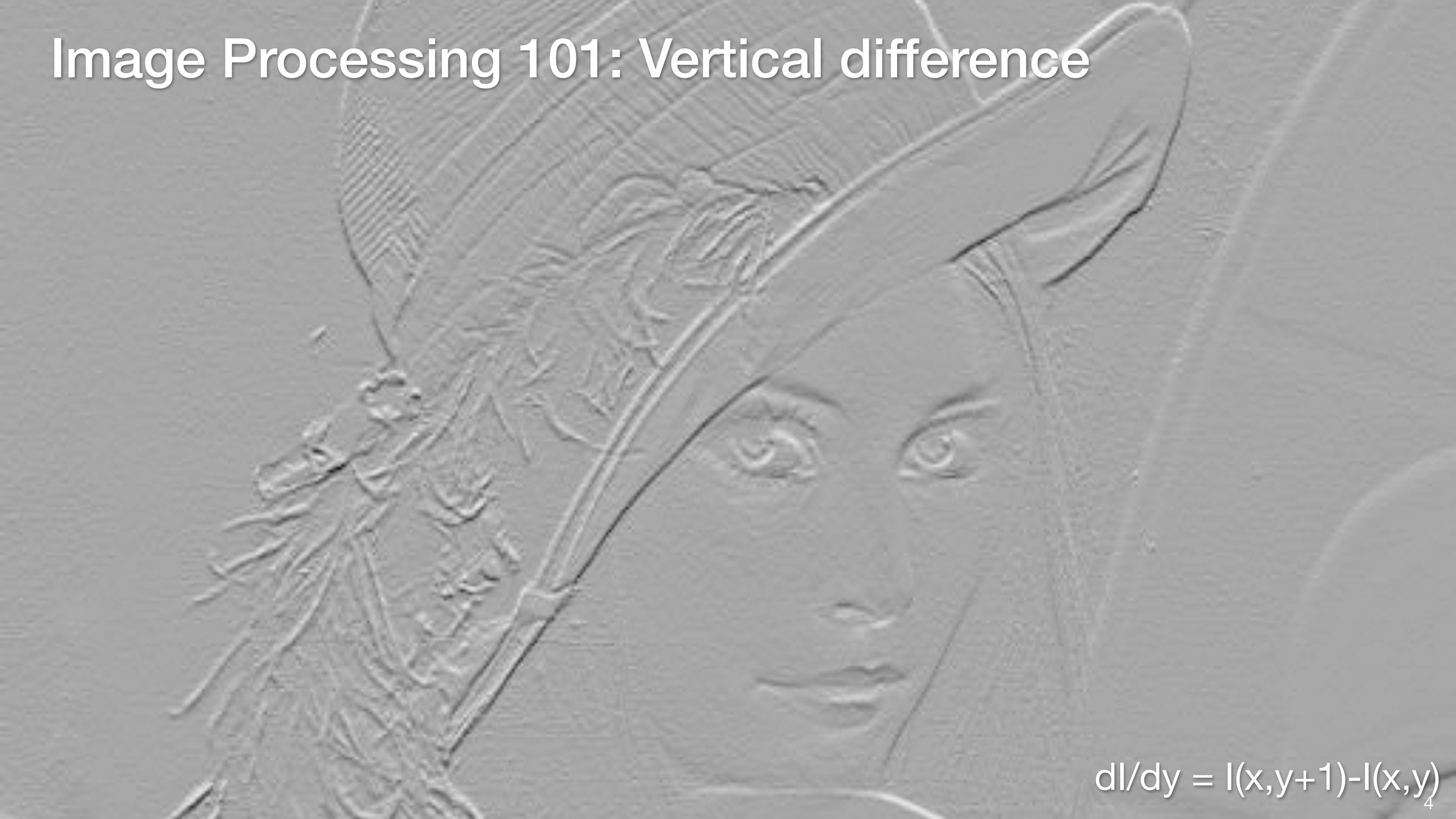
I(x,y)

Image Processing 101: Horizontal difference



$dI/dx = I(x+1,y)-I(x,y)$

Image Processing 101: Vertical difference



$dI/dy = I(x,y+1)-I(x,y)$

Classic Observation: Derivatives are Sparse

Classic Observation: Derivatives are Sparse

Derivatives fire **where the action is**

Yet they contain the **same information as the actual image**

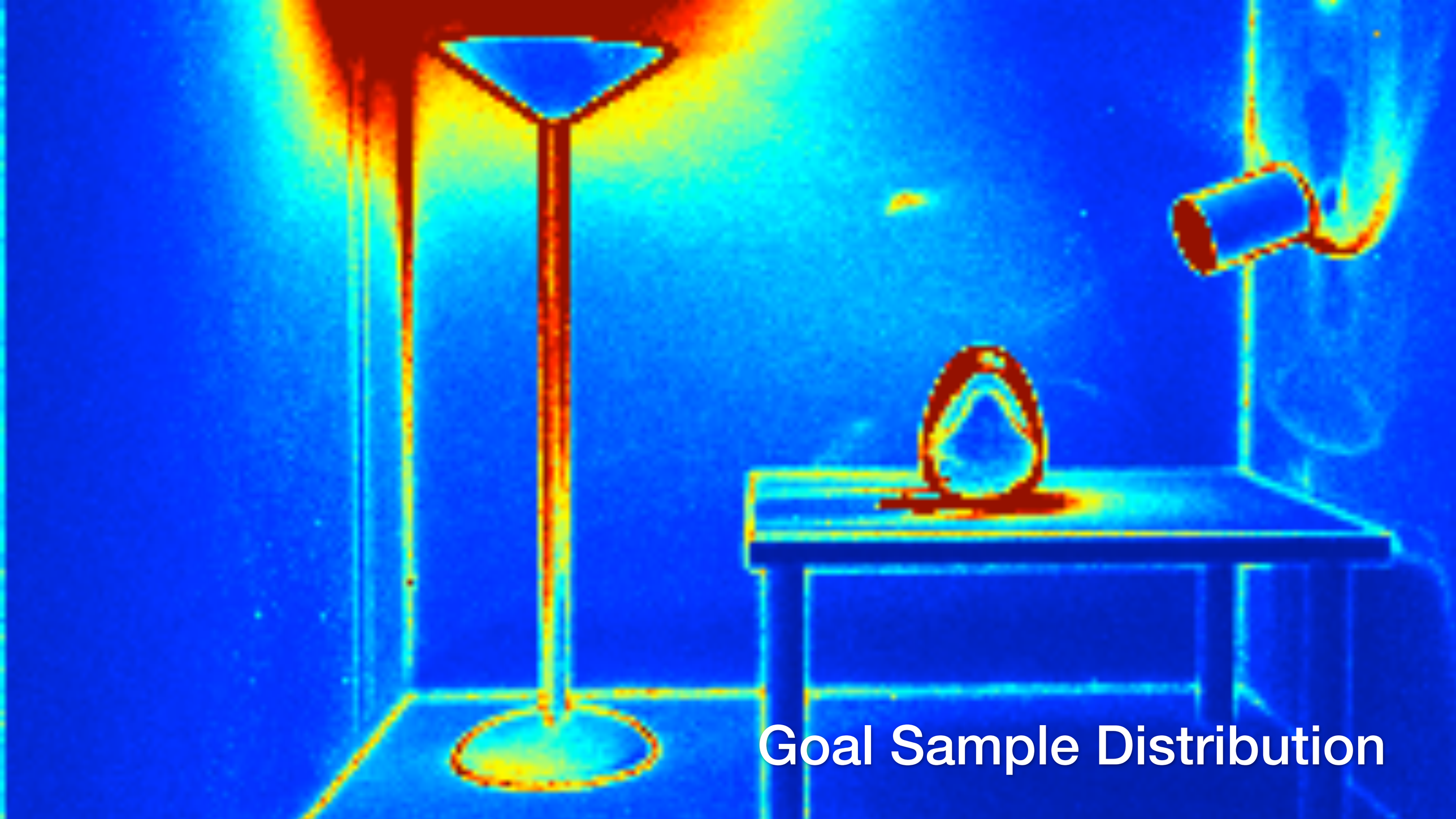
Can we render derivatives directly?

If so**, can we compute less in smooth areas?**

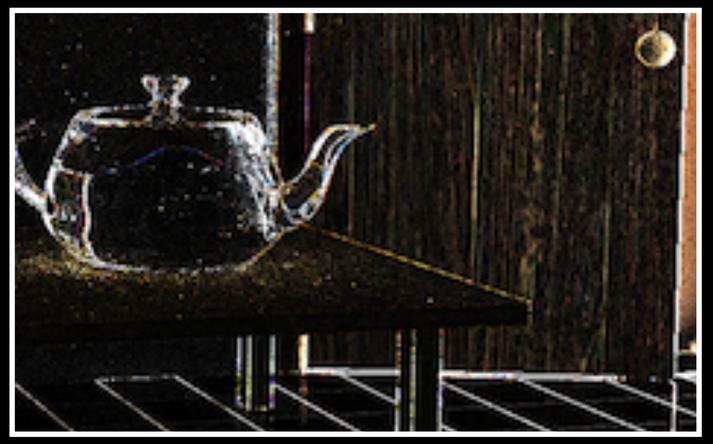
Pia Raben

Pia Raben


Indirect Illumination


Goal Sample Distribution

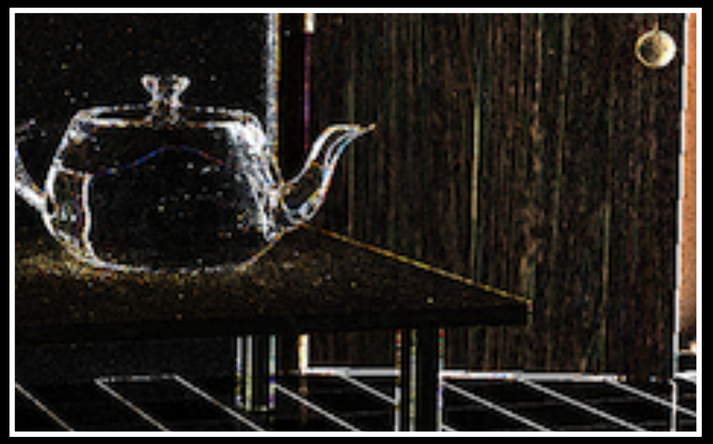
Desired Algorithm Overview



Horizontal Difference

Vertical Difference

Desired Algorithm Overview

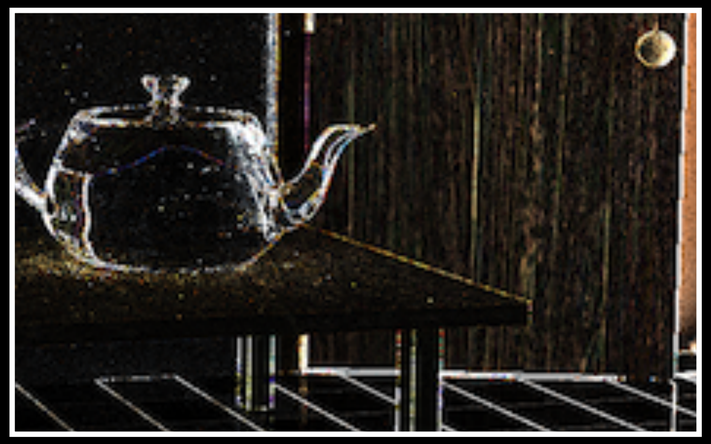


Horizontal Difference

Vertical Difference

Poisson Solver

Desired Algorithm Overview



Horizontal Difference

Vertical Difference

Poisson Solver

Final Image

Desired Algorithm Overview

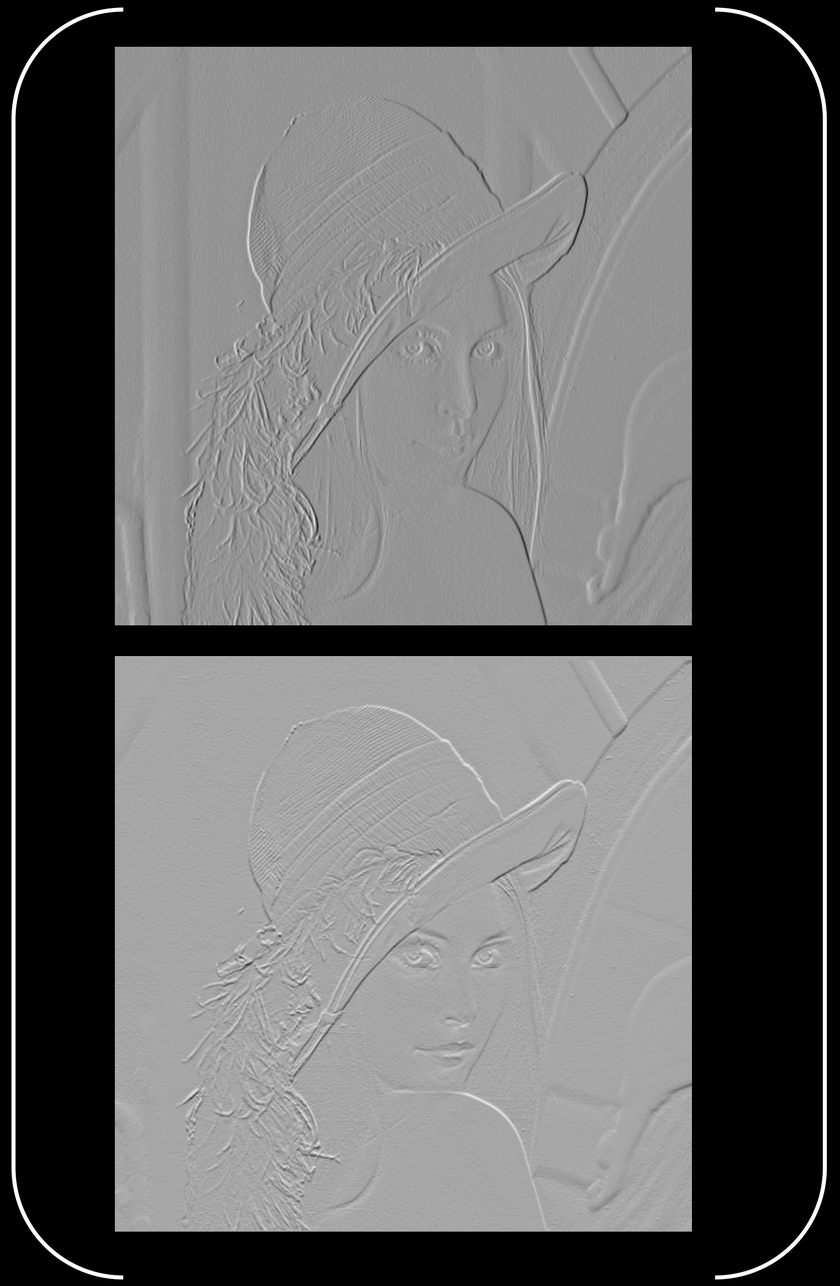
Horizontal Difference

Vertical Difference

Poisson Solver

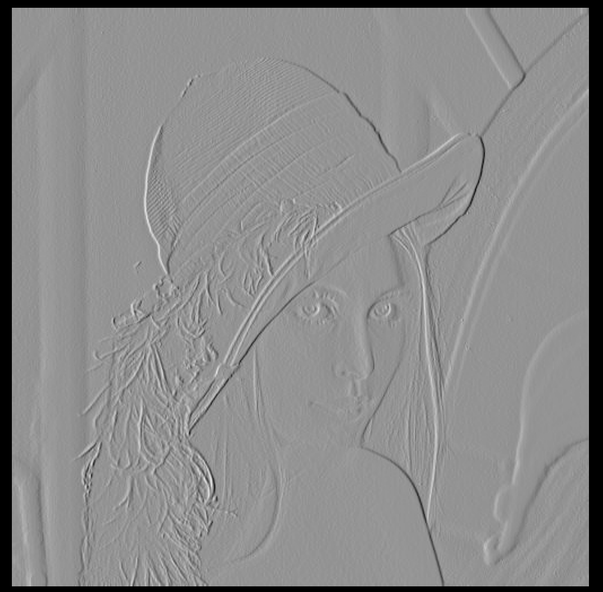
Final Image

“Fundamental Theorem of Calculus”

∇  $=$ 

“Fundamental Theorem of Calculus”

$=$ $\int$



Path Sampling Light Transport

Most traditional rendering algorithms fire rays through each pixel

— path tracing, etc.

— determine intensity by averaging many samples (MC sampling)

— sample all possible paths light can take from source to sensor

Not useful for finding derivatives

Metropolis Light Transport (MLT) [Veach97]

Metropolis Light Transport (MLT) [Veach97]

Construct a Markov Chain of light paths

— start from a random path

— mutate with a carefully chosen probability

— repeat

Count number of paths that landed in each pixel

Metropolis Light Transport (MLT) [Veach97]

Tends to direct computation to paths where lots of light flows..

Metropolis Light Transport (MLT) [Veach97]

Tends to direct computation to paths where lots of light flows..

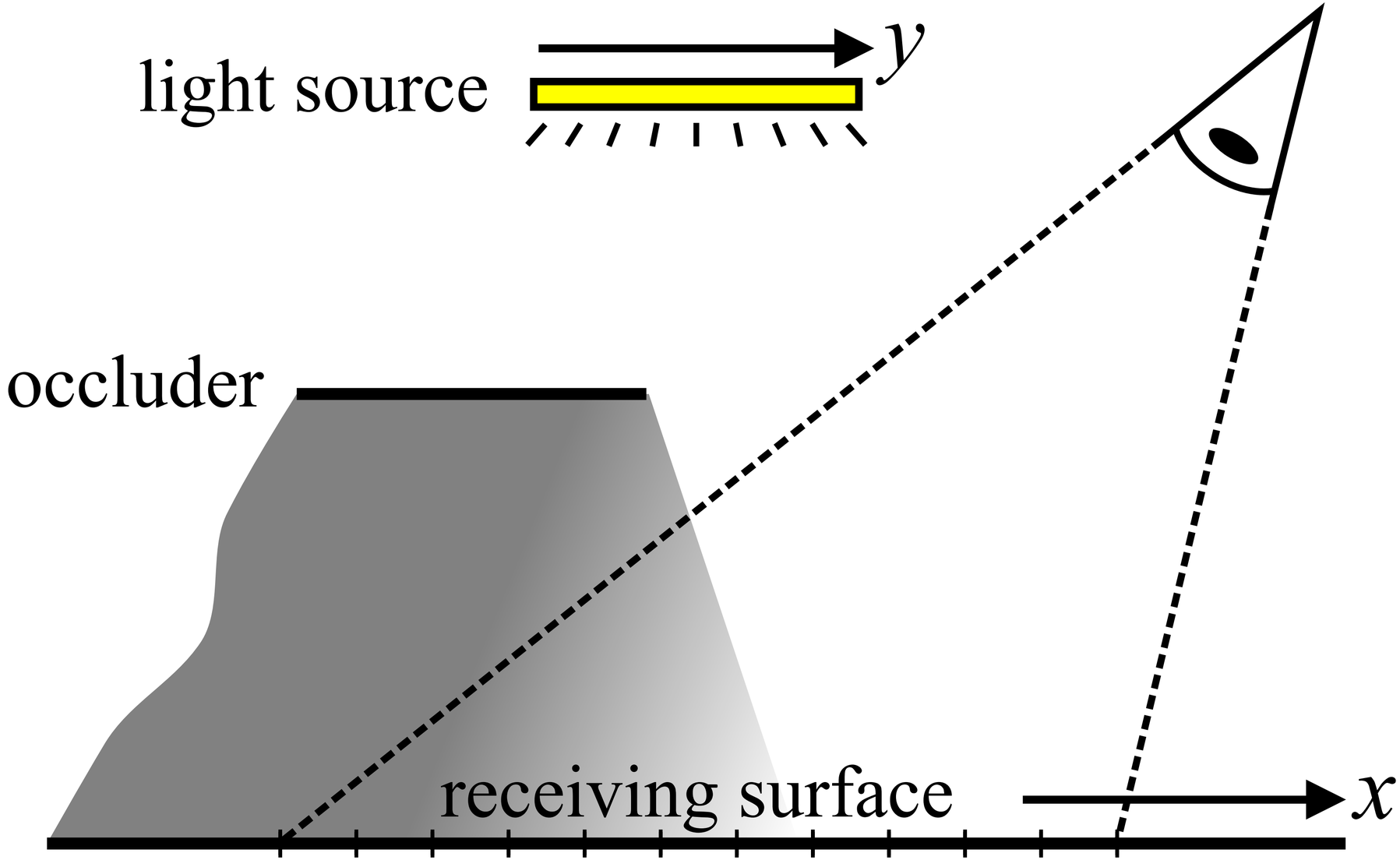
..without knowing where in advance!

Gradient-Domain MLT Intuition

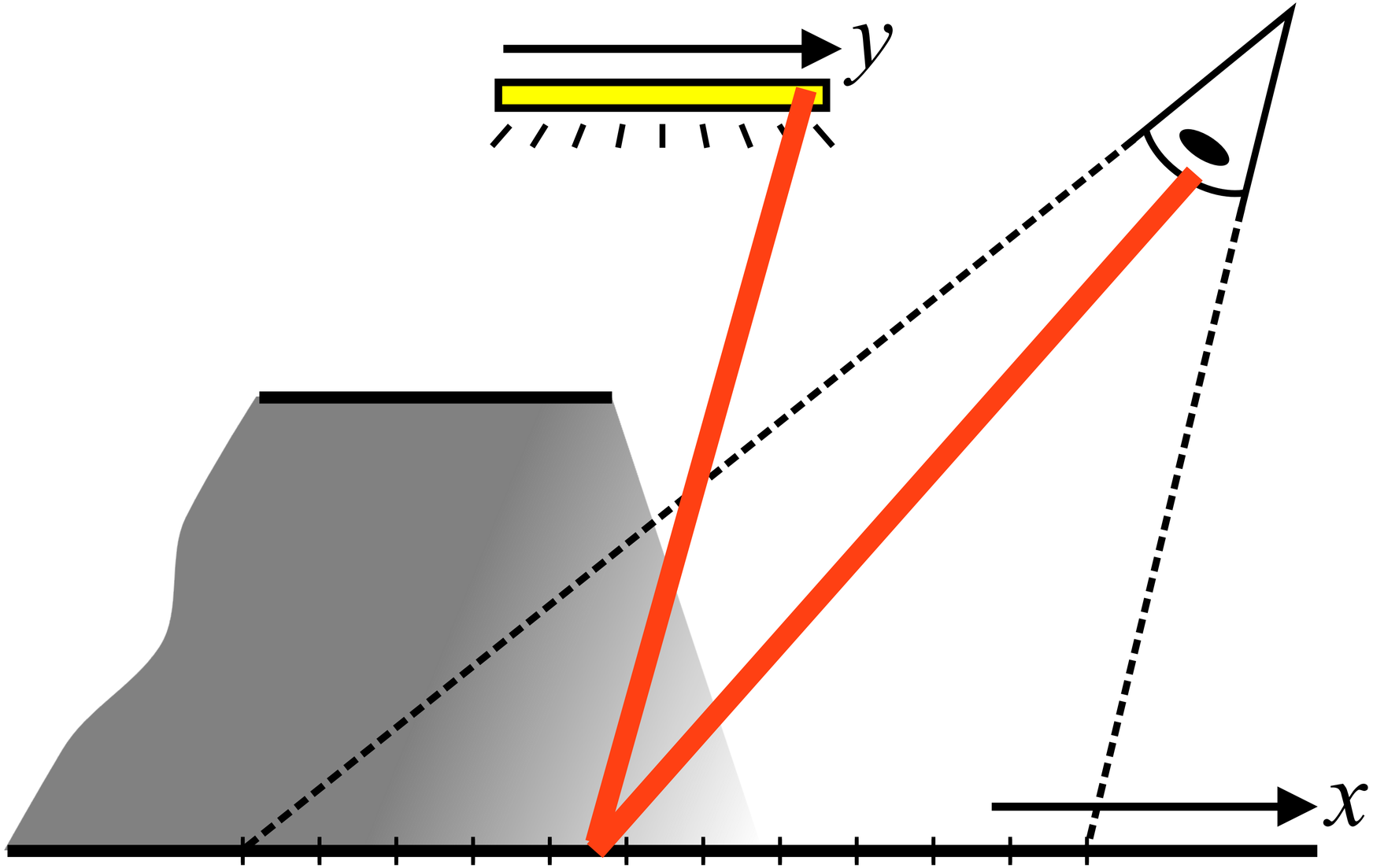
Gradient-Domain MLT Intuition

We concentrate sampling on paths where **light flow changes**.

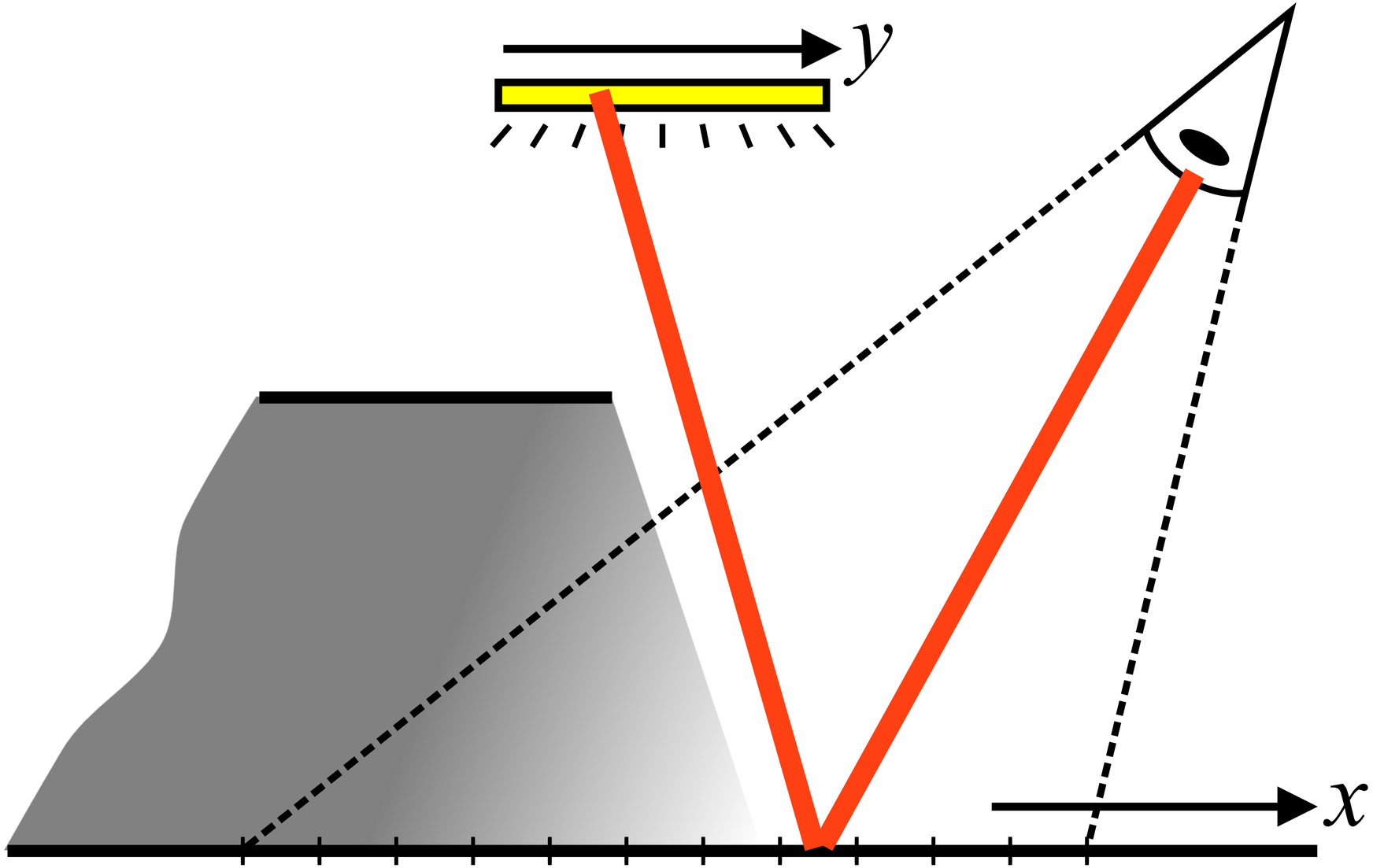
Simple 2D Example



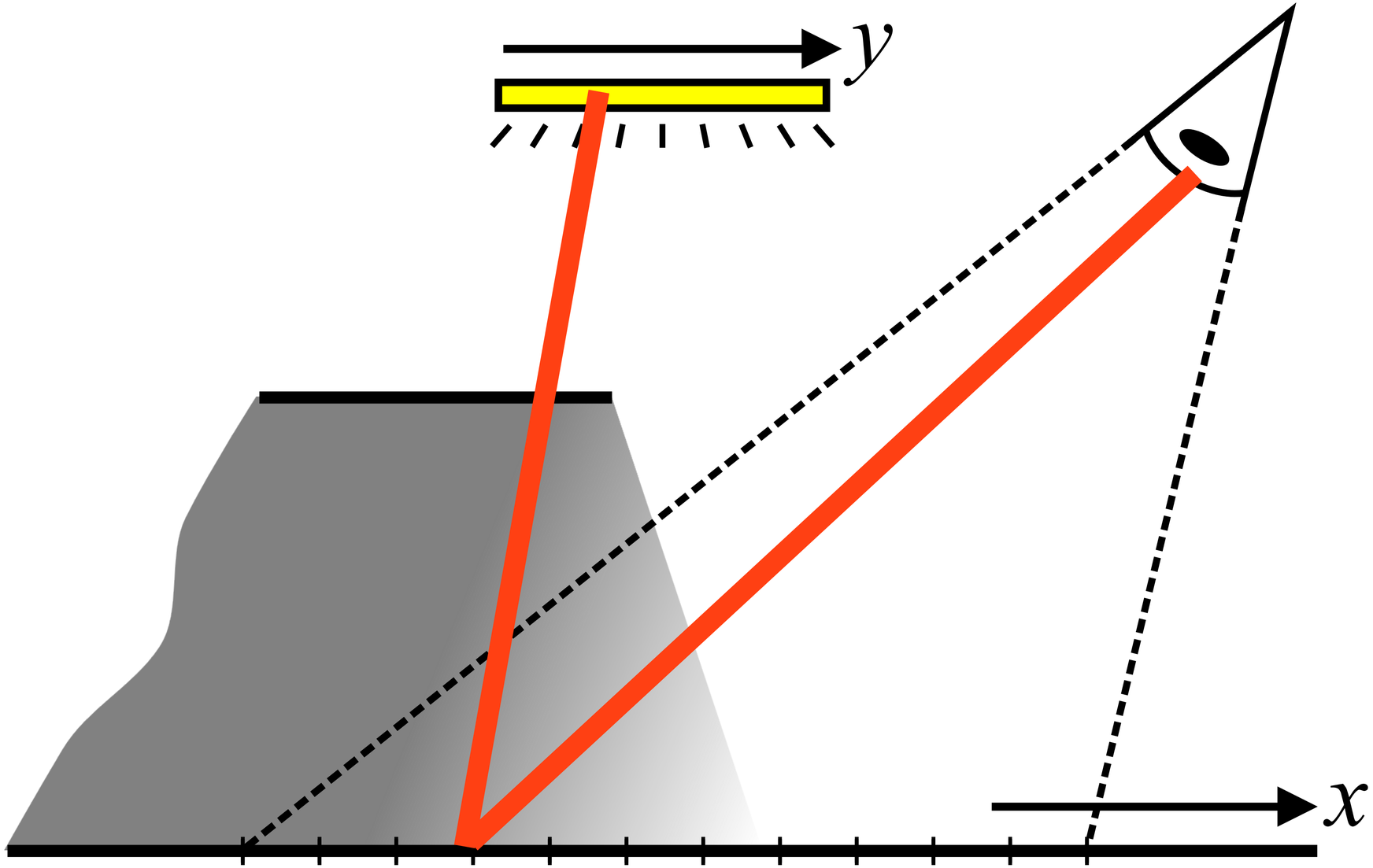
Simple 2D Example



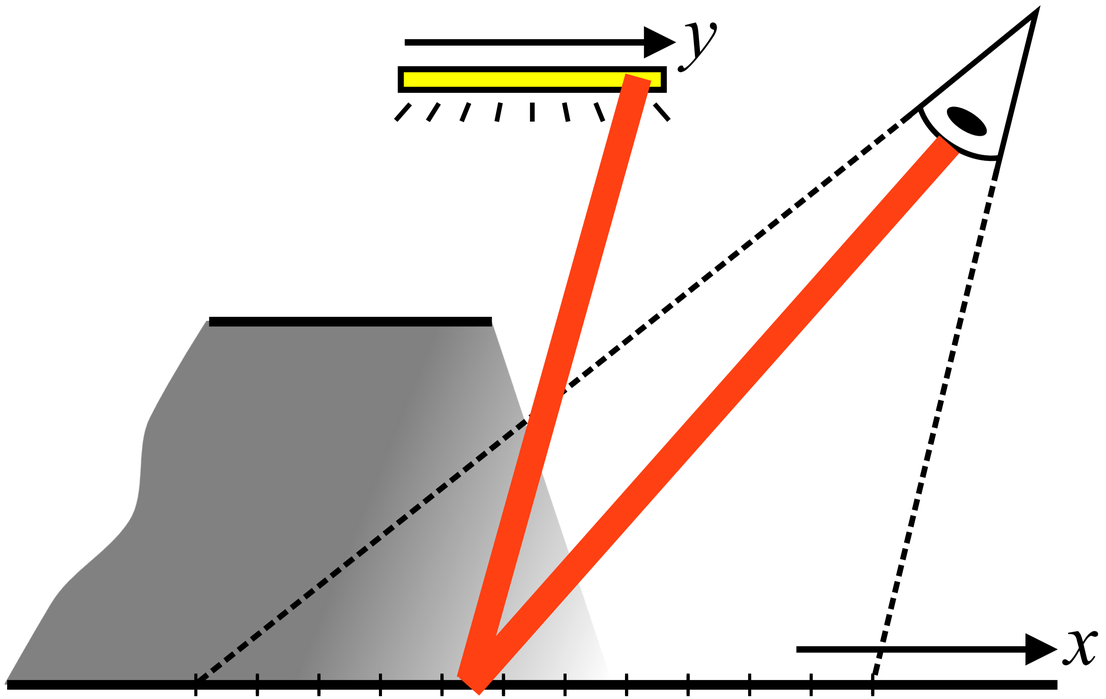
Simple 2D Example



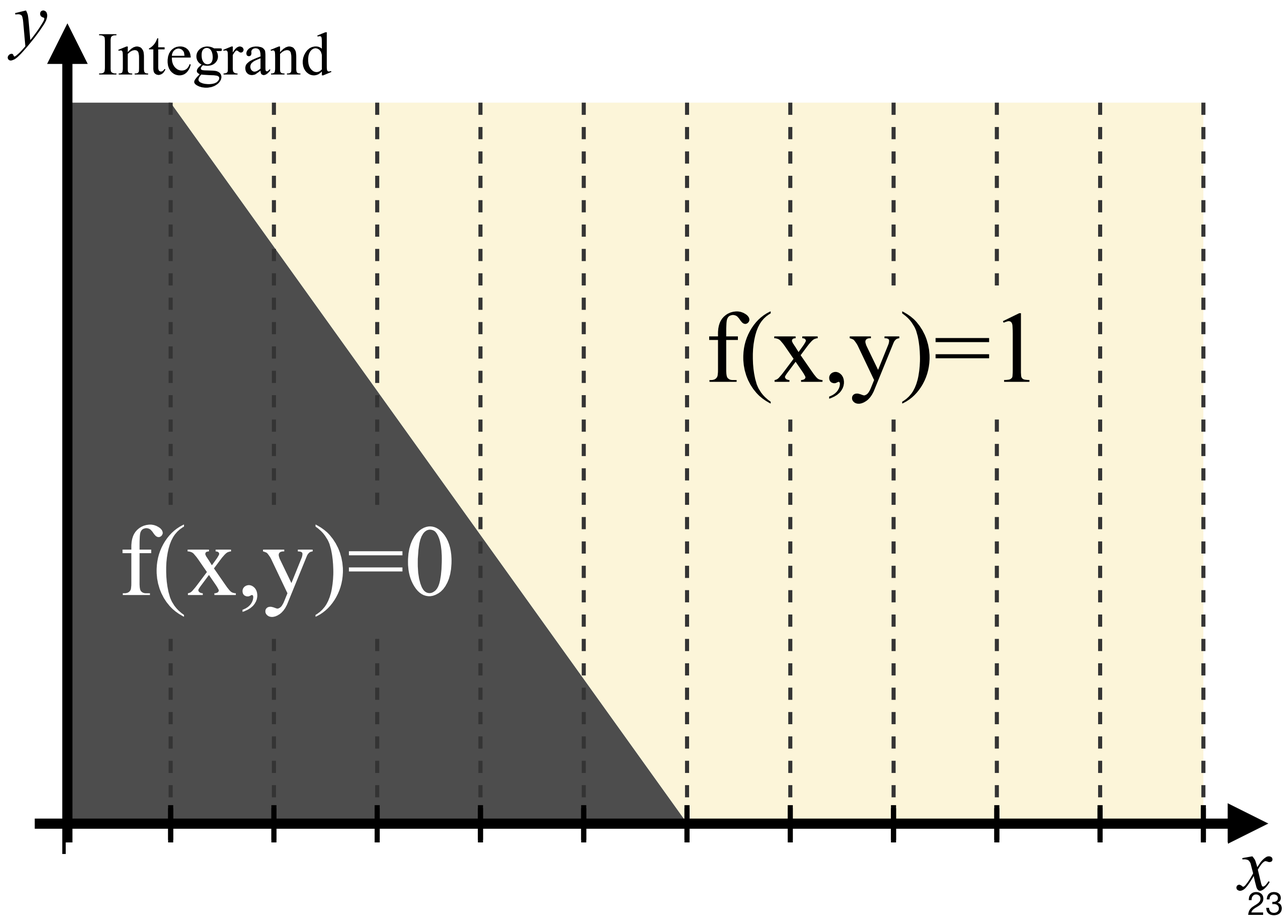
Simple 2D Example



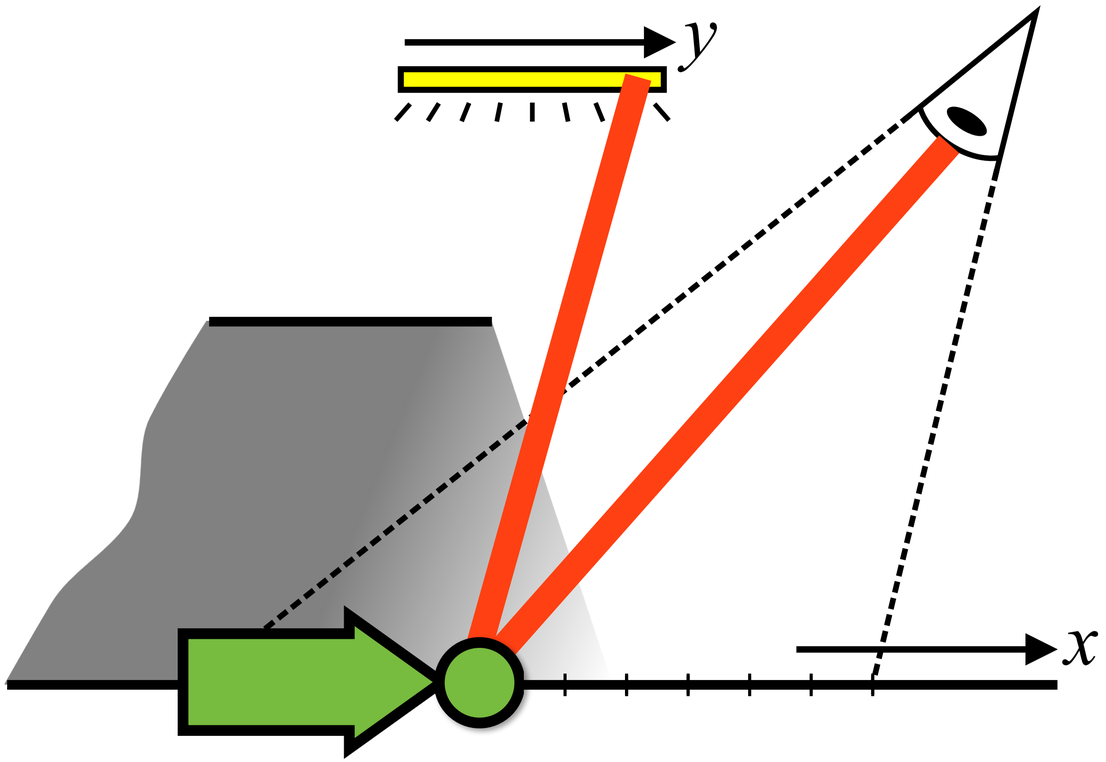
2D Path Space



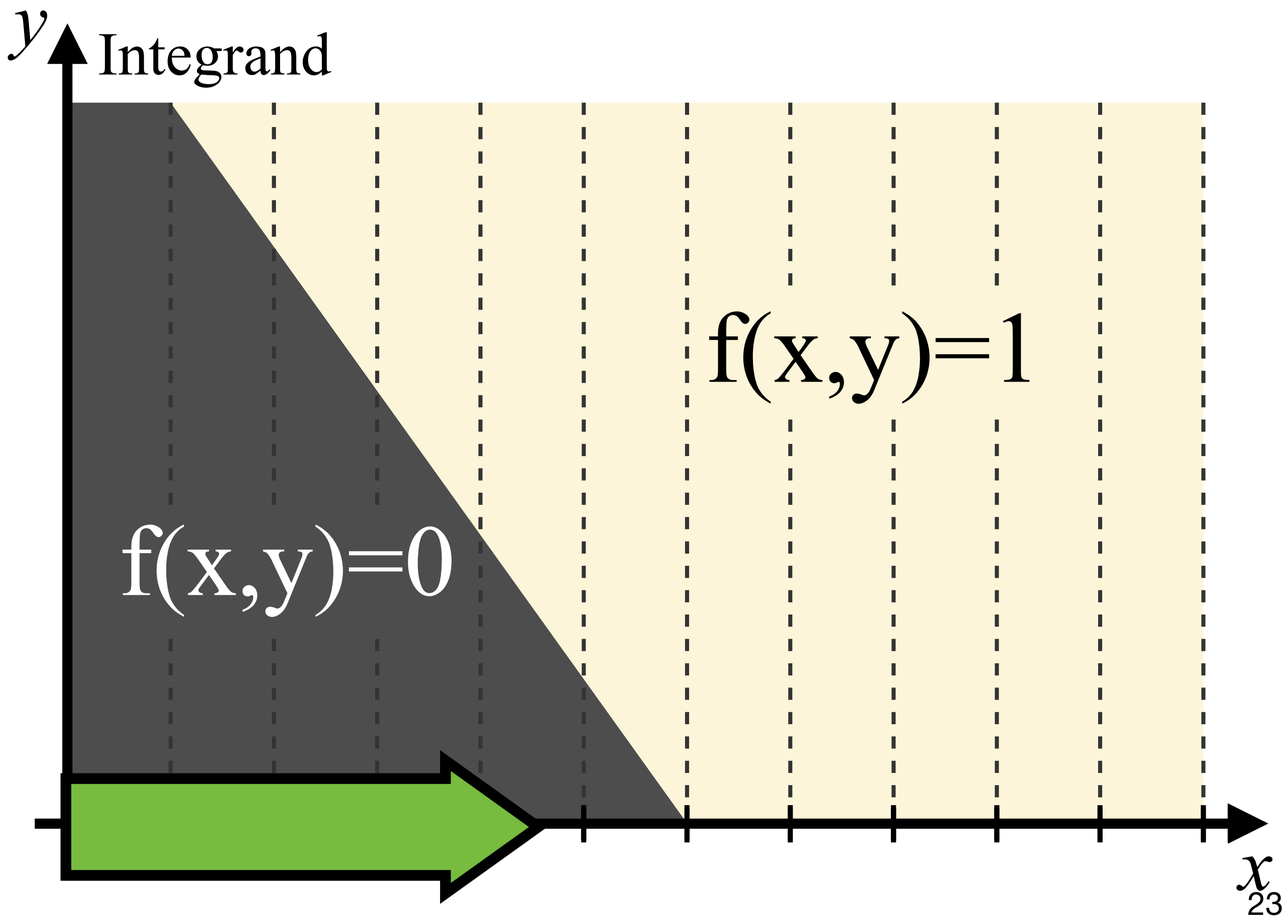
x = position on receiver,
y = position on light



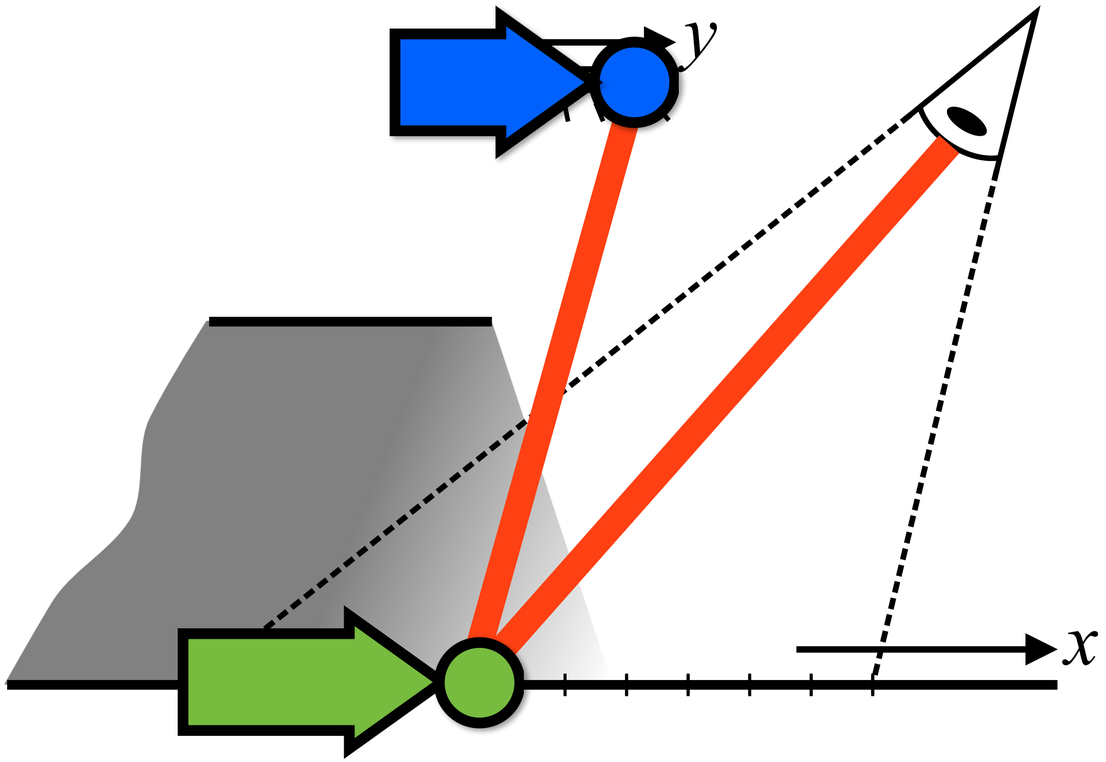
2D Path Space



x = position on receiver,
y = position on light

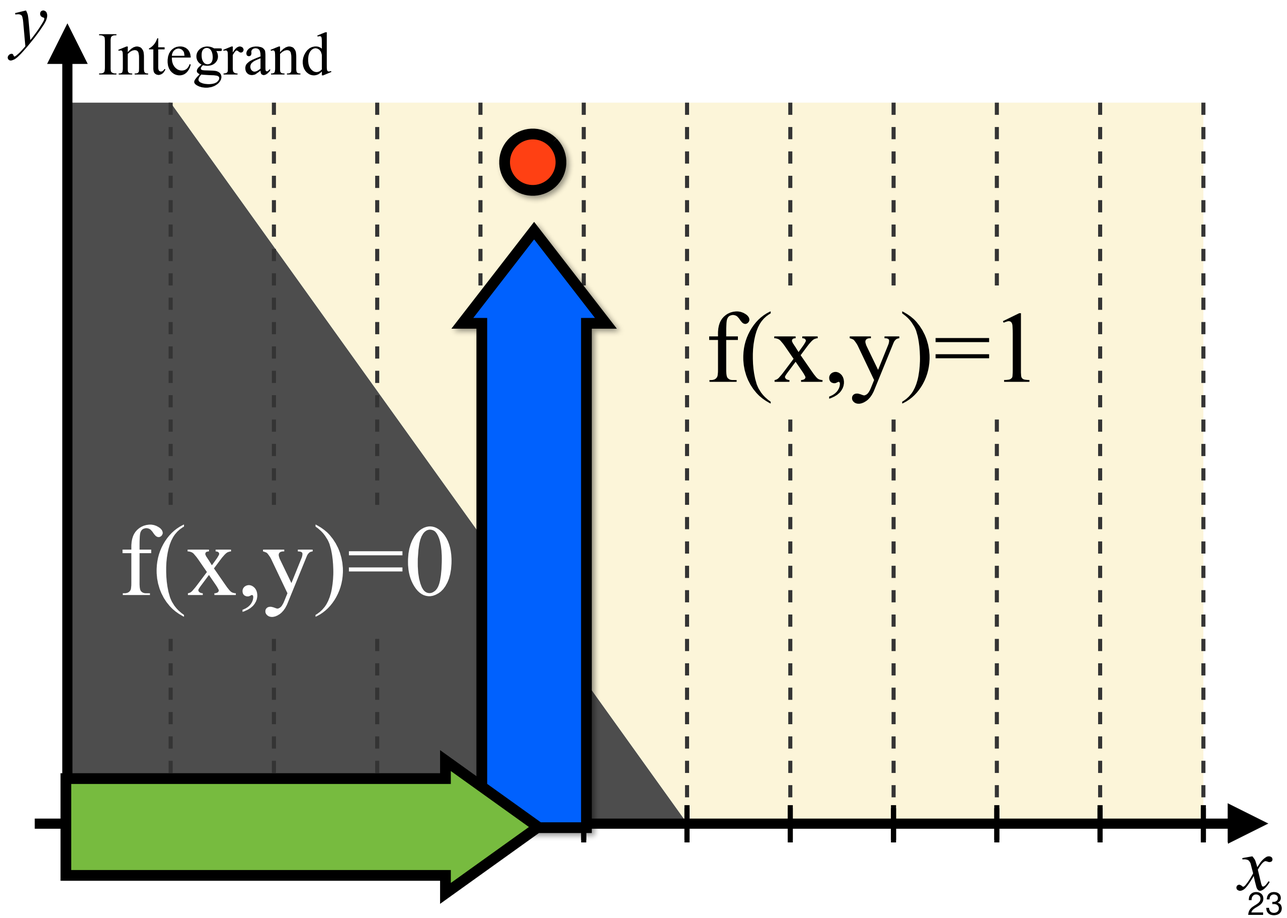


2D Path Space

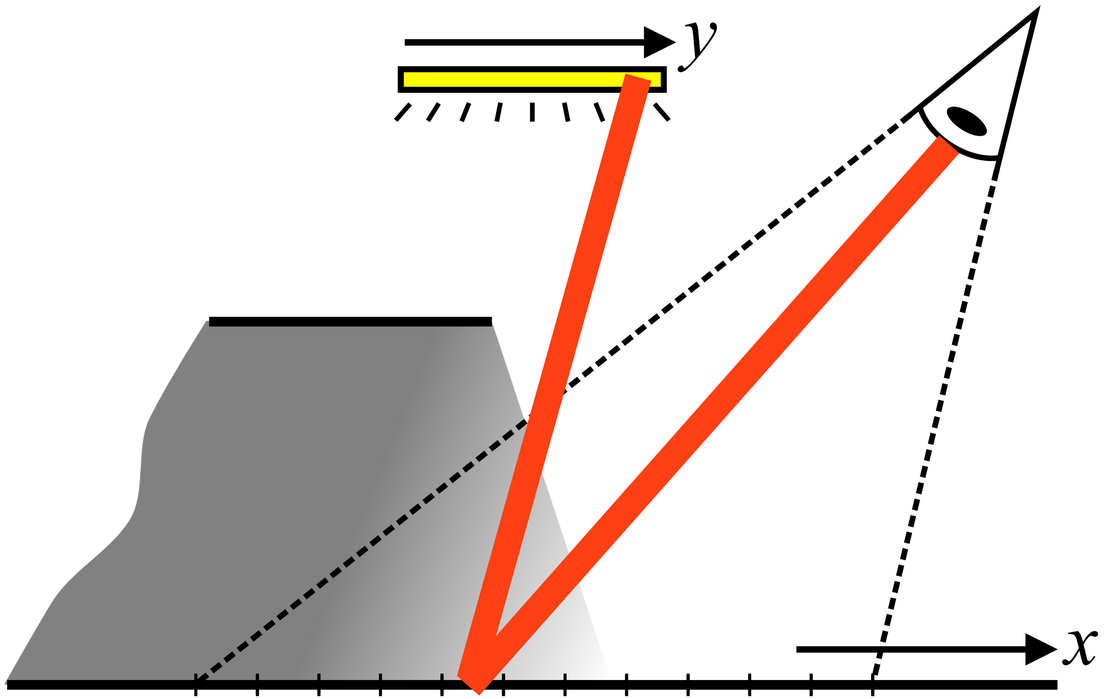


x = position on receiver,
y = position on light

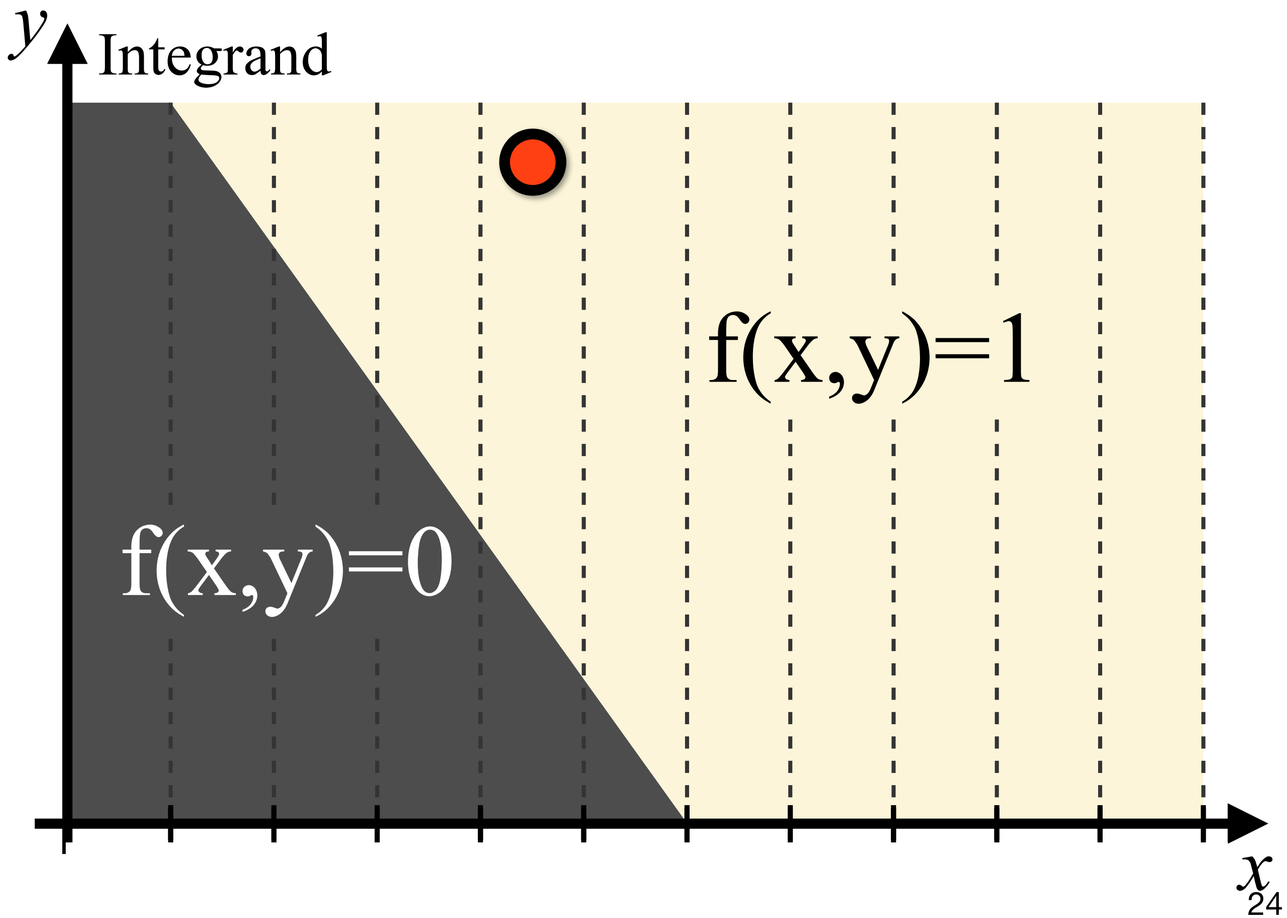
Each path is a point in path space



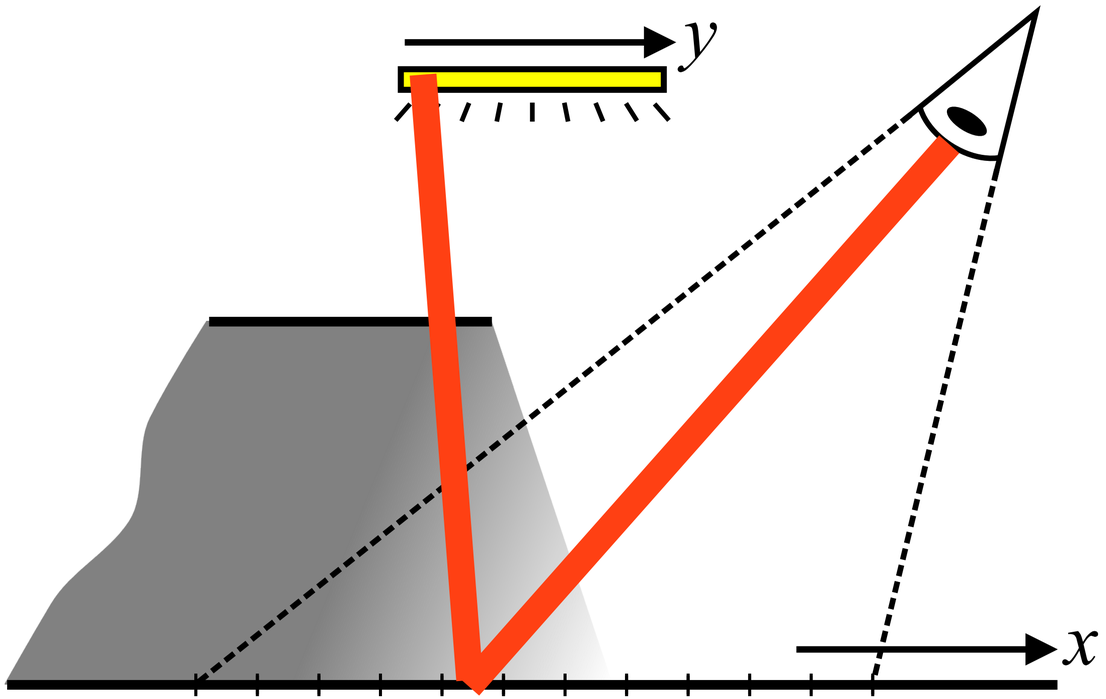
2D Path Space



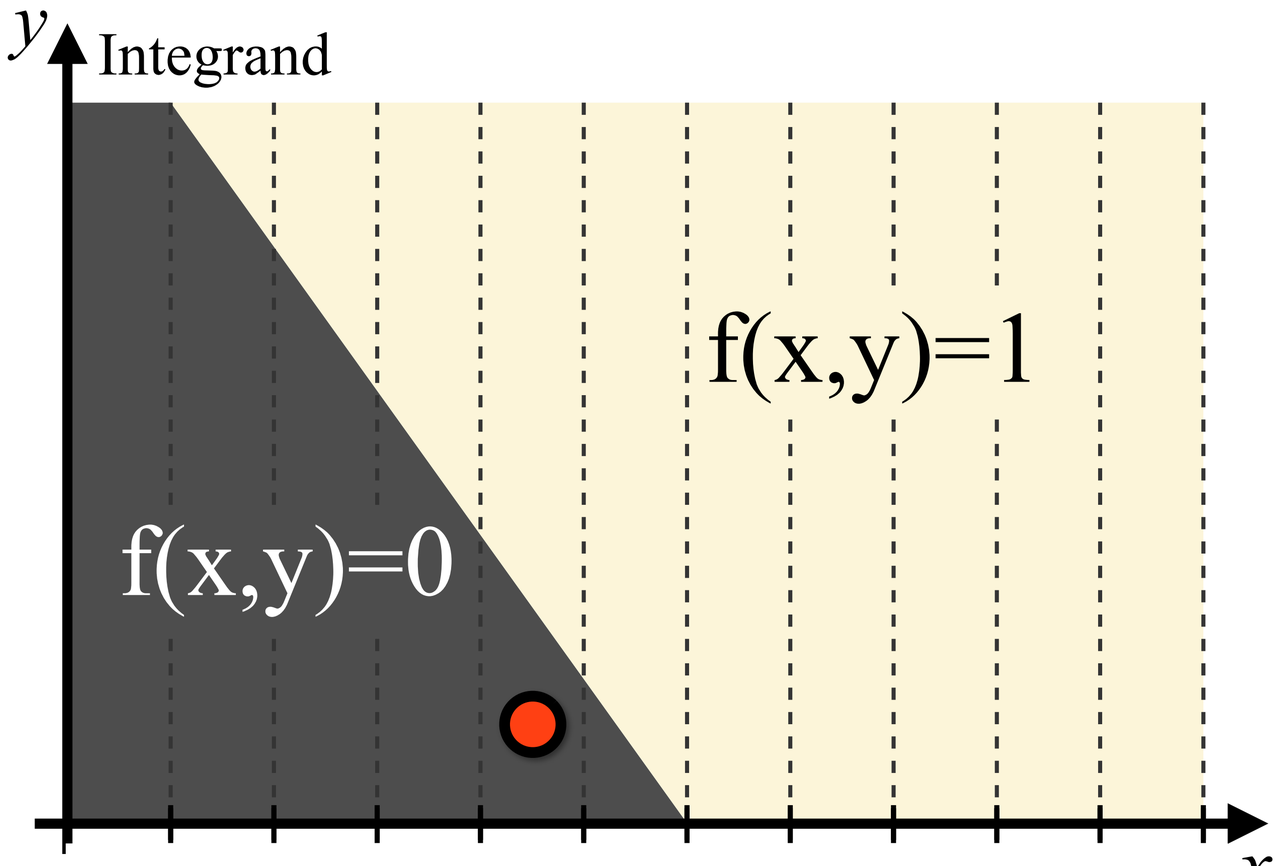
Path throughput function f
carries radiance, except
it is **zero for blocked paths**



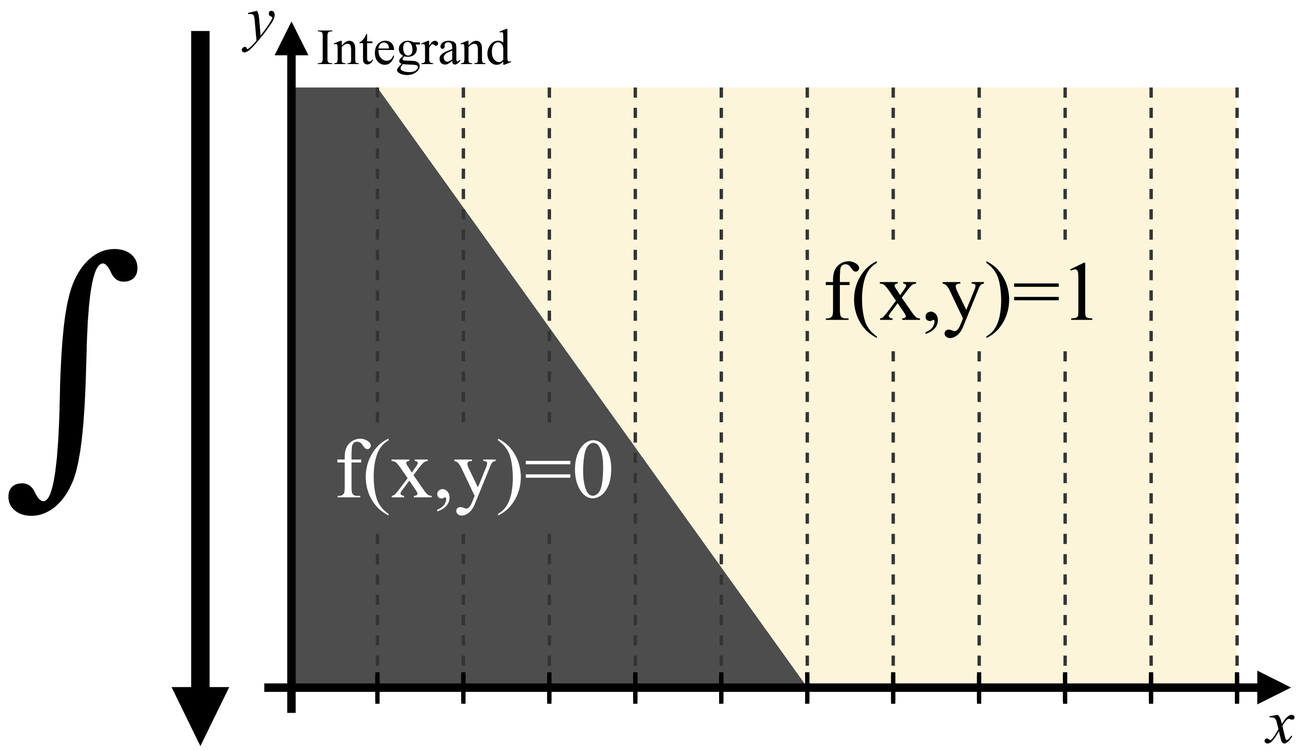
2D Path Space



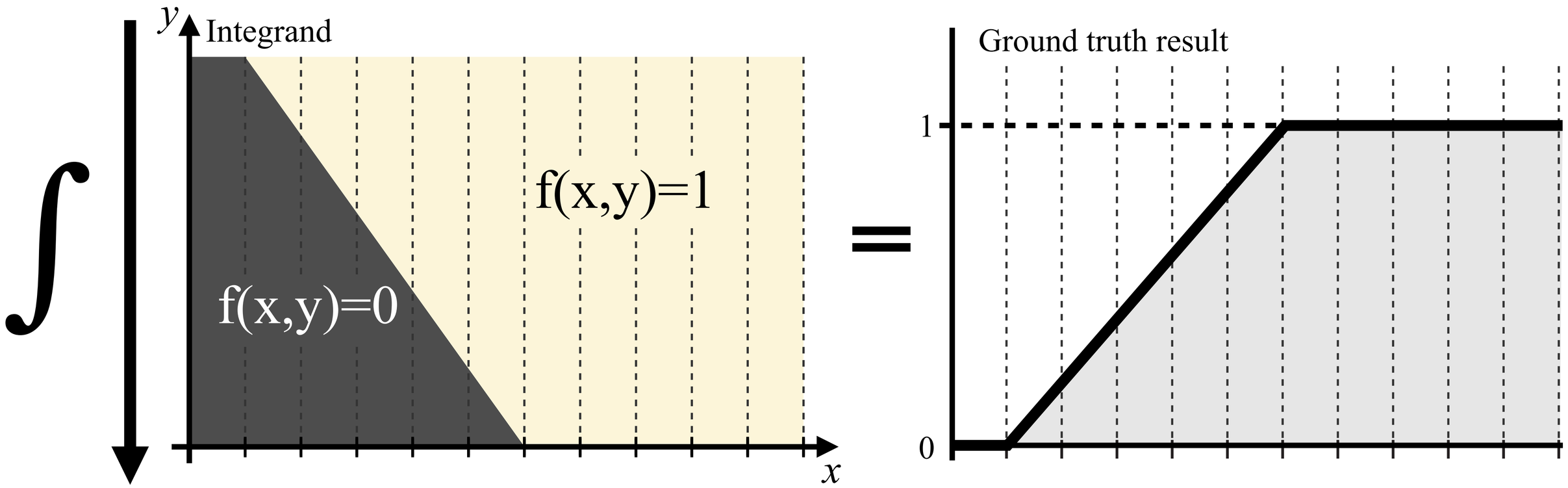
Path throughput function f
carries radiance, except
it is **zero for blocked paths**



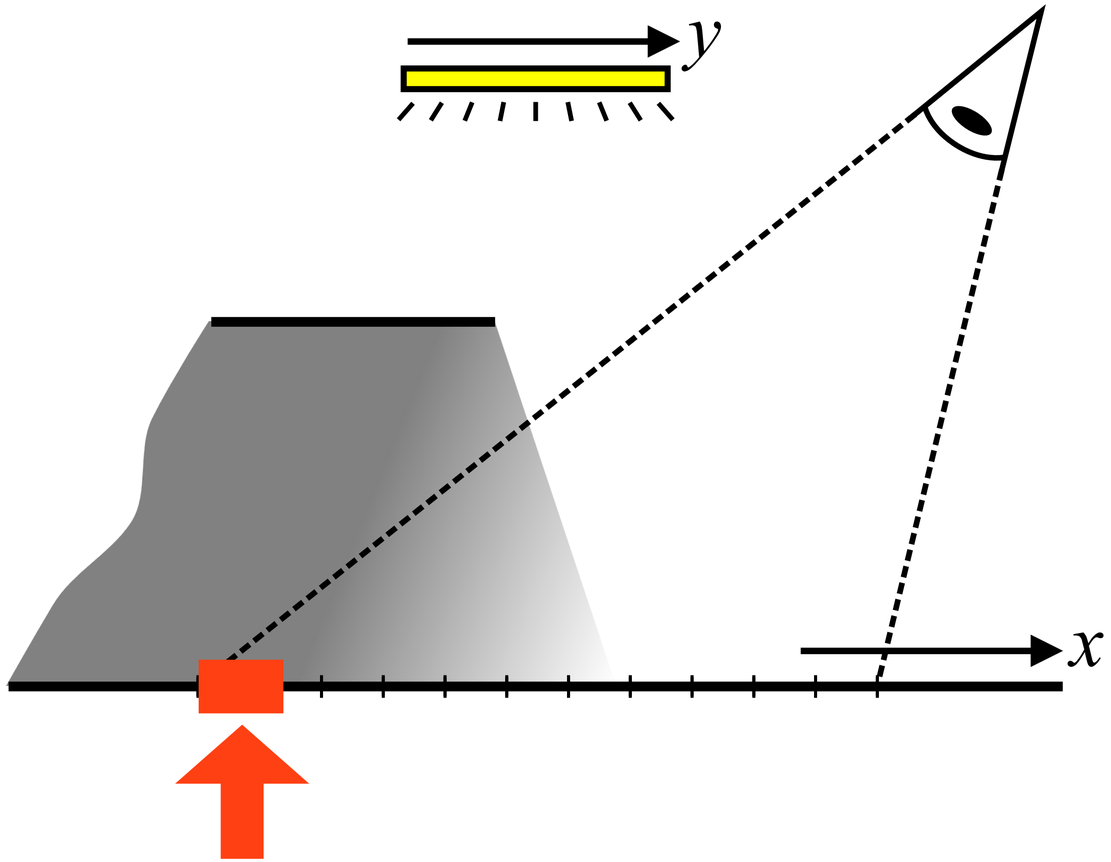
Shadowed Result is Integral Along Columns

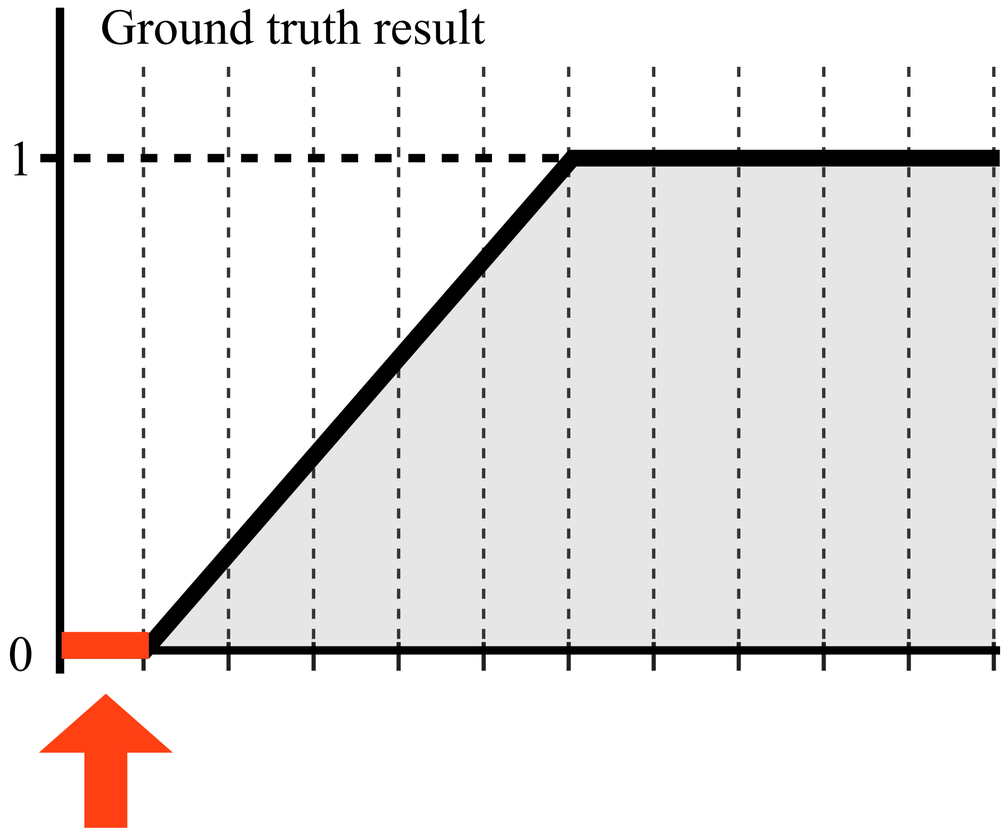


Shadowed Result is Integral Along Columns

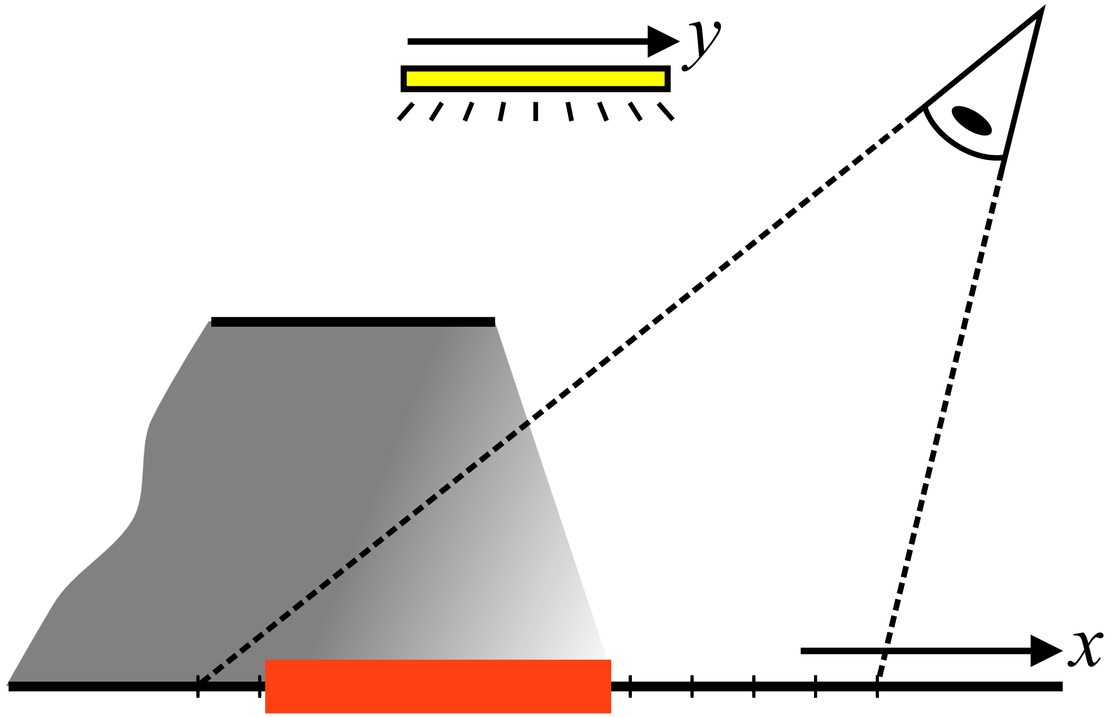


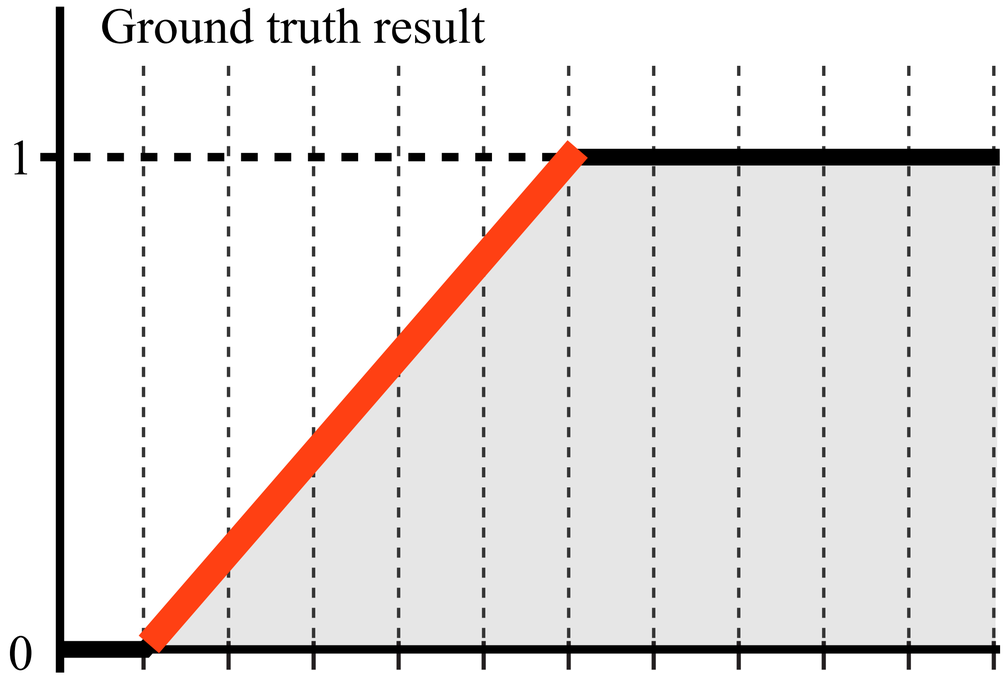
Full Shadow



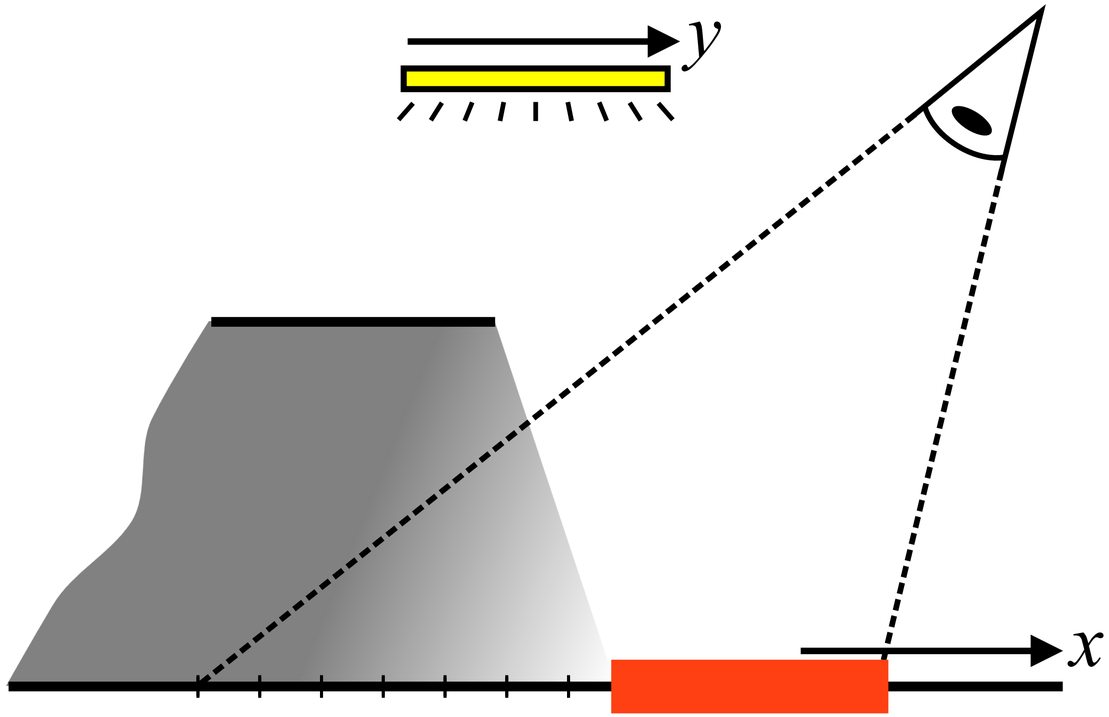


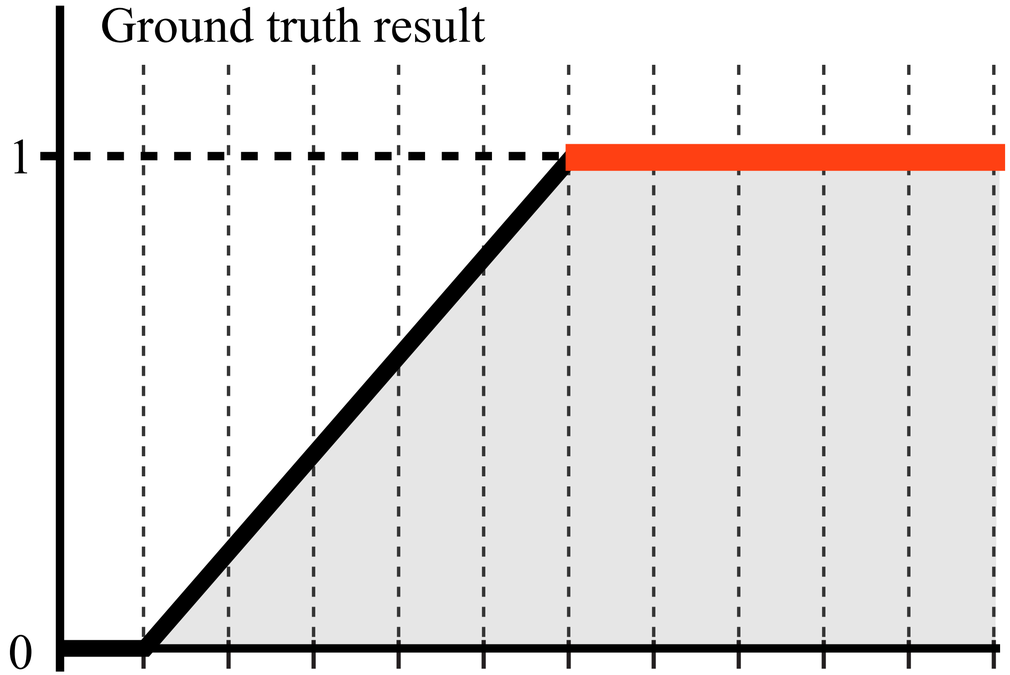
Penumbra



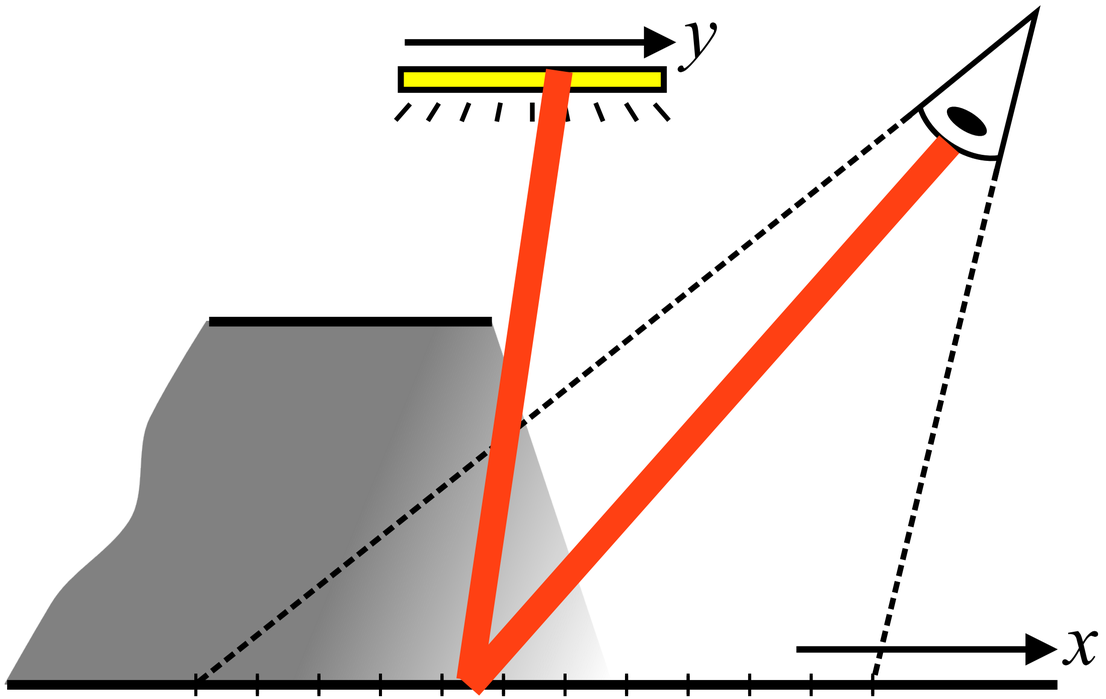


Fully Lit



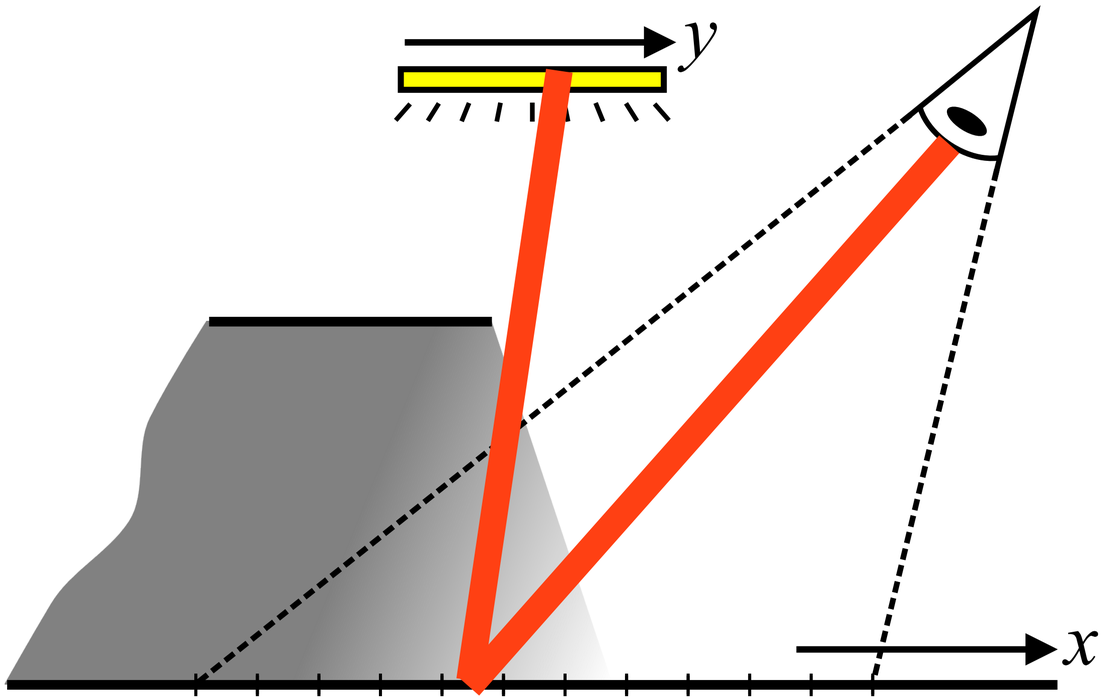


Metropolis Light Transport

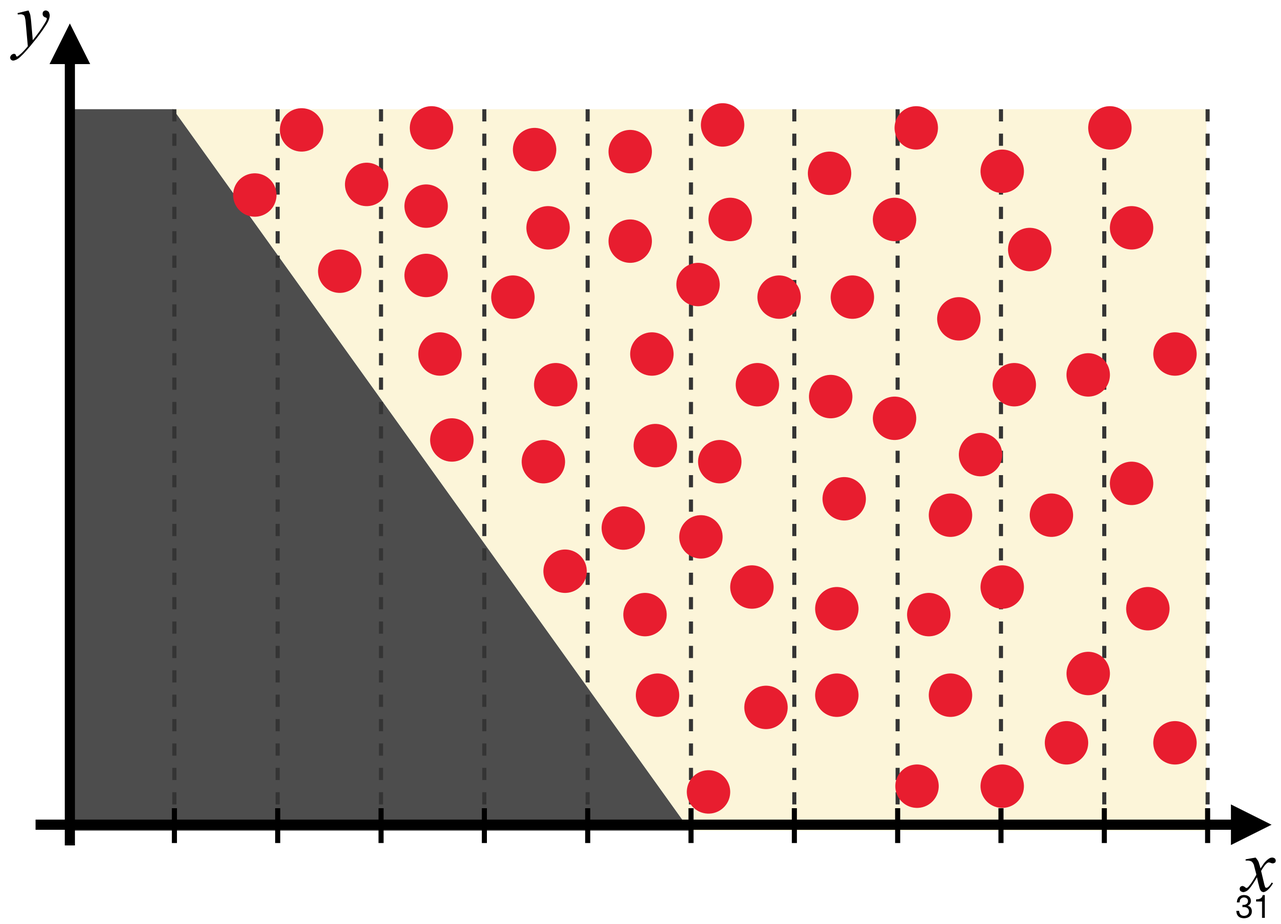


Samples distributed according to magnitude of f

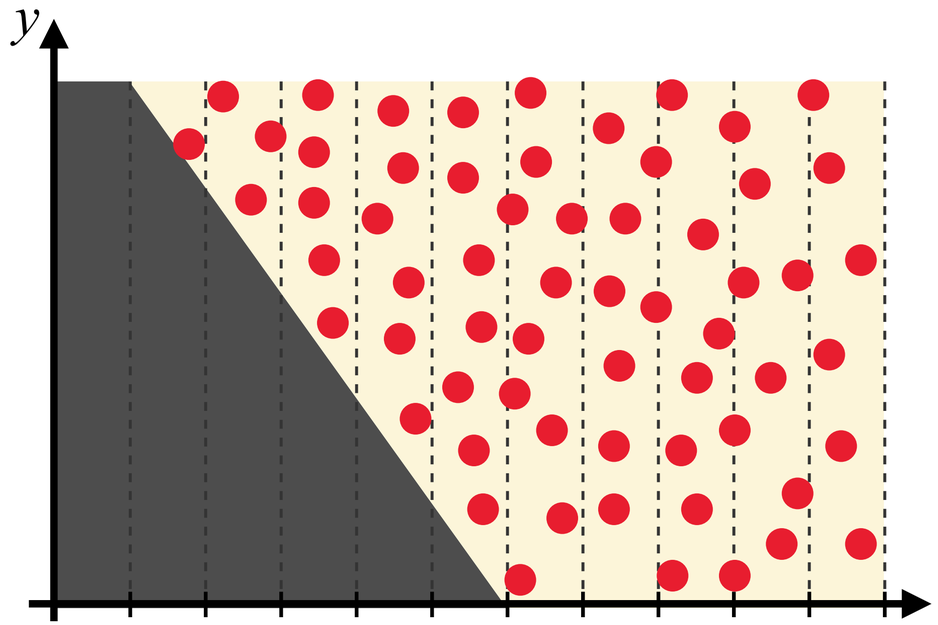
Metropolis Light Transport



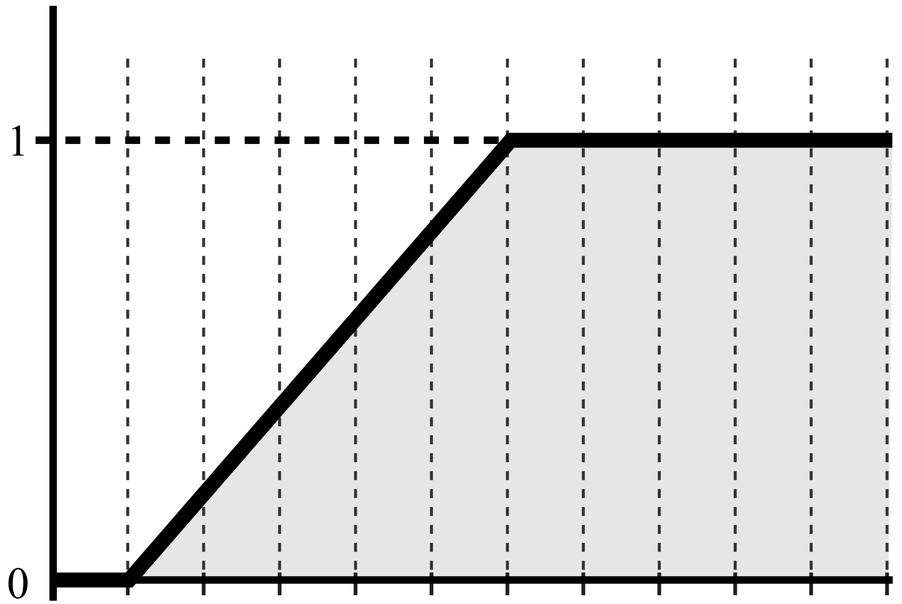
Samples distributed according to magnitude of f



Hmm.. Lots of Samples Where Nothing Happens

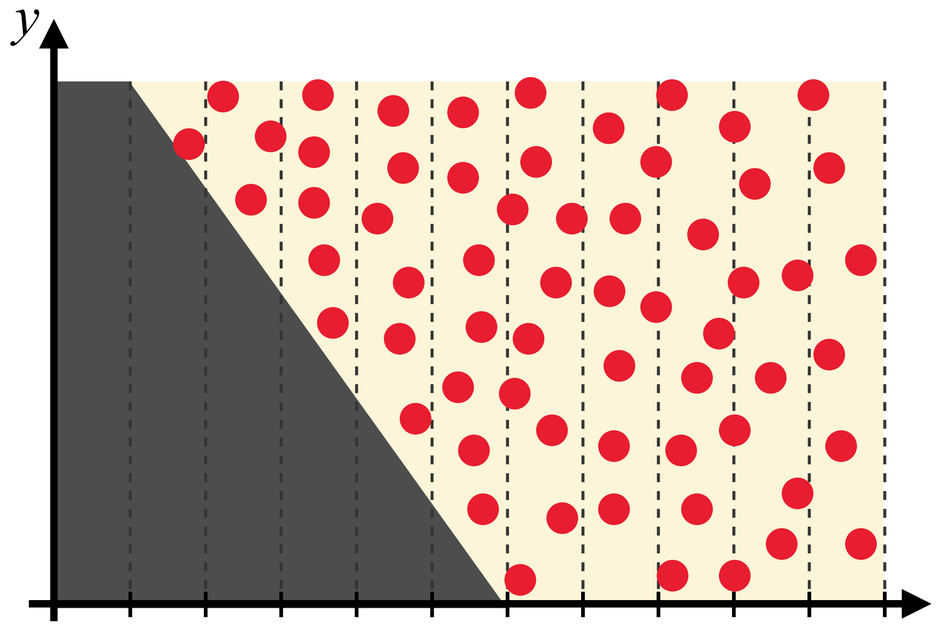


Metropolis Sample Distribution

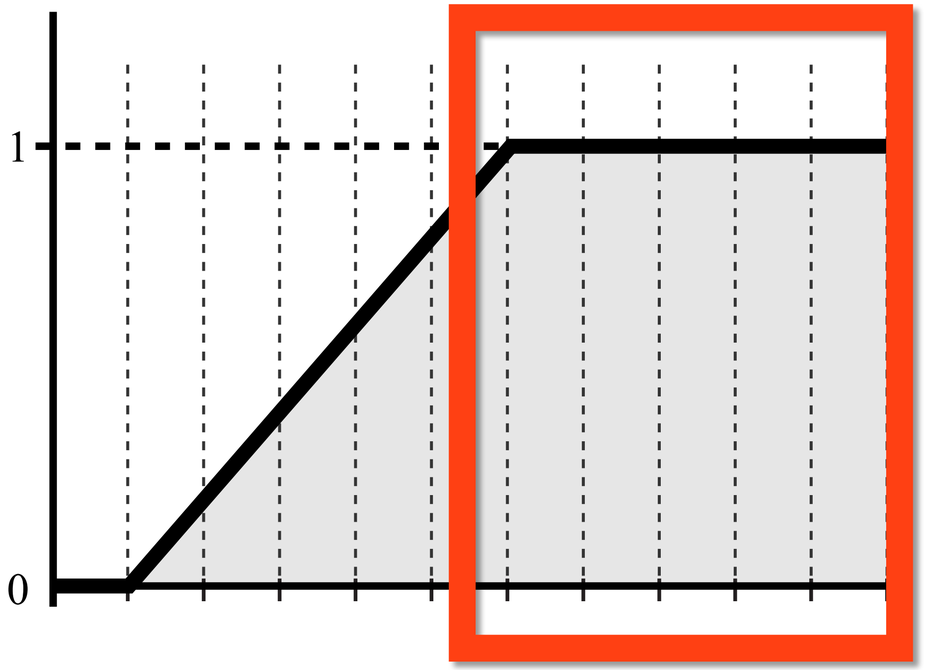


Rendering result

Hmm.. Lots of Samples Where Nothing Happens

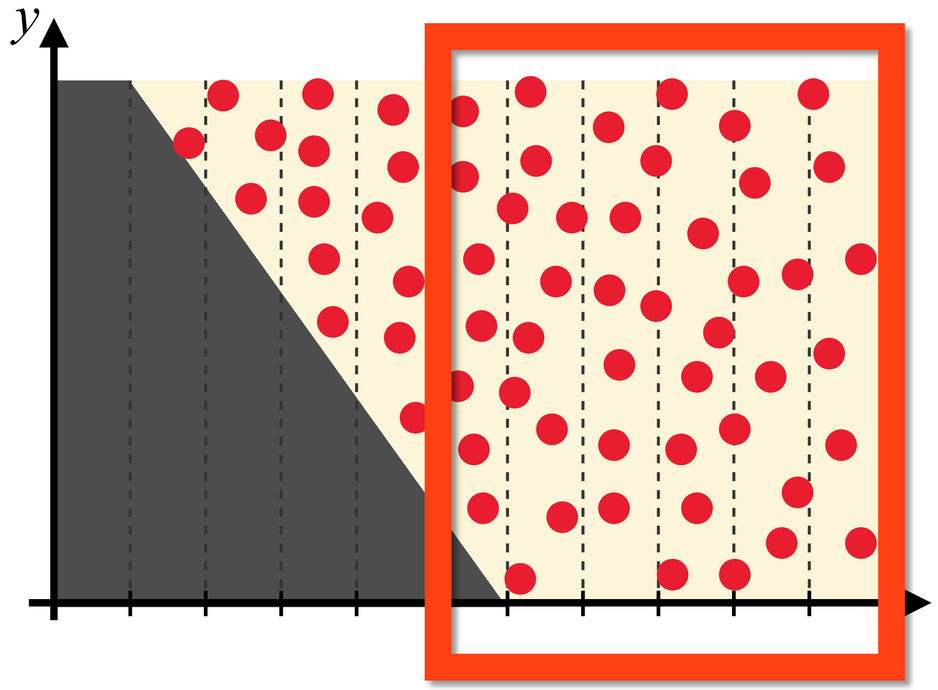


Metropolis Sample Distribution

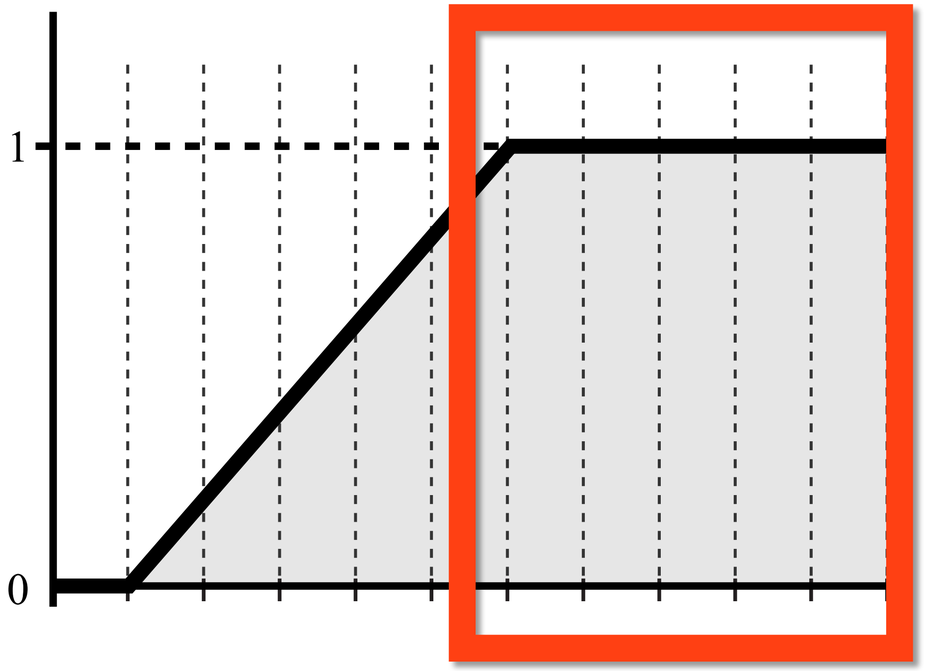


Rendering result

Hmm.. Lots of Samples Where Nothing Happens



Metropolis Sample Distribution



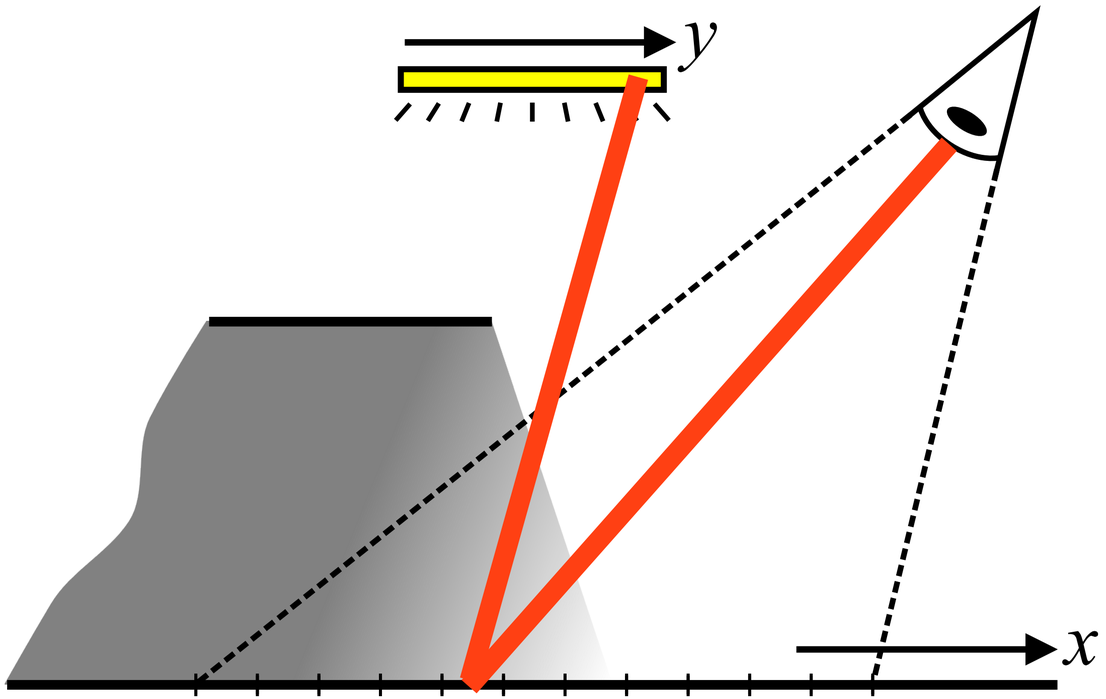
Rendering result

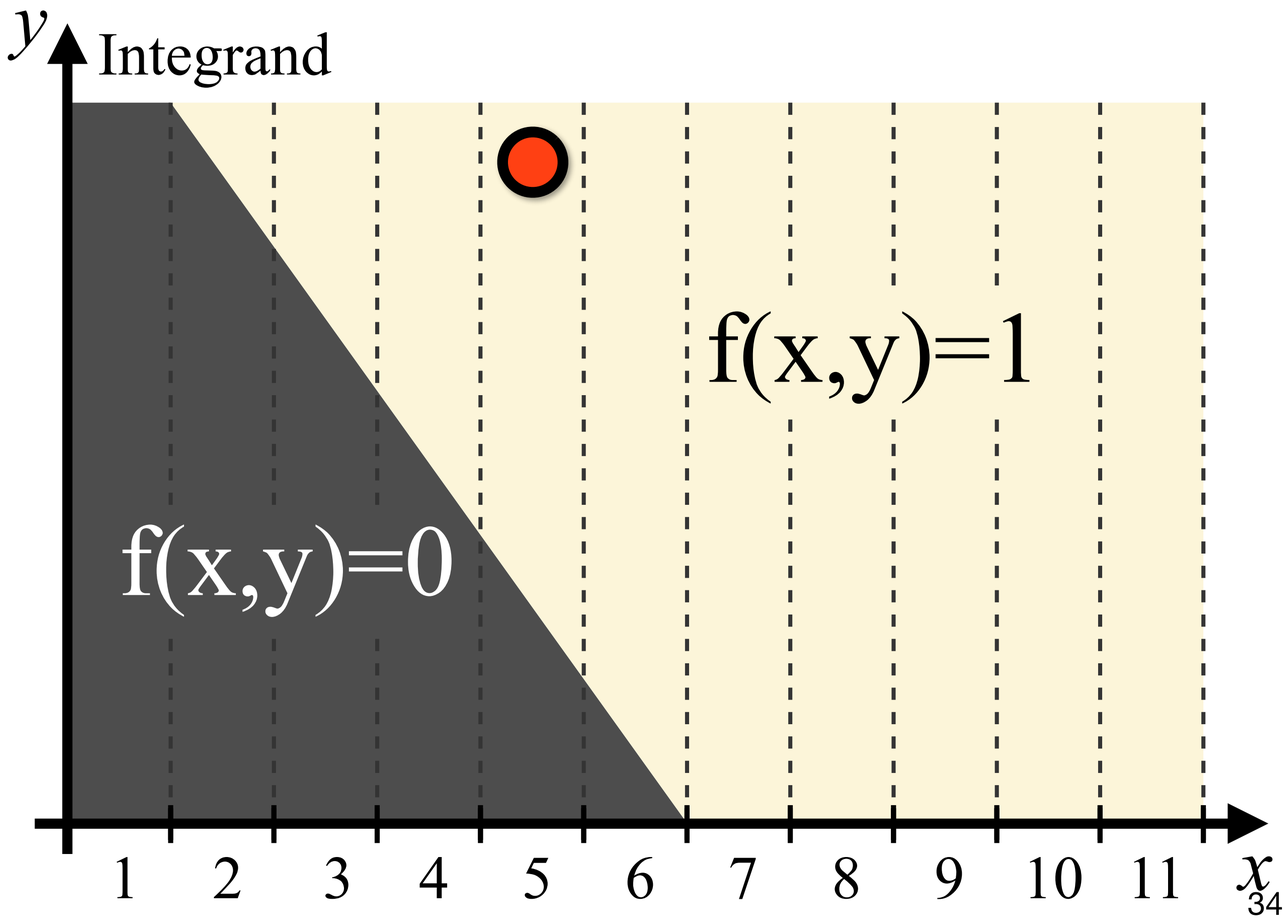
Key Idea

Let’s make each sample measure
the difference in the integrand across one pixel

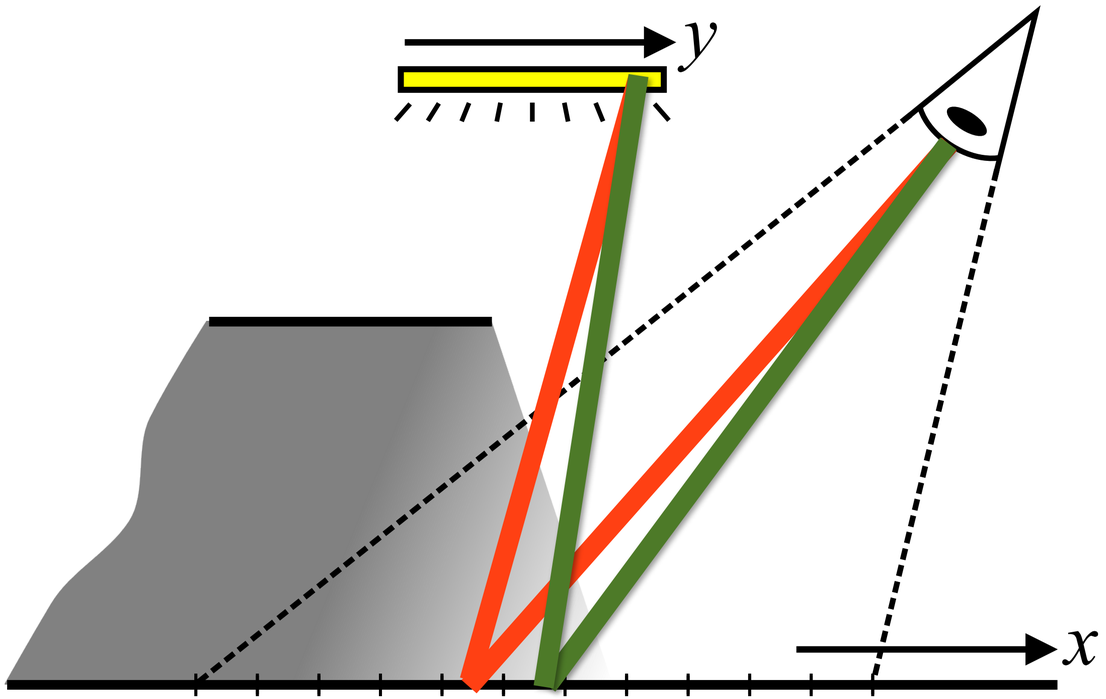
Think image gradients

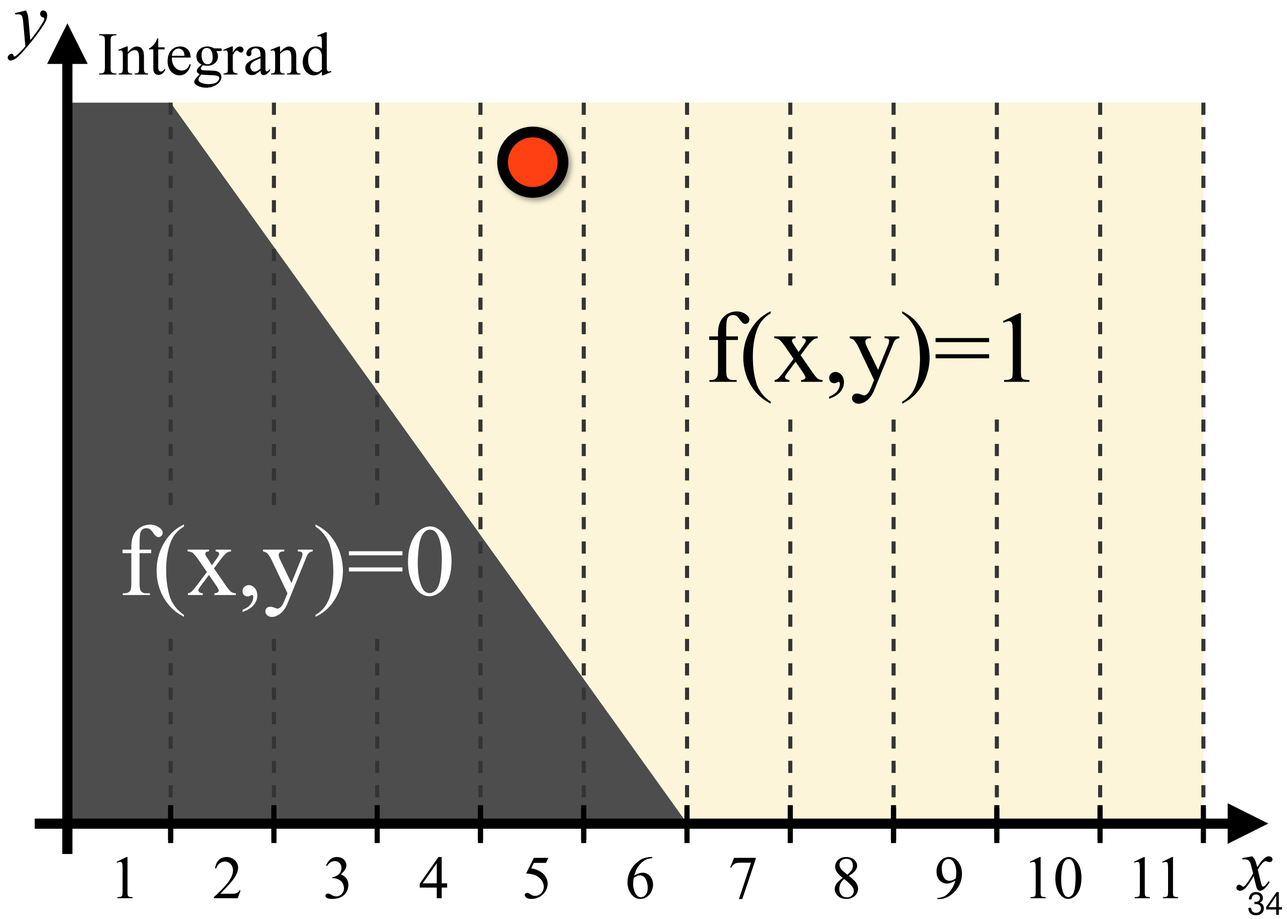
Path Space
Finite Differences



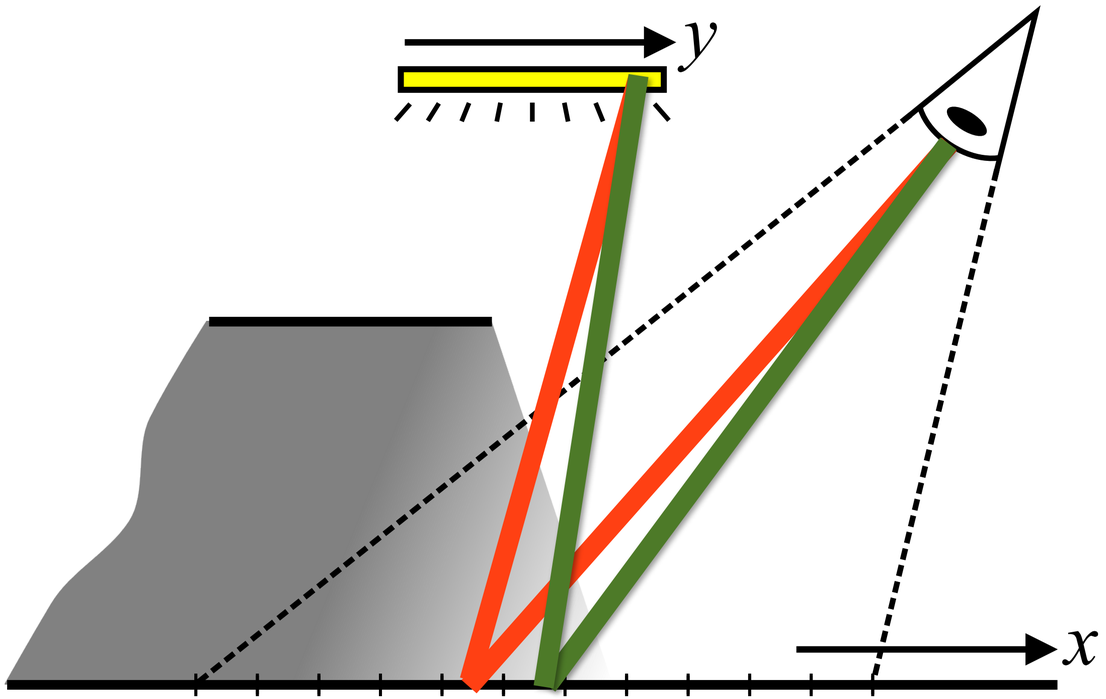


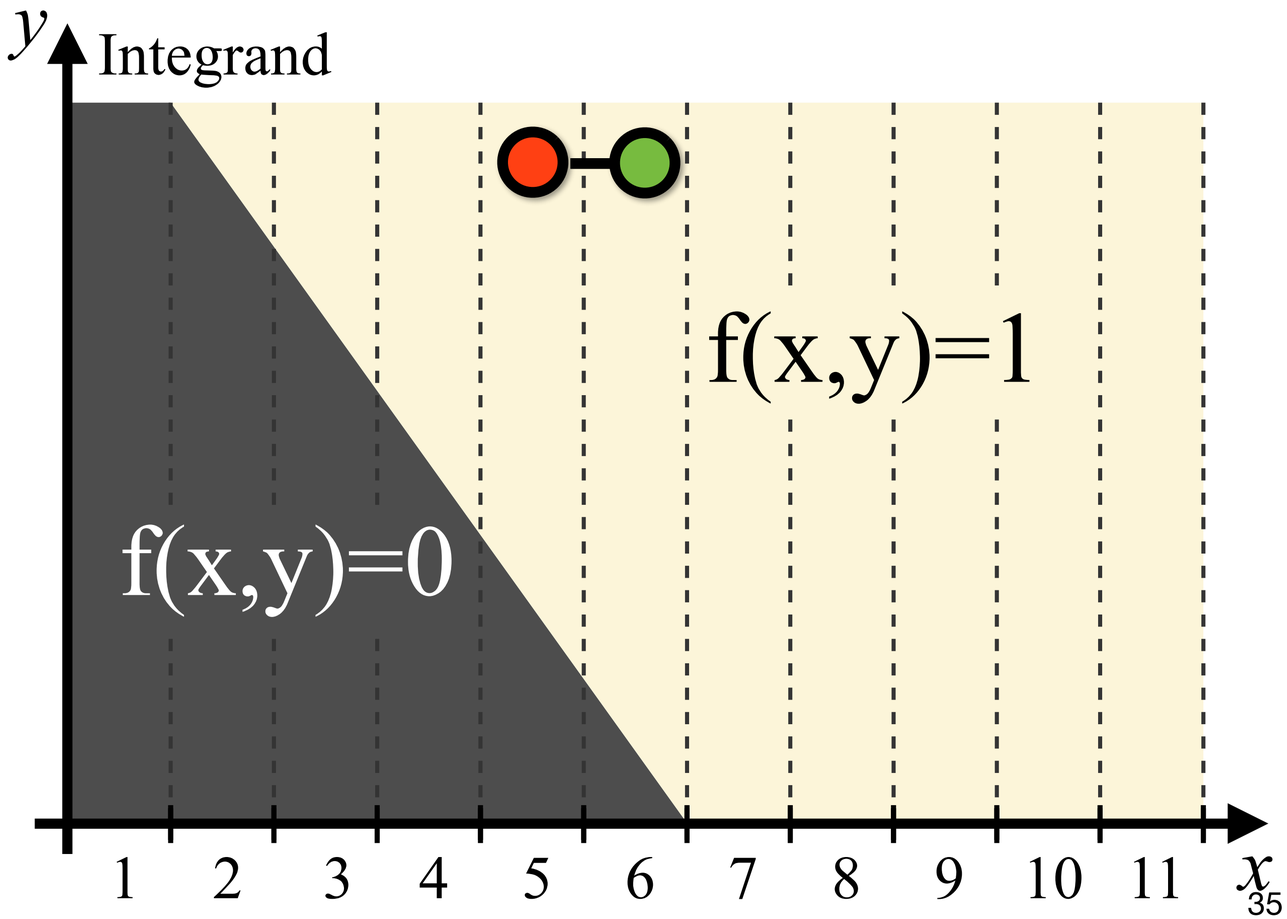
Path Space
Finite Differences



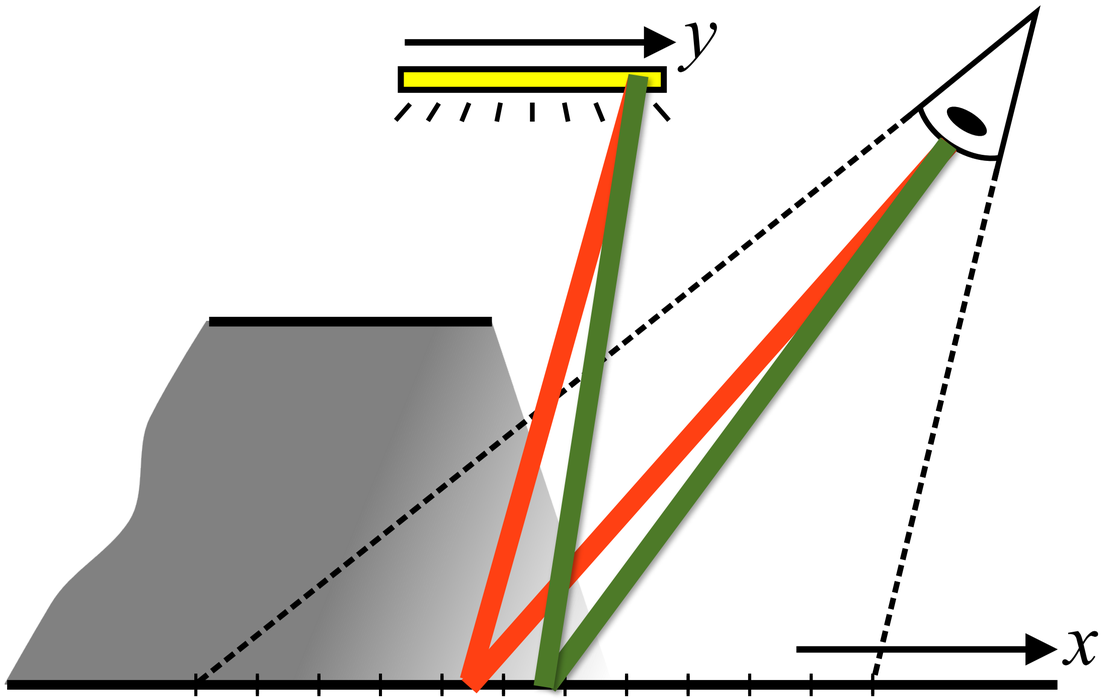


Path Space
Finite Differences

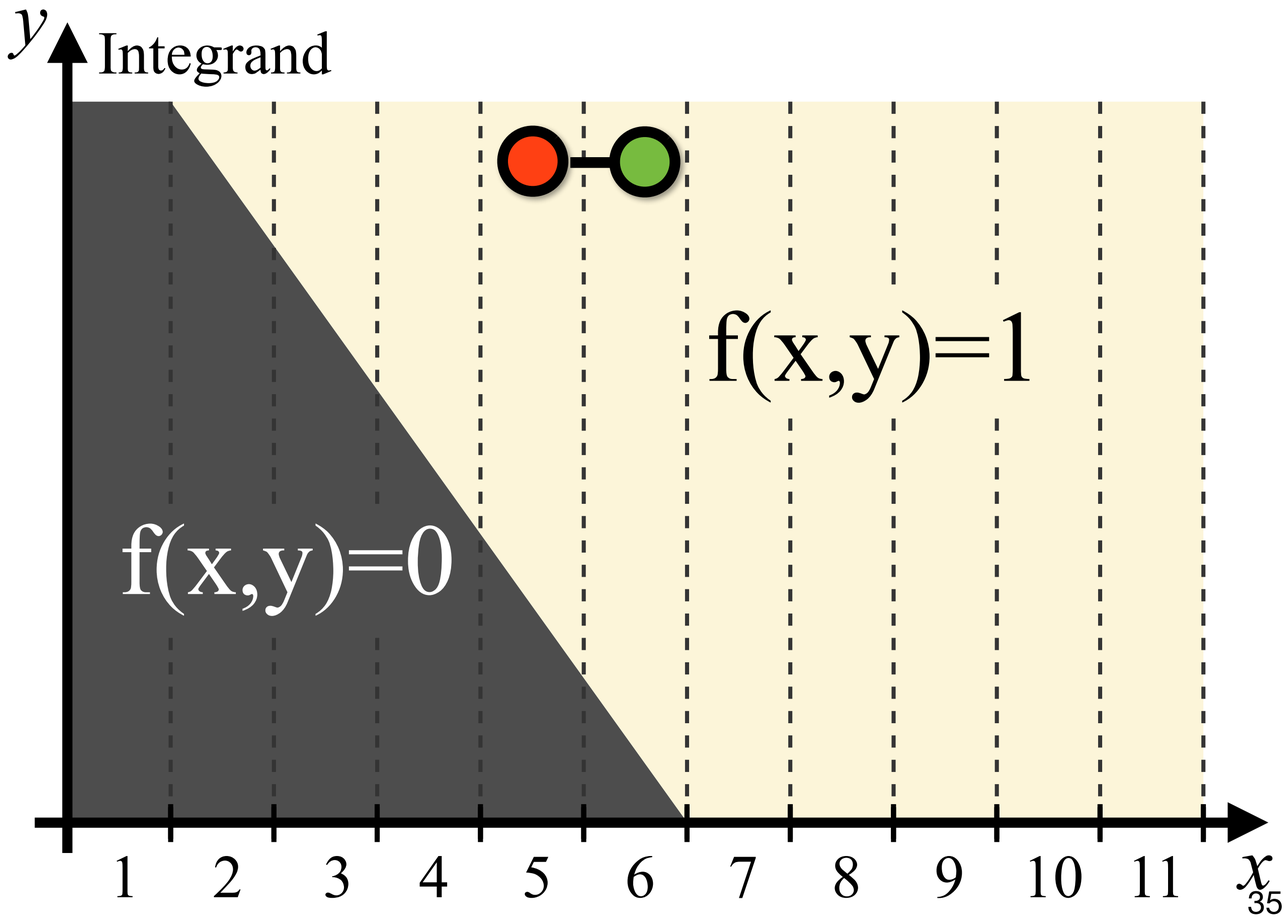




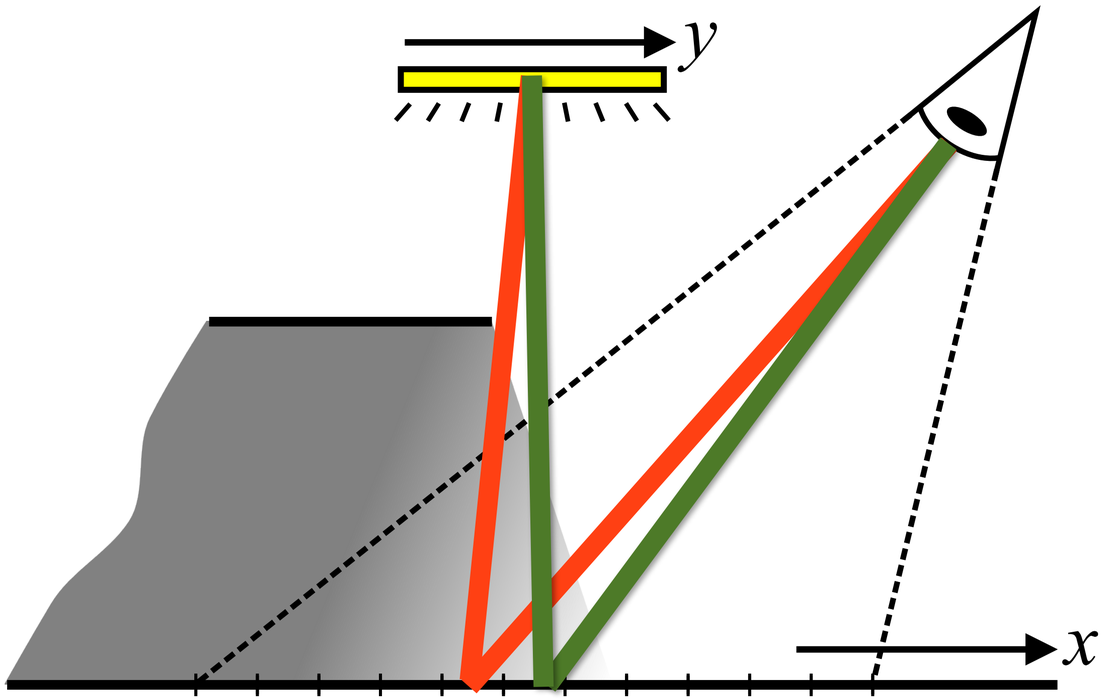
Path Space
Finite Differences



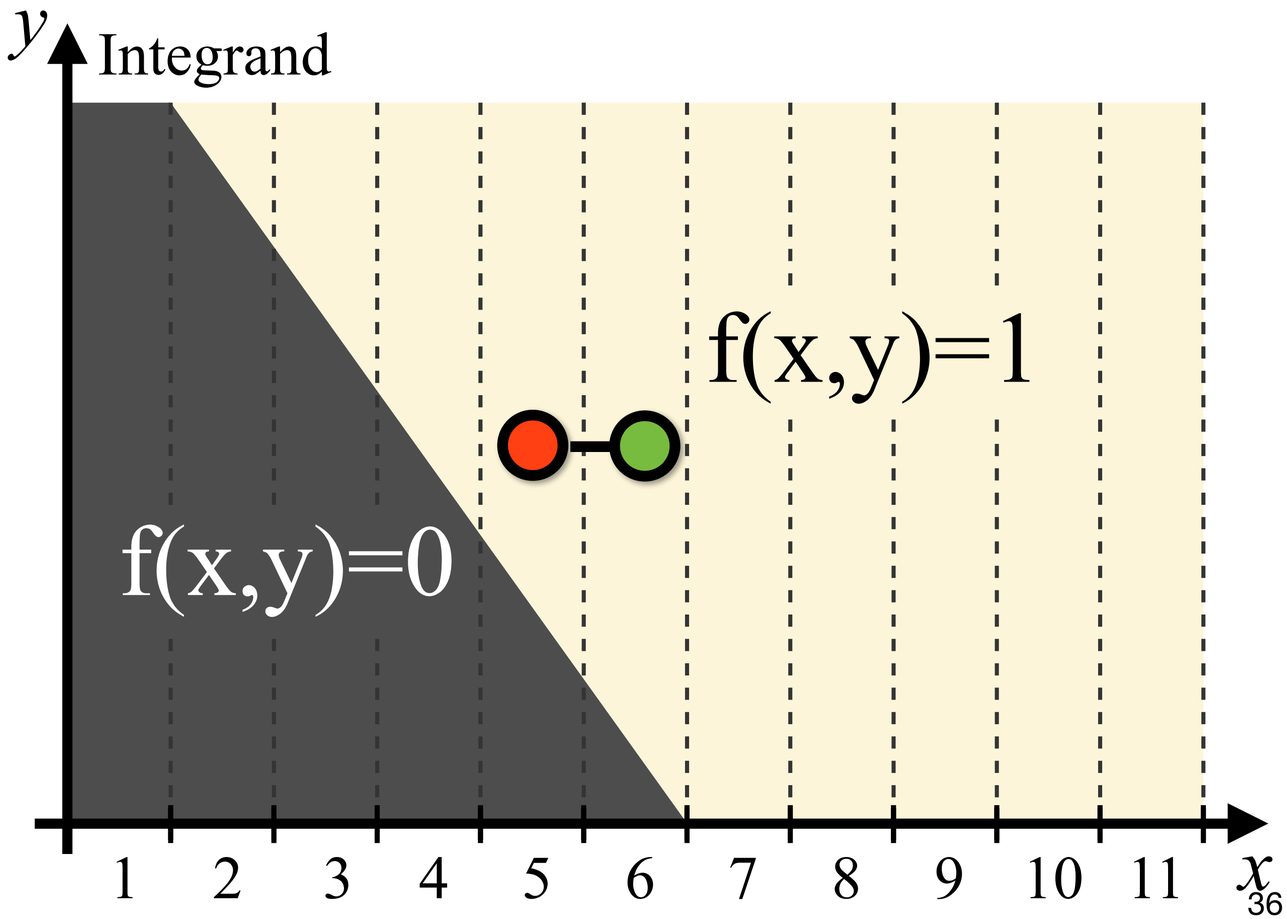
Both paths unblocked,
difference is zero



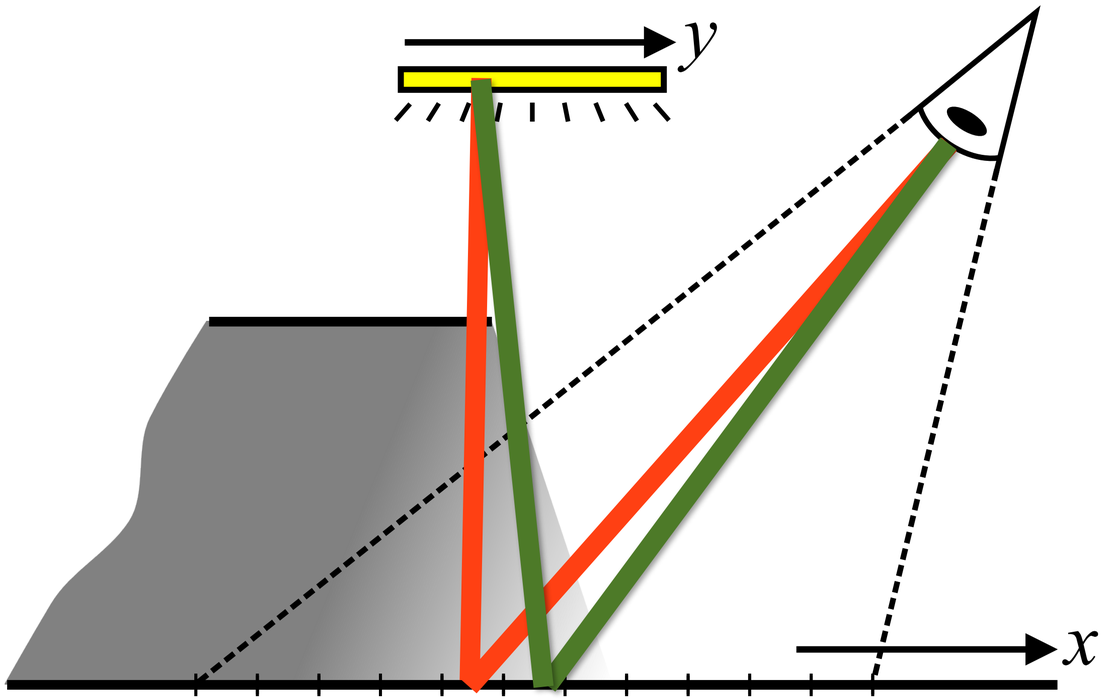
Path Space
Finite Differences



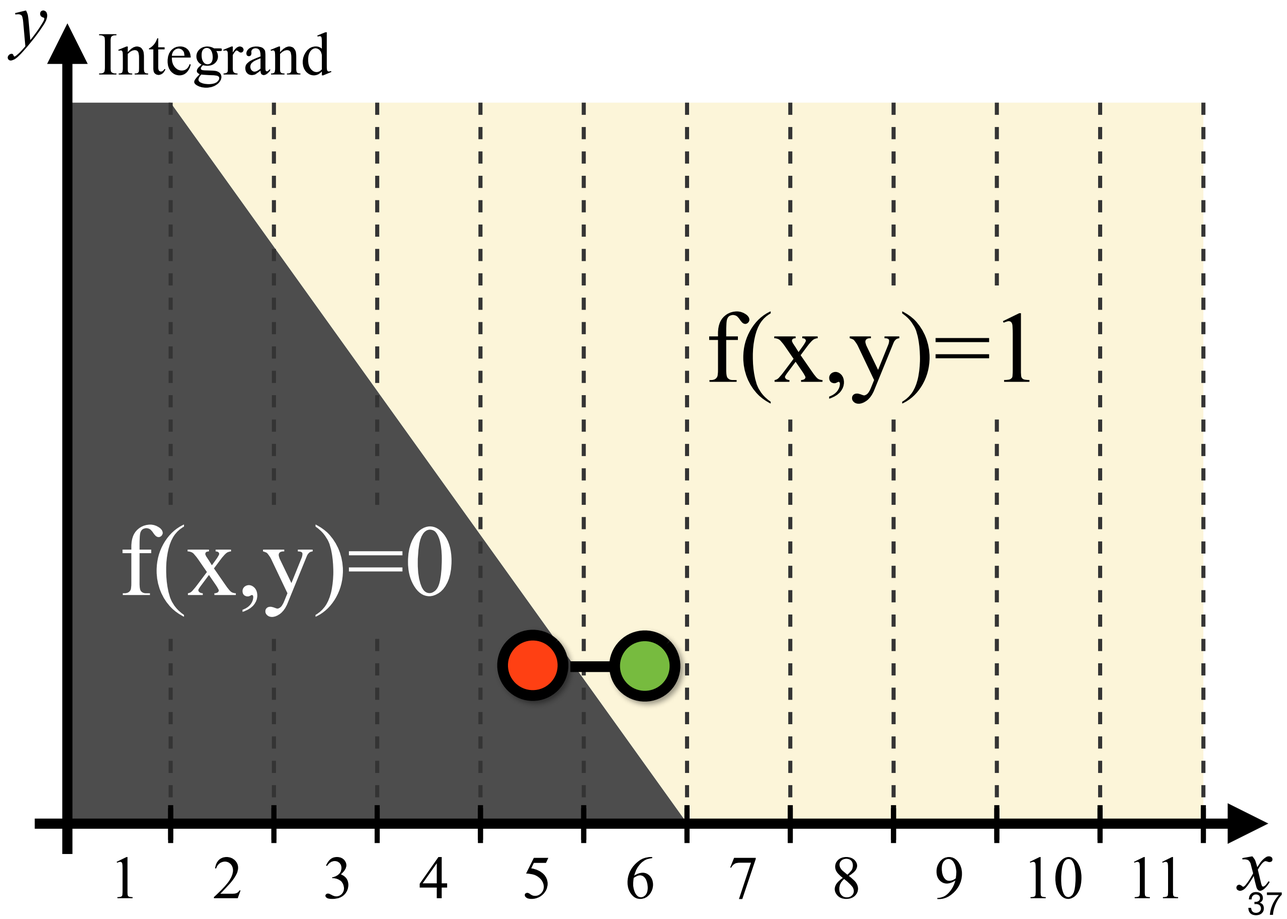
Both paths unblocked,
difference is zero



Path Space
Finite Differences



One path blocked,
abs(difference)=1



Gradient-Domain Metropolis Sampling

Gradient-Domain Metropolis Sampling

Each sample is a pair of paths

Gradient-Domain Metropolis Sampling

Each sample is a pair of paths

We drive Metropolis by the **difference in path throughput**

— Random process walks along **regions of change in path space**

— Result: **many of samples where stuff happens**

Gradient-Domain Metropolis Sampling

Each sample is a pair of paths

We drive Metropolis by the **difference in path throughput**

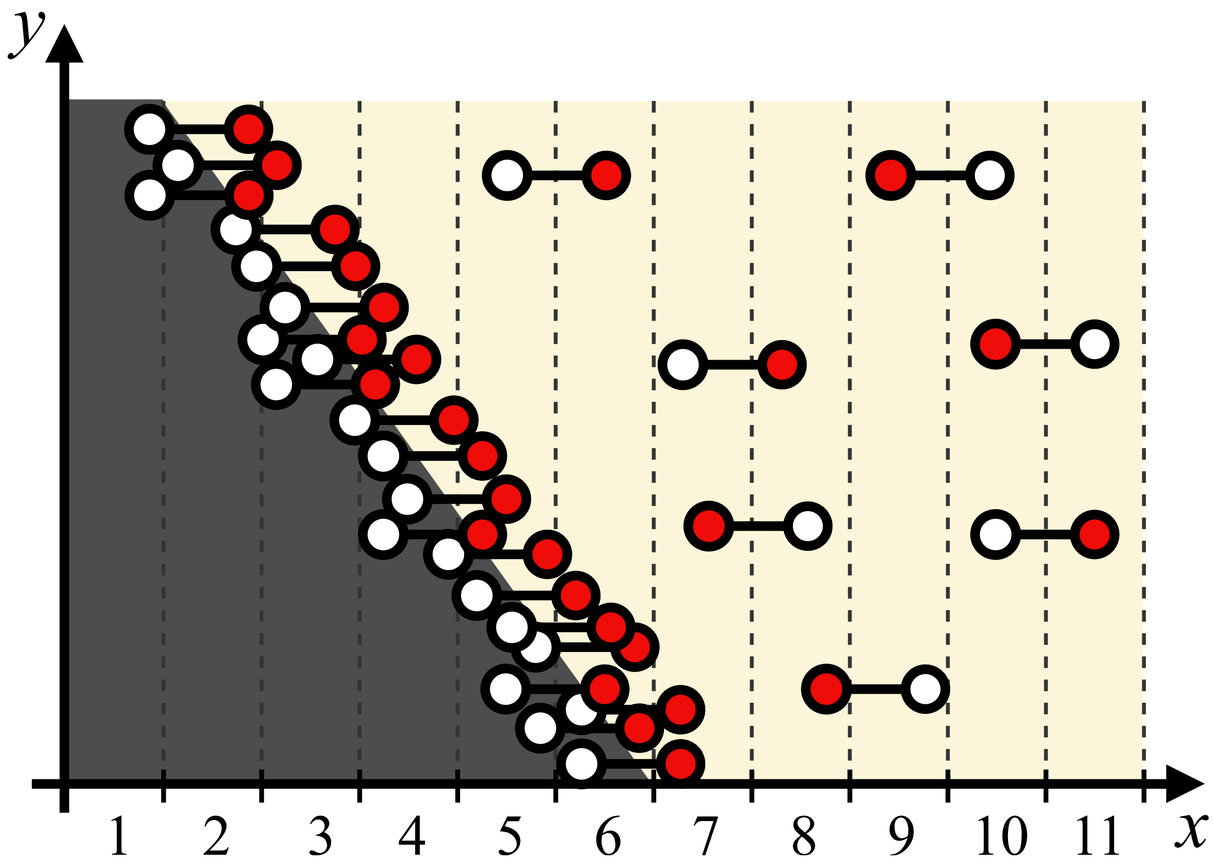
— Random process walks along **regions of change in path space**

— Result: **many of samples where stuff happens**

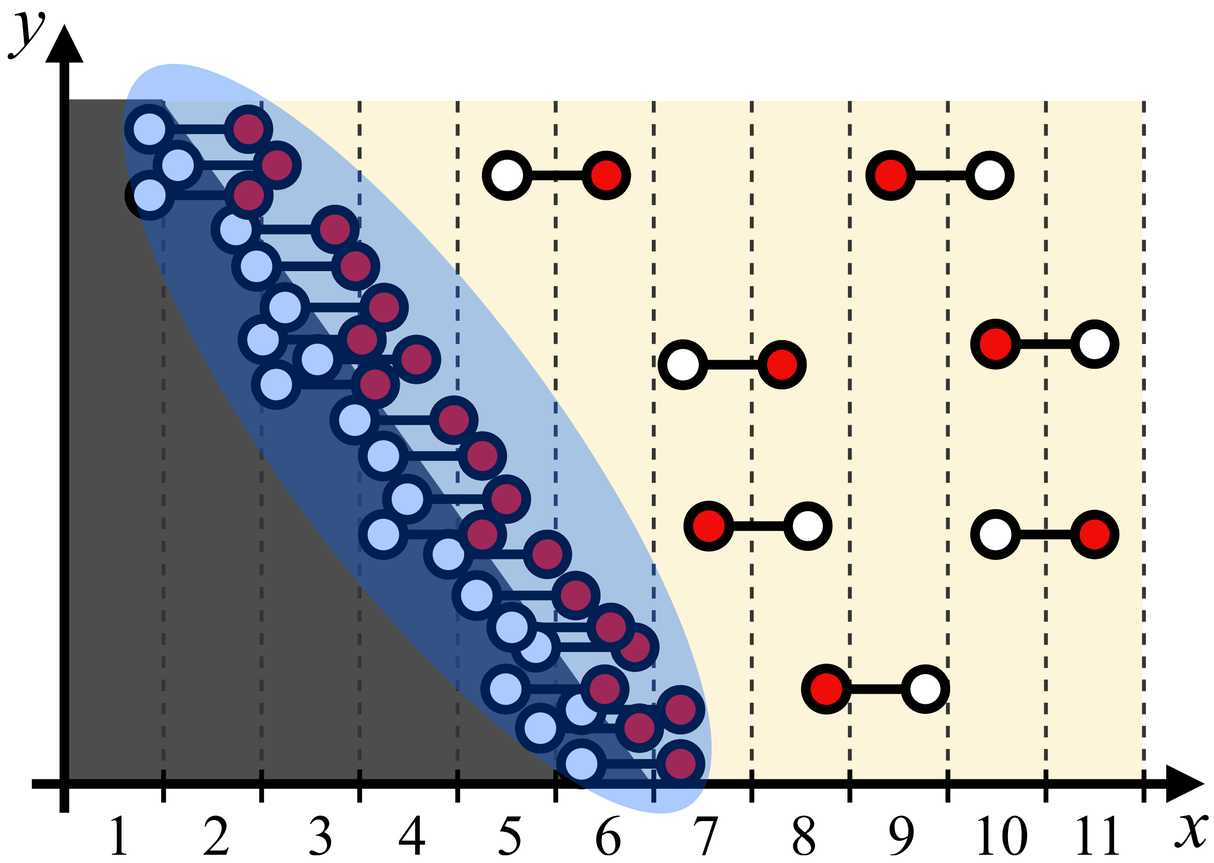
Small detail: We also mix in a little of “primal” path throughput

— So we get some samples everywhere

Gradient-Domain Sampling Result

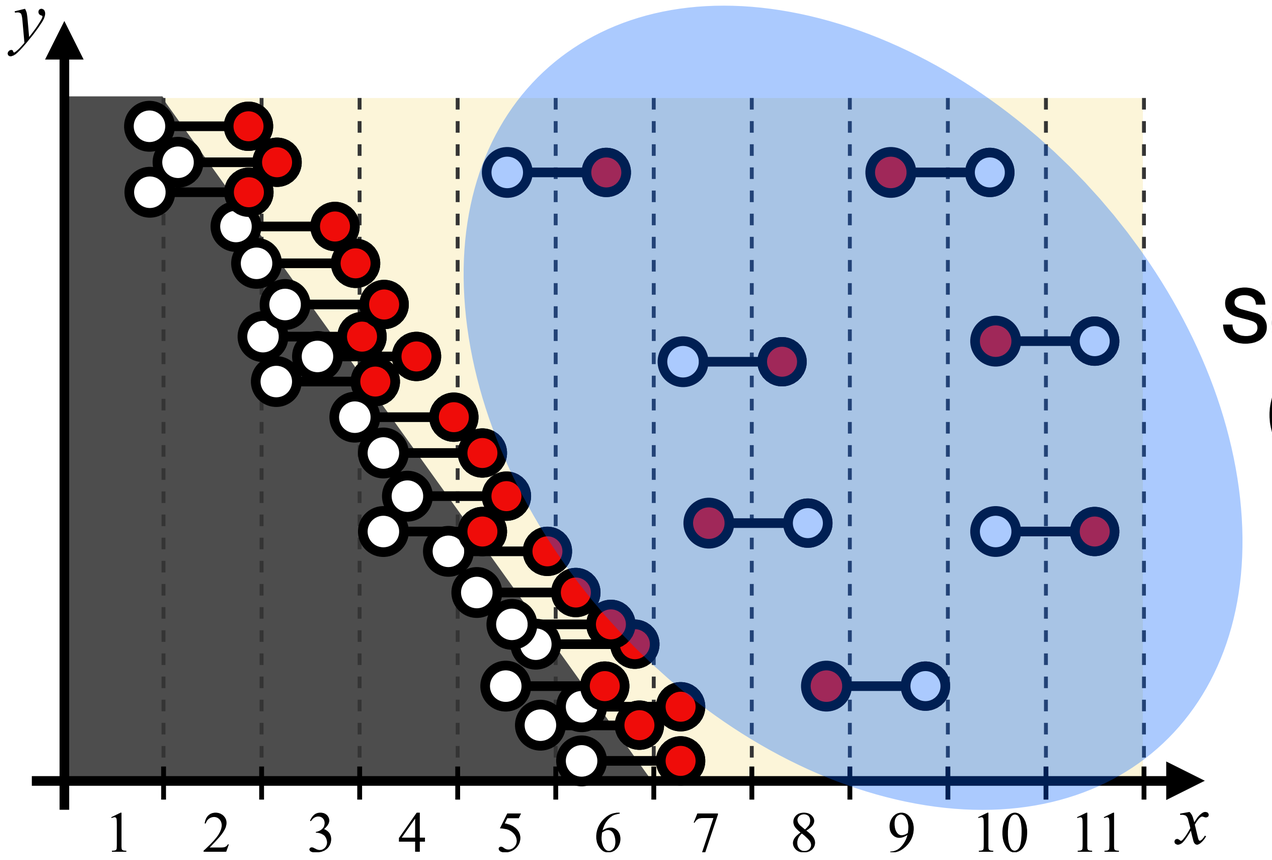


Gradient-Domain Sampling Result



Many samples along shadow boundary

Gradient-Domain Sampling Result



Many samples along shadow boundary

Some samples in lit region (so we get an idea of its magnitude)

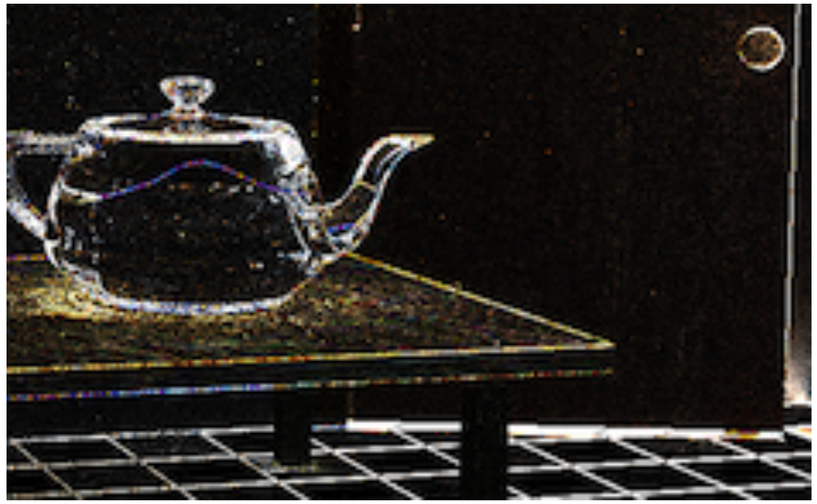
Gradient-Domain MLT

When done: **count positive and negative contributions in each pixel**

Result**: three images, all unbiased**

— horizontal and vertical pixel differences (**gradient**)

— “primal” path throughput (**noisy version of image**)



Gradient-Domain MLT Final Step

Integrate gradient by solving screened Poisson equation [Bhat09]

— Coarse image helps low frequencies

— **With standard L_2 solver, final image is unbiased**

— In practice, use more robust L_1 solver instead of usual L_2 [Levin04]

Questions

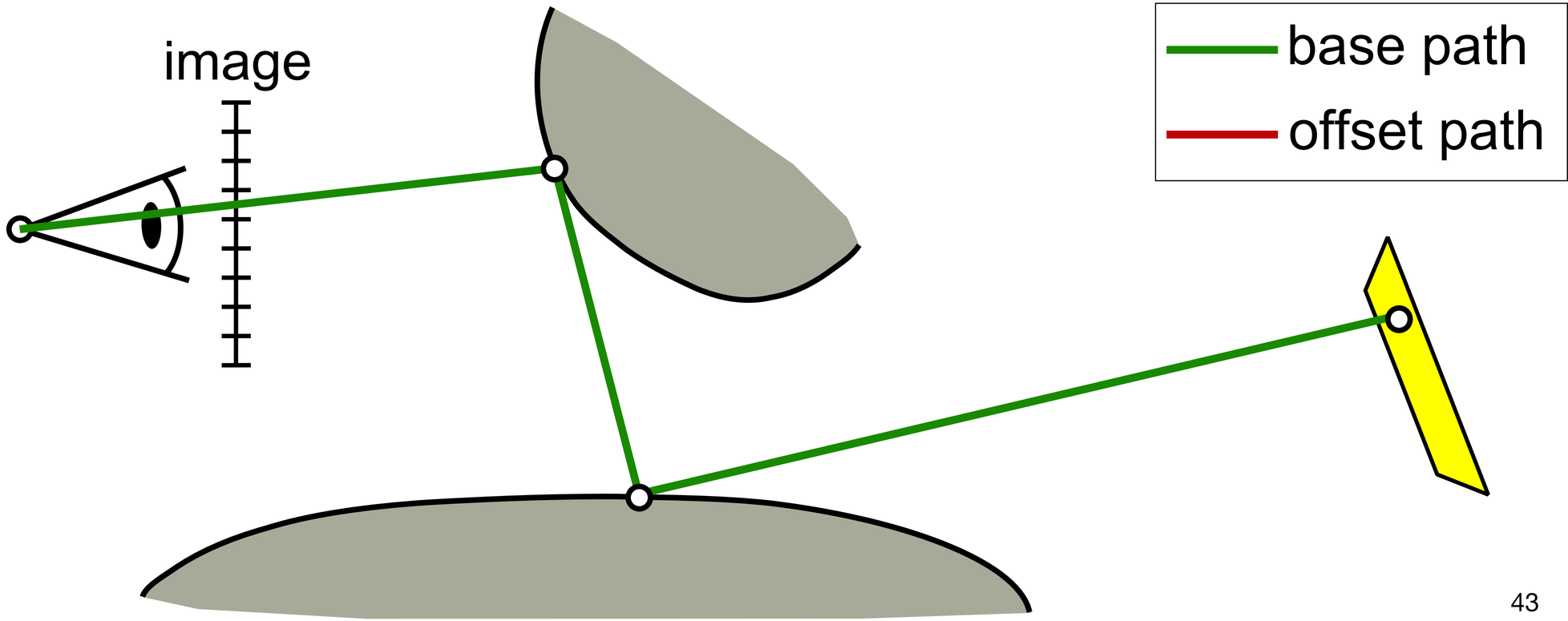
Real light paths have more than two segments

— How to construct the “offset path”?

Is the result the actual finite difference?

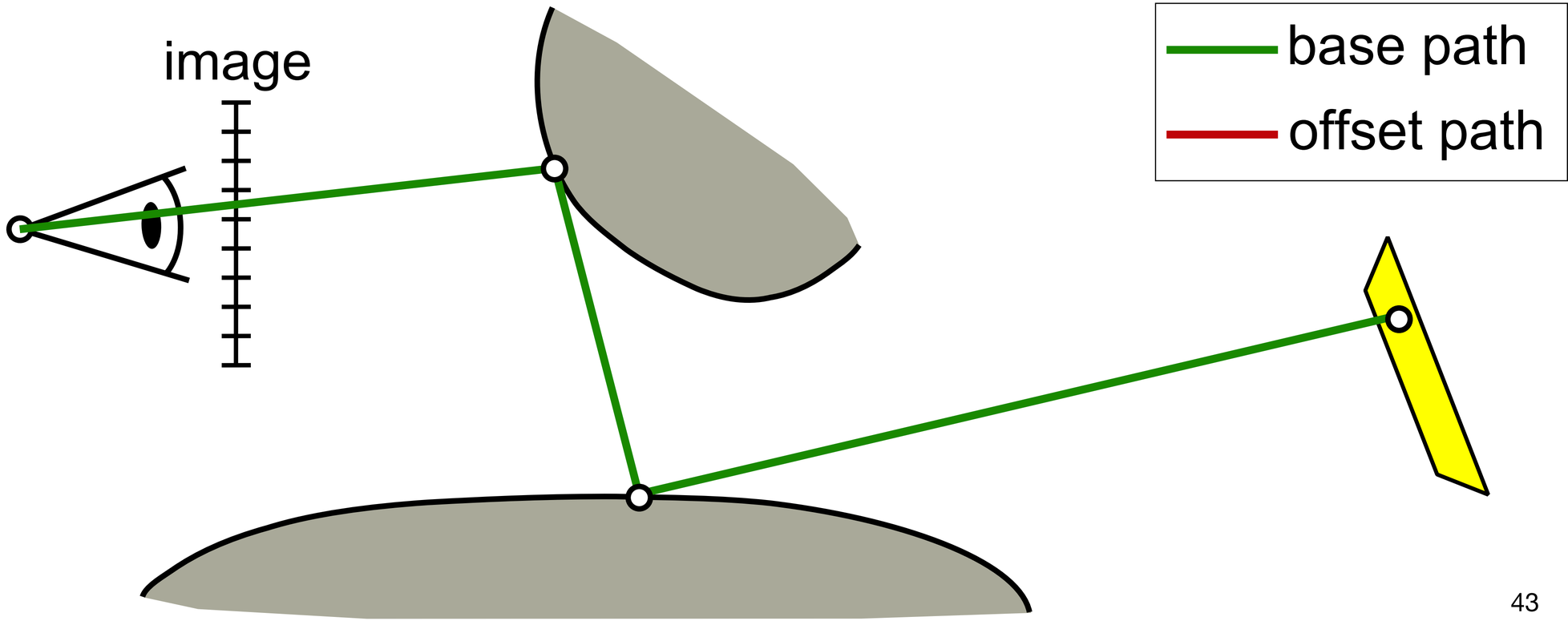
— Subtlety involved

Offset Path, Simple Case



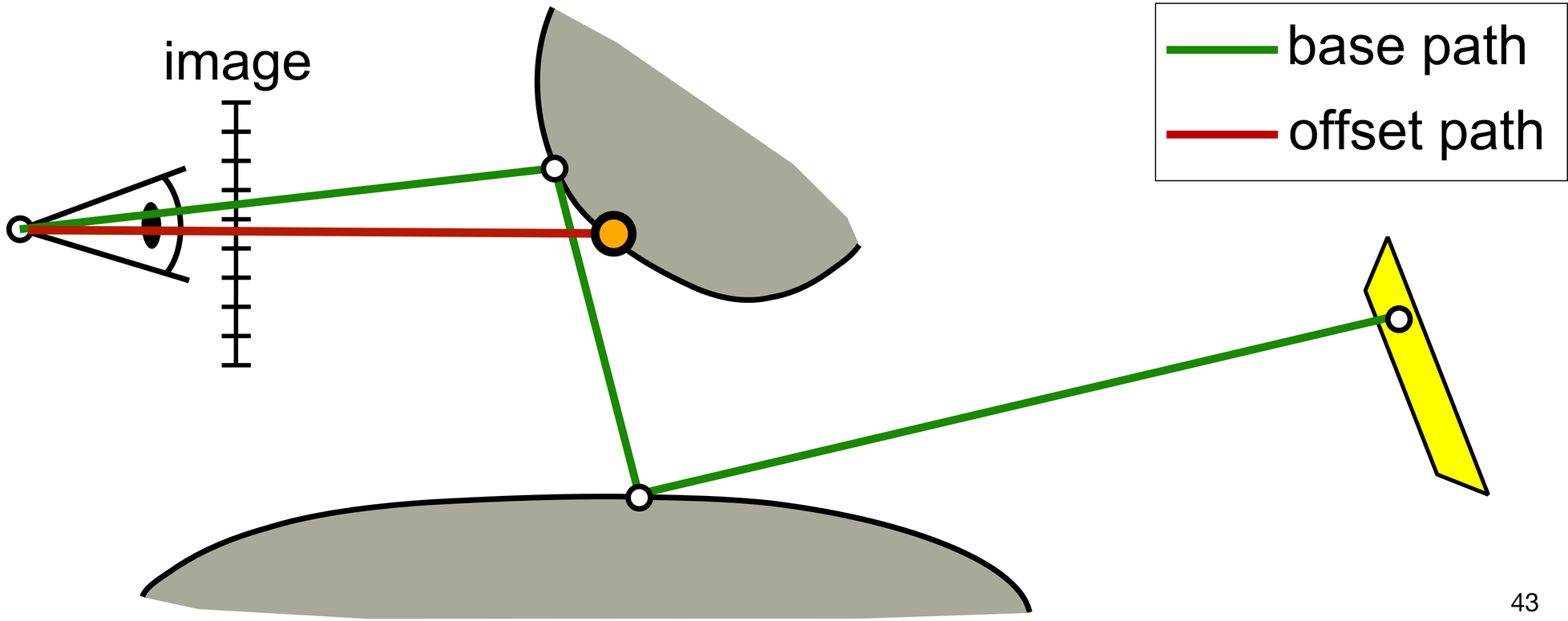
Offset Path, Simple Case

take green “base path”, shift one pixel, shoot new primary ray



Offset Path, Simple Case

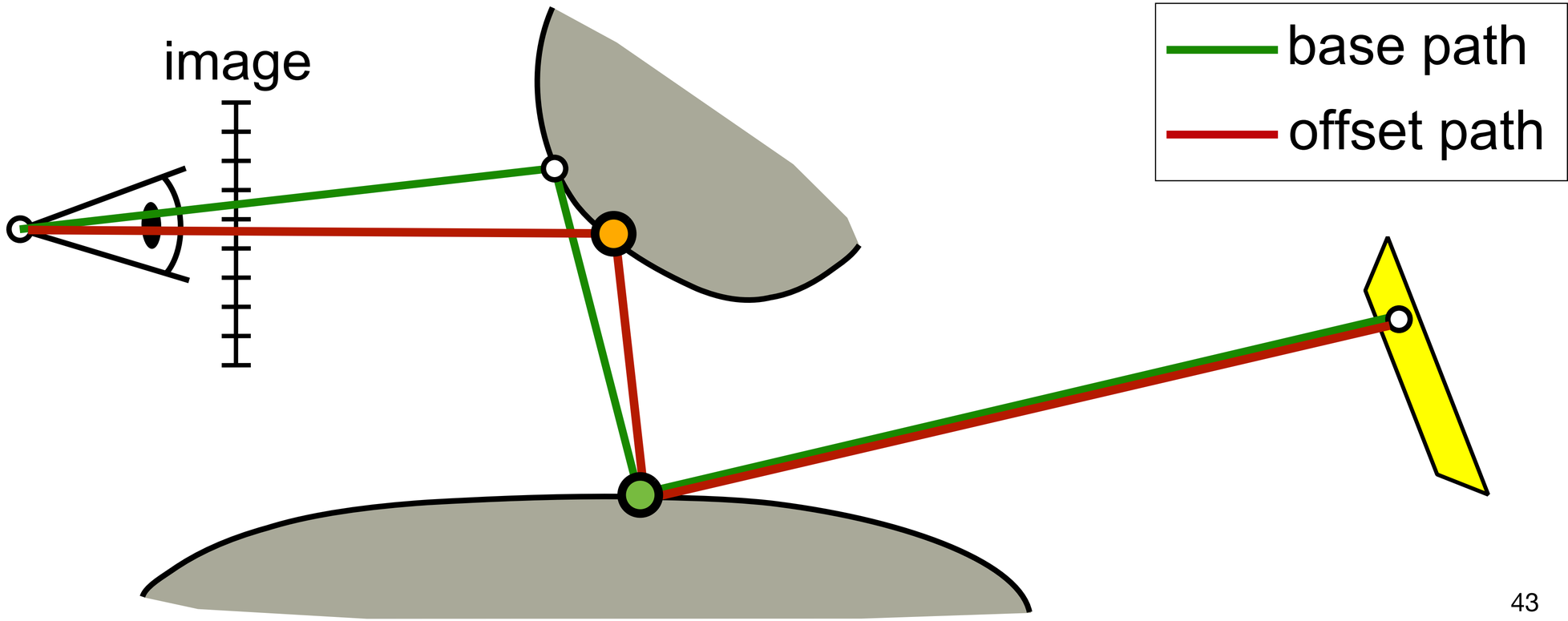
take green “base path”, shift one pixel, shoot new primary ray



Offset Path, Simple Case

take green “base path”, shift one pixel, shoot new primary ray

connect new primary hit ● to old secondary hit ●

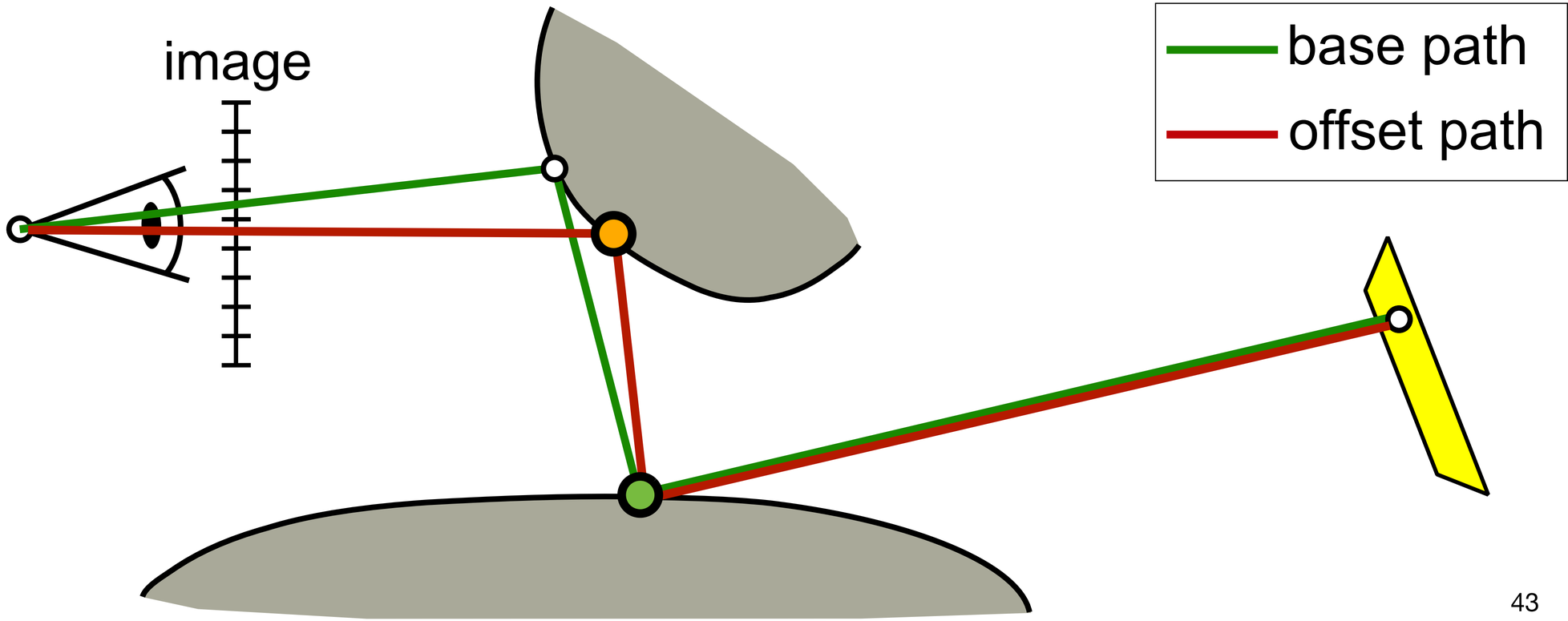


Offset Path, Simple Case

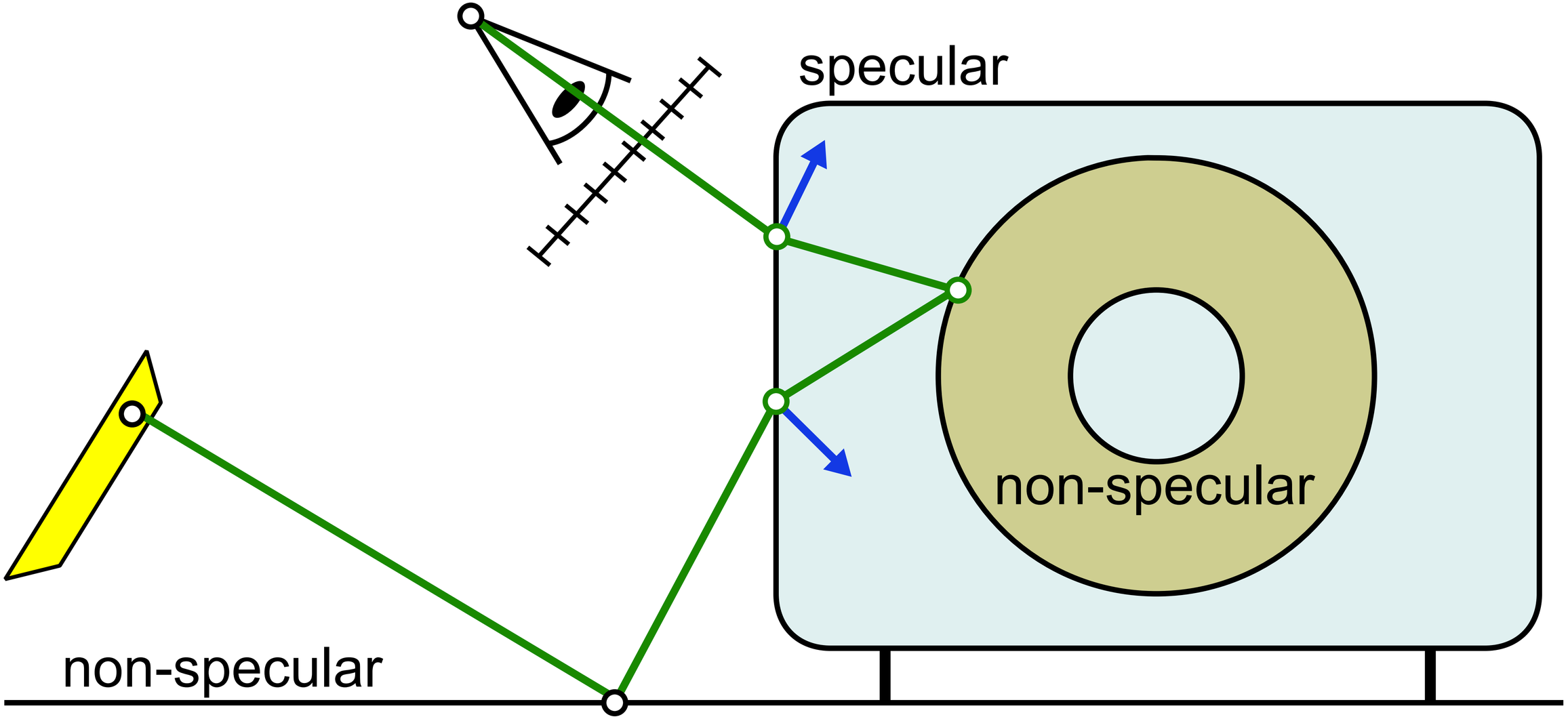
take green “base path”, shift one pixel, shoot new primary ray

connect new primary hit ● to old secondary hit ●

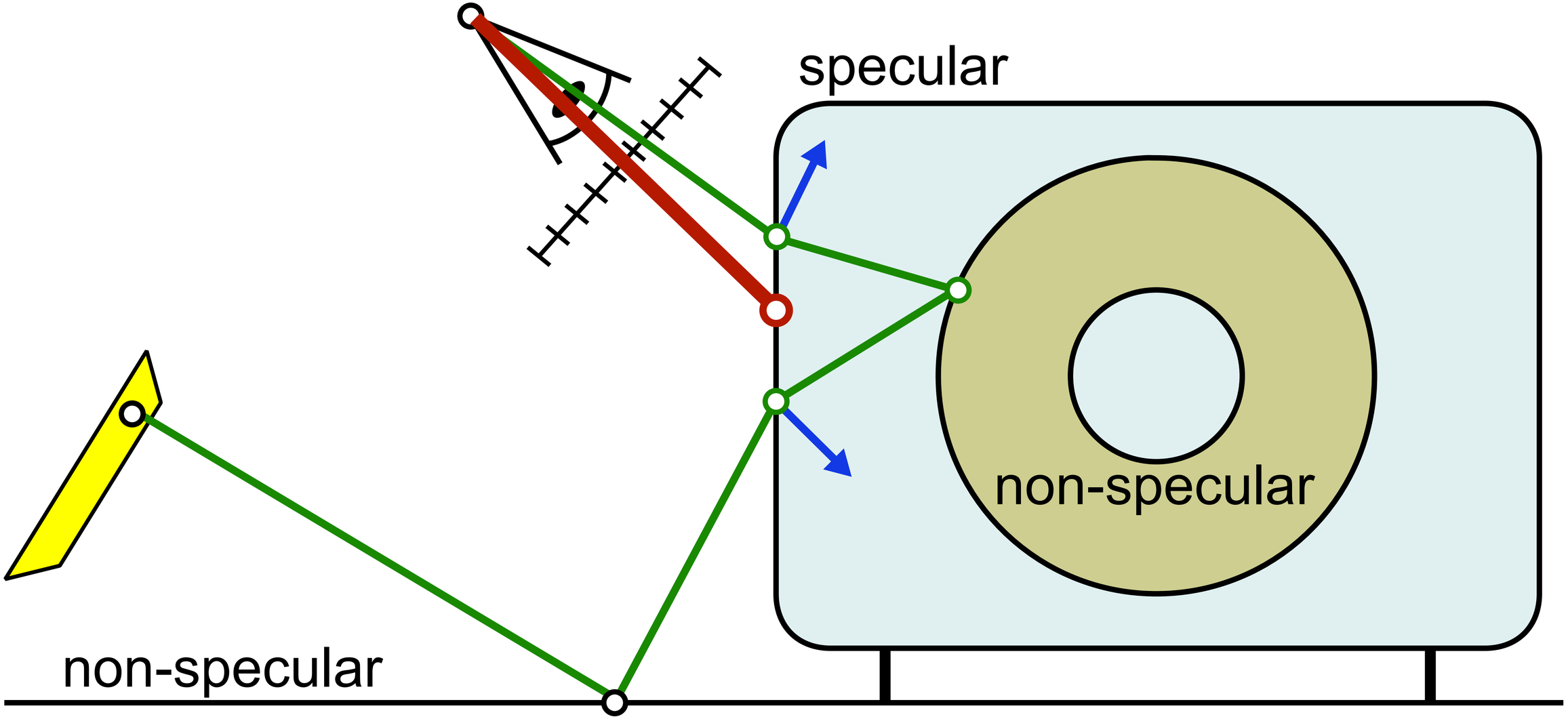
done!



Specularity is More Complex

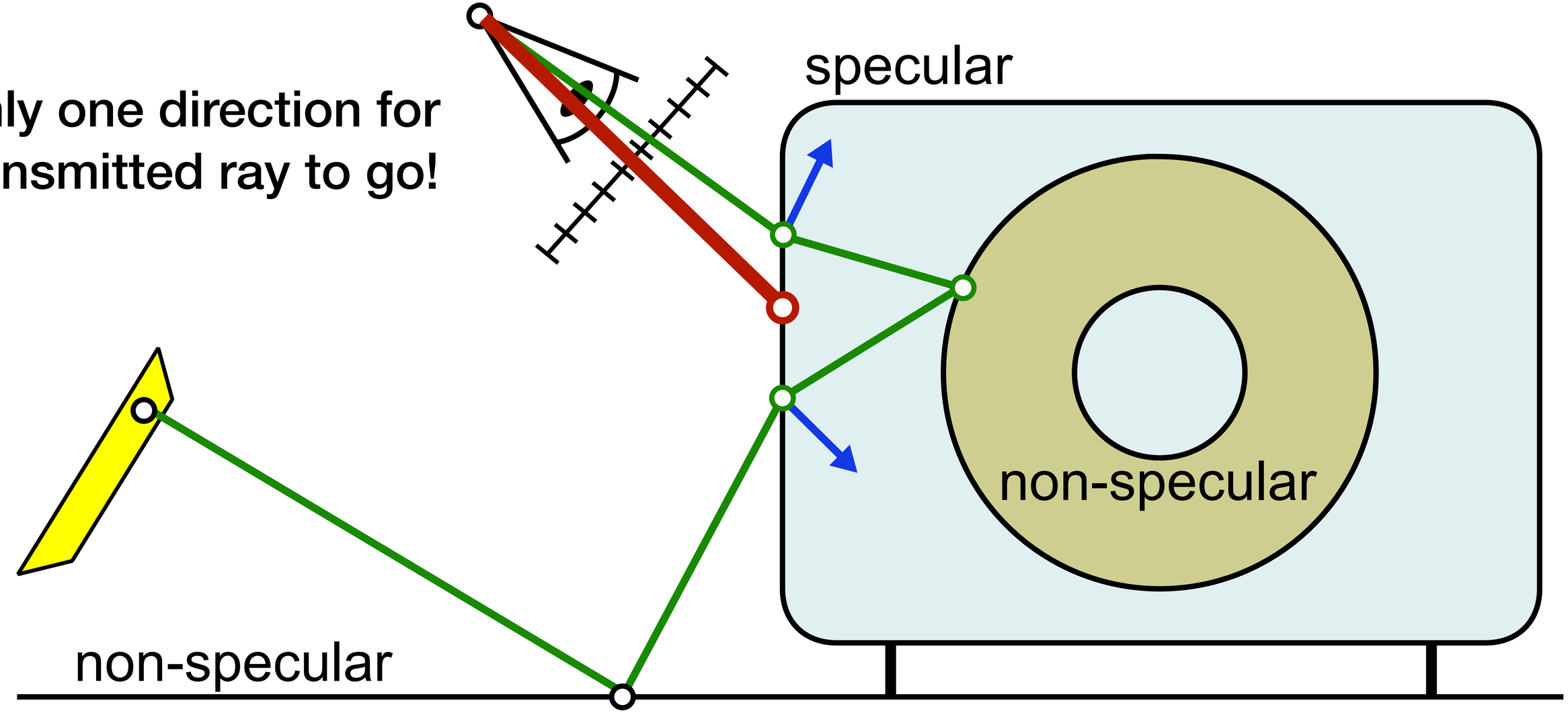


Specularity is More Complex



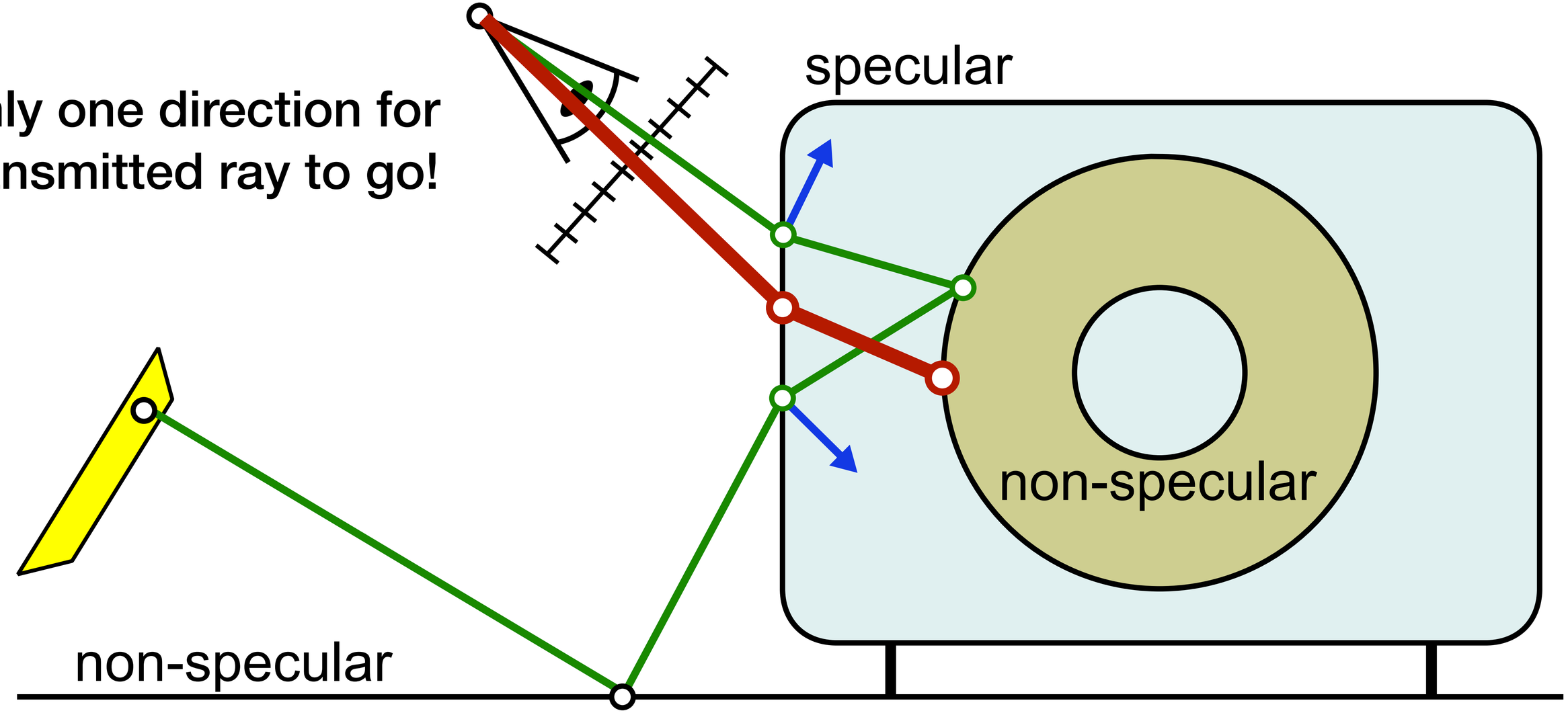
Specularity is More Complex

Only one direction for transmitted ray to go!



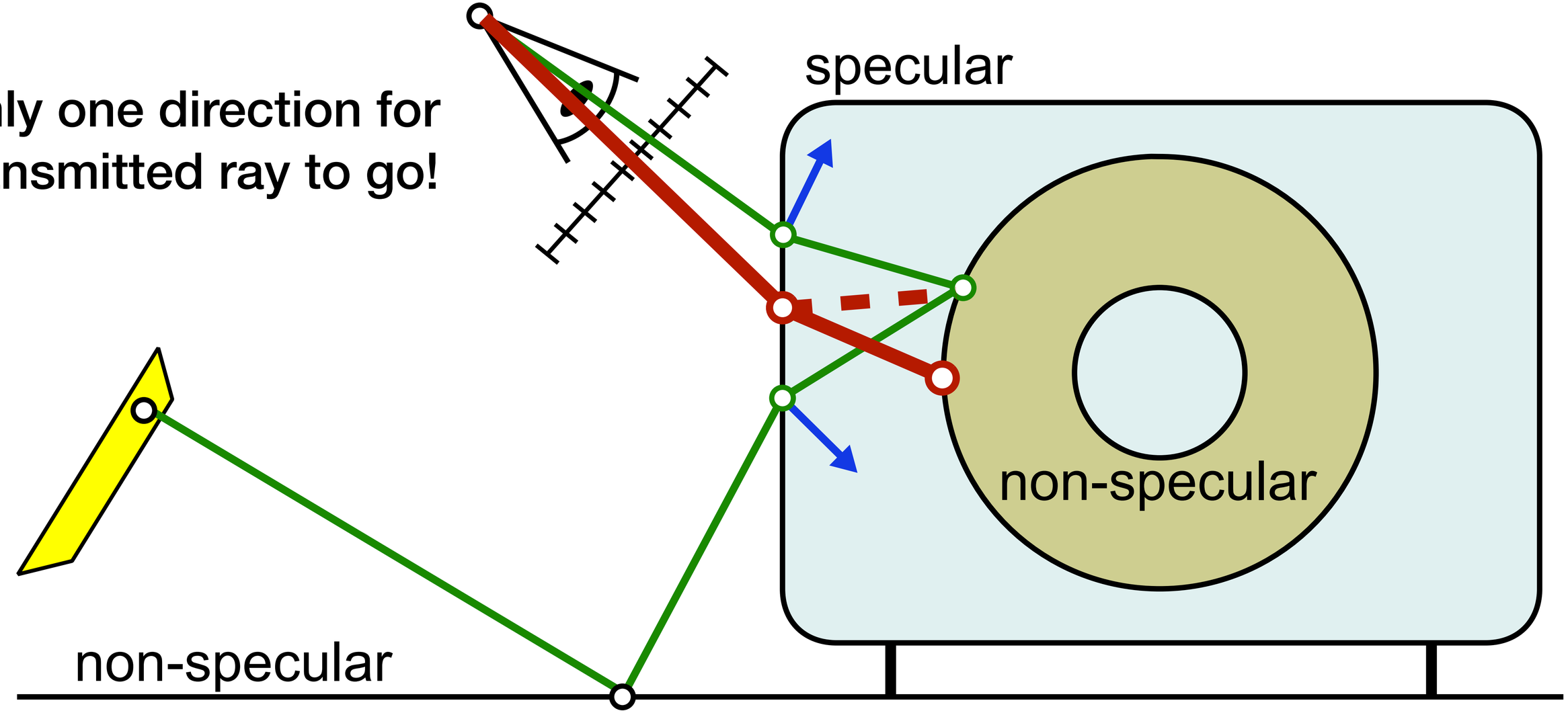
Specularity is More Complex

Only one direction for transmitted ray to go!

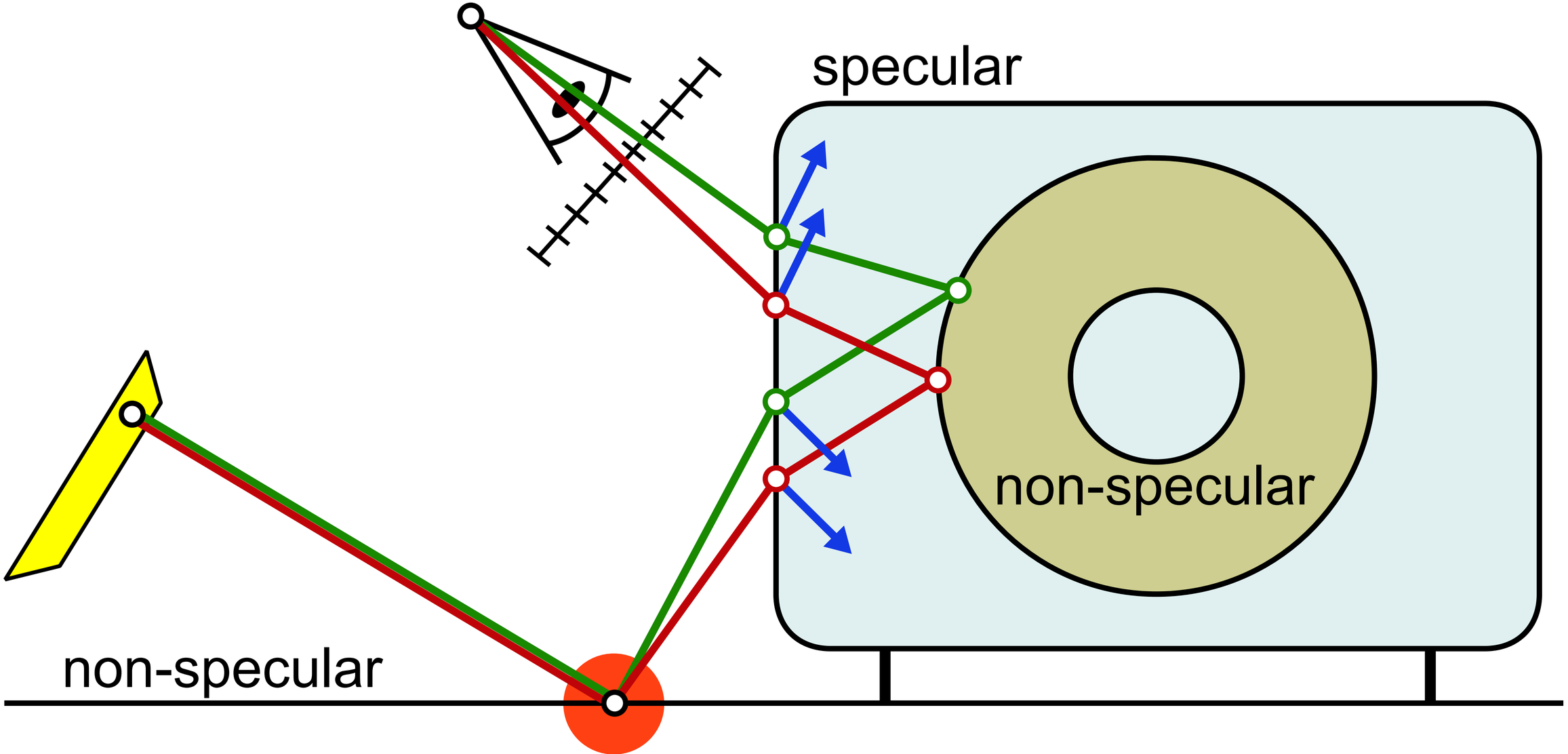


Can’t Connect to Old Secondary Hit

Only one direction for transmitted ray to go!

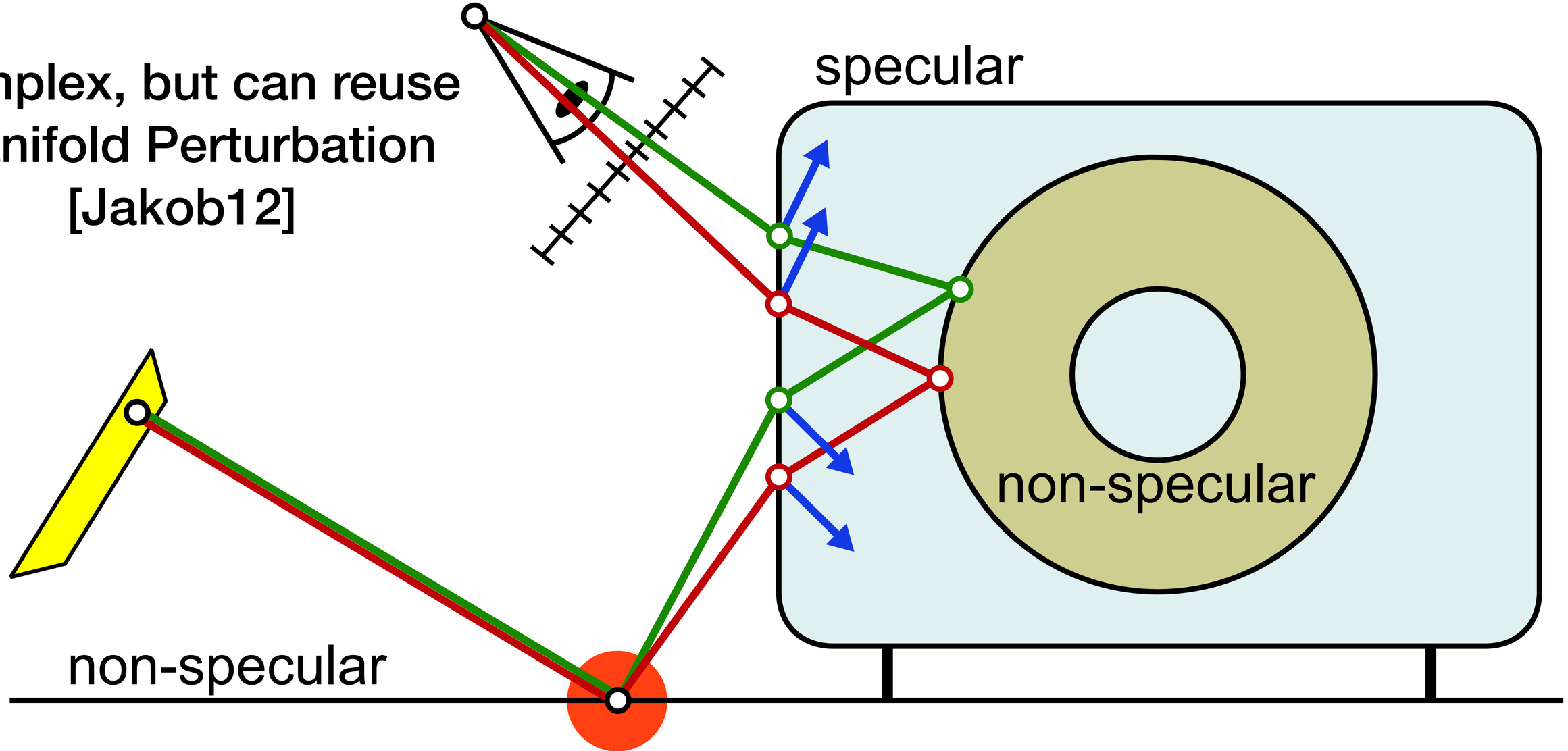


Must Perturb Specular Chain Until

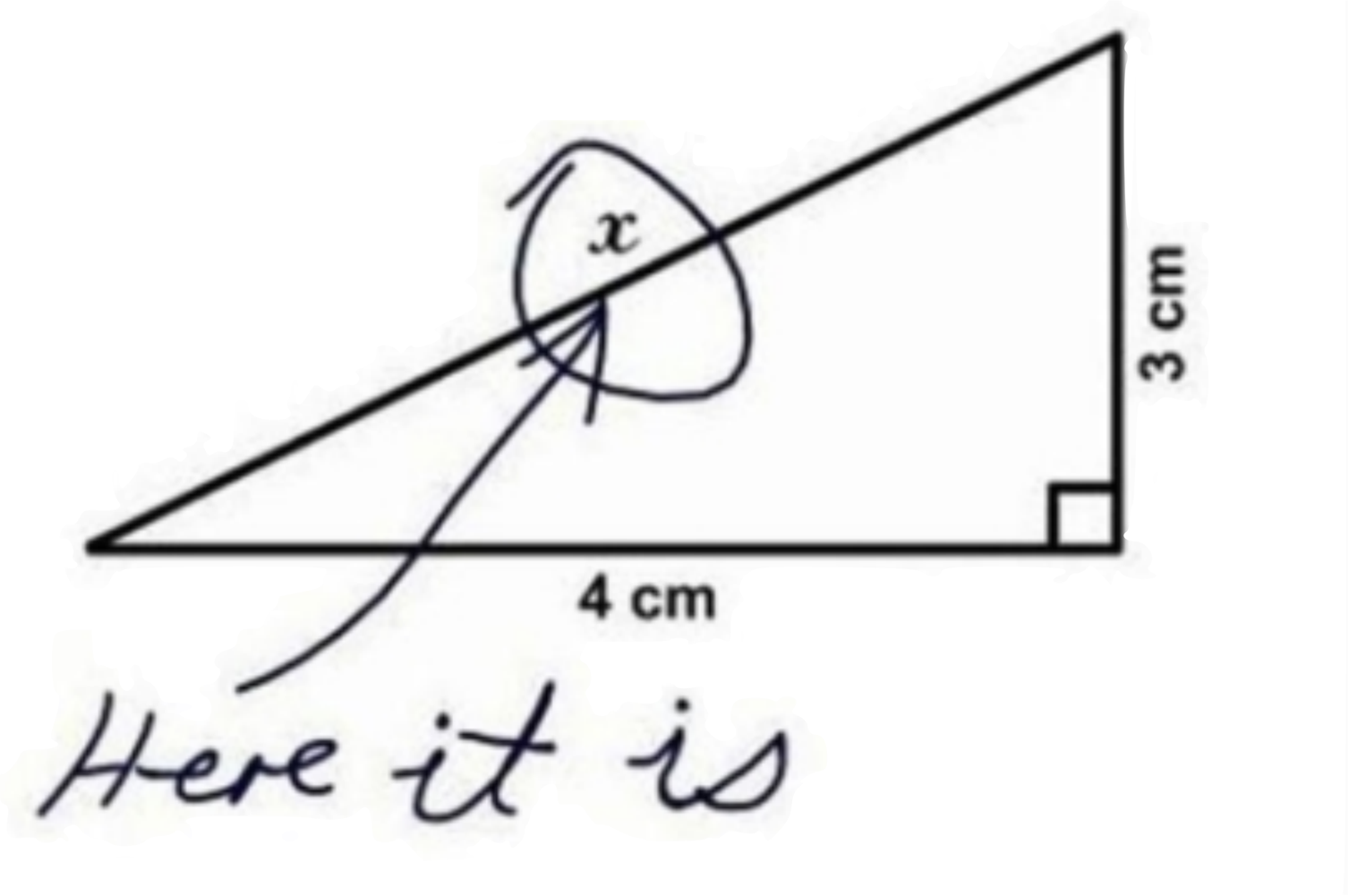


Must Perturb Specular Chain Until

Complex, but can reuse Manifold Perturbation [Jakob12]



Find x



Correct Differences?

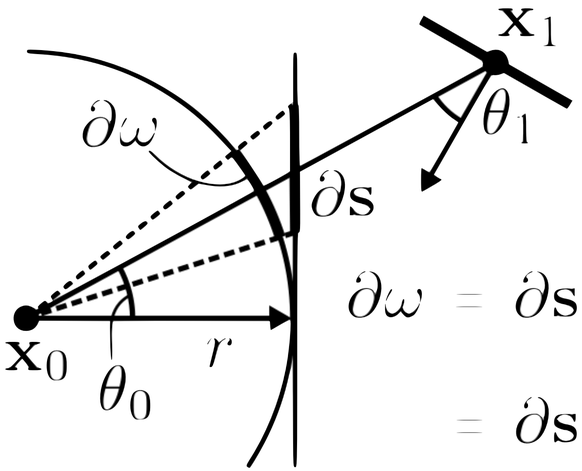
We have to account between different
path space measures between primary and offset
paths by a **path-space Jacobian**

Ground truth | With Jacobian | Without Jacobian

Path-Space Jacobian

Path-Space Jacobian

$$\left|\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_j}\right|_{ij} = \left|\frac{\partial \tilde{\mathbf{x}}_i}{\partial \tilde{\mathbf{O}}_k}\frac{\partial \tilde{\mathbf{O}}_k}{\partial \mathbf{O}_l}\frac{\partial \mathbf{O}_l}{\partial \mathbf{x}_j}\right|_{ij} = \left|\frac{\partial \tilde{\mathbf{x}}_i}{\partial \tilde{\mathbf{O}}_k}\right|_{ik}\left|\frac{\partial \tilde{\mathbf{O}}_k}{\partial \mathbf{O}_l}\right|_{kl}\left|\frac{\partial \mathbf{O}_l}{\partial \mathbf{x}_j}\right|_{lj}$$



$$\partial\omega = \partial\mathbf{s}\frac{\cos\theta_0}{r^2} = \partial\mathbf{s}\cos^3\theta_0$$

$$\frac{\mathrm{d}\mu(T(\bar{x}))}{\mathrm{d}\mu(\bar{x})} = \left|\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_j}\right|_{ij}, \quad i,j = 1,\ldots,c-1.$$

$$\frac{\partial \mathbf{x}_b}{\partial \mathbf{s}} = \frac{\partial \mathbf{x}_b}{\partial \mathbf{x}_1}\frac{\partial \mathbf{x}_1}{\partial \mathbf{s}} = \frac{\partial \mathbf{x}_b}{\partial \omega_0^\perp}\frac{\partial \omega_0^\perp}{\partial \mathbf{x}_1}\frac{\partial \mathbf{x}_1}{\partial \mathbf{s}} = \frac{G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1)}{G(\mathbf{x}_0 \leftrightarrow \ldots \leftrightarrow \mathbf{x}_b)}\frac{\partial \mathbf{x}_1}{\partial \mathbf{s}}$$

$$\left|\frac{\partial \tilde{\mathbf{x}}_i}{\partial \mathbf{x}_j}\right|_{ij} = \left|\frac{\partial \tilde{\mathbf{x}}_i}{\partial \tilde{\mathbf{O}}_k}\right|_{ik}\left|\frac{\partial \tilde{\mathbf{x}}_b}{\partial \mathbf{s}}\frac{\partial \mathbf{s}}{\partial \mathbf{x}_b}\right|\left|\frac{\partial \mathbf{O}_l}{\partial \mathbf{x}_j}\right|_{lj}$$

$$= \left(\left|\frac{\partial \tilde{\mathbf{x}}_b}{\partial \mathbf{s}}\right|\left|\frac{\partial \tilde{\mathbf{x}}_i}{\partial \tilde{\mathbf{O}}_k}\right|_{ik}\right)\left(\left|\frac{\partial \mathbf{x}_b}{\partial \mathbf{s}}\right|\left|\frac{\partial \mathbf{x}_j}{\partial \mathbf{O}_l}\right|_{jl}\right)^{-1}$$

$$\frac{\partial \tilde{\mathbf{x}}_1}{\partial \mathbf{s}}\left(\frac{\partial \mathbf{x}_1}{\partial \mathbf{s}}\right)^{-1} = \frac{\|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_0\|^2 \cos\theta_1 \cos^3\tilde{\theta}_0}{\|\mathbf{x}_1 - \mathbf{x}_0\|^2 \cos\tilde{\theta}_1 \cos^3\theta_0}$$

$$\frac{\partial \mathbf{x}_1}{\partial \mathbf{s}} = \frac{\|\mathbf{x}_1 - \mathbf{x}_0\|^2 \cos^3\theta_0}{\cos\theta_1}$$

Does It Work?

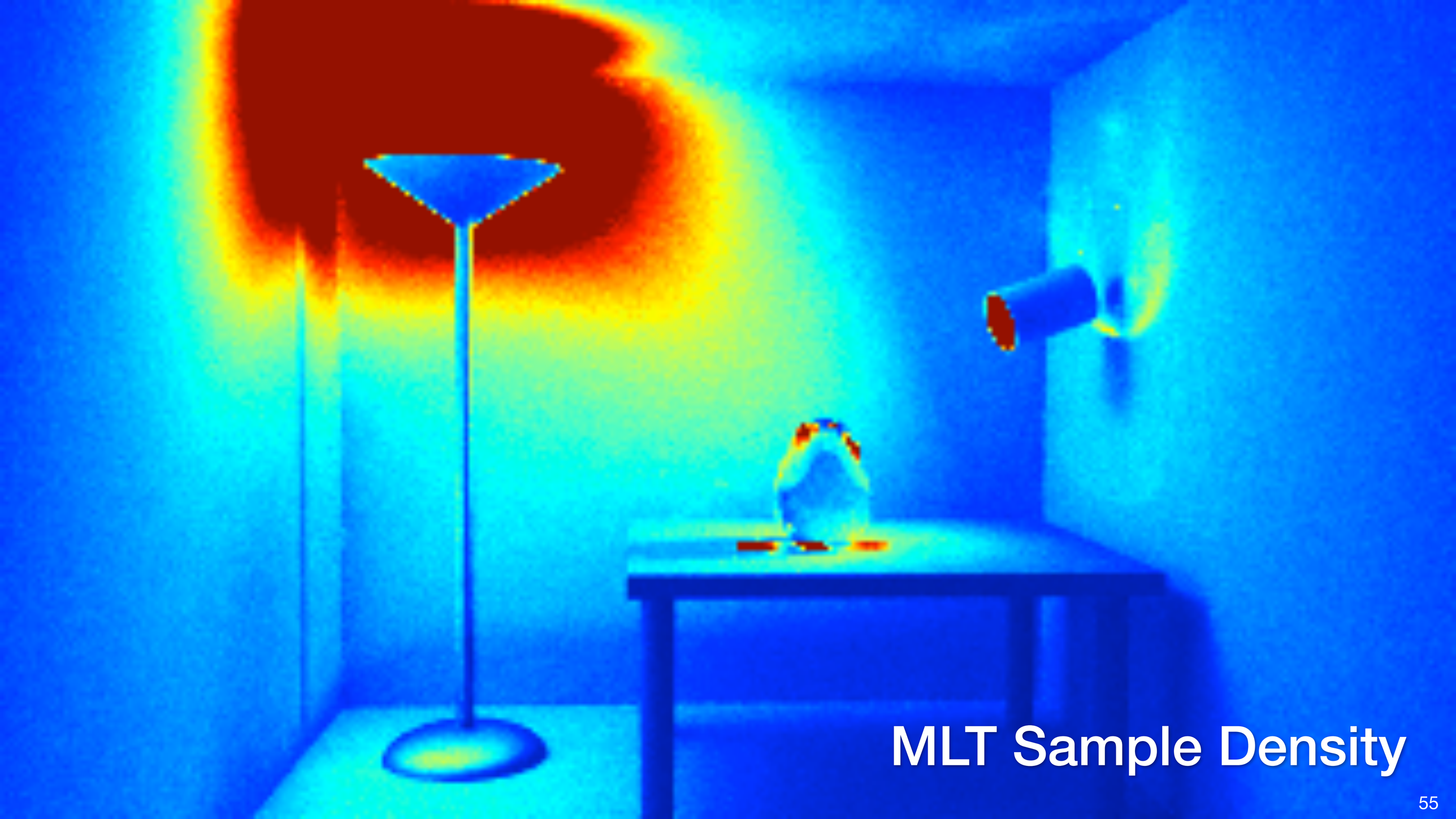
Implementation in Mitsuba [Jakob]

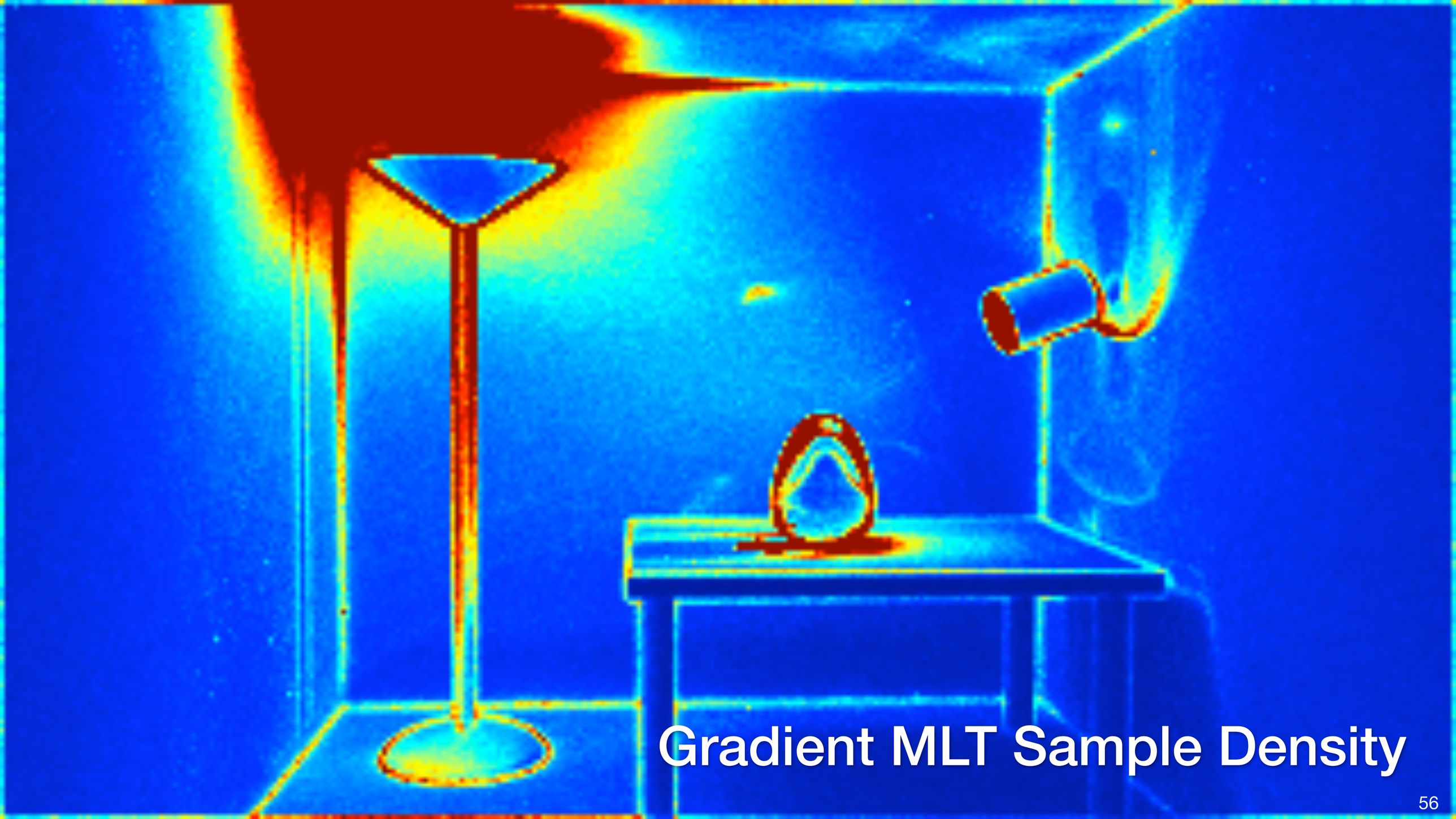
Comparison to previous unbiased algorithms

— Veach MLT, Kelemen MLT, ERPT, Bidirectional Path Tracing

Showing uncoverged images to highlight differences


Indirect Only


MLT Sample Density


Gradient MLT Sample Density

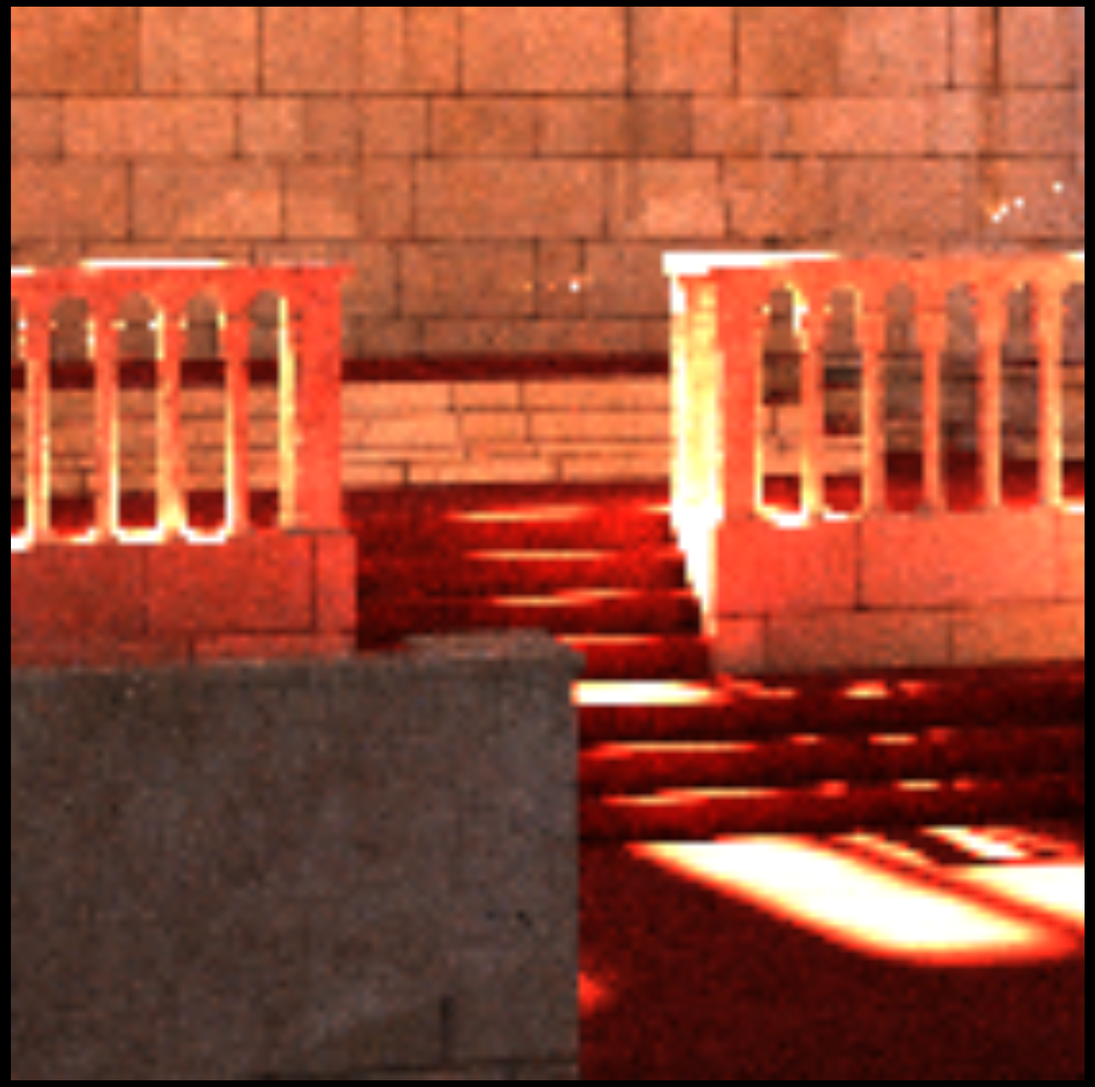
Sample Density

Path space gradients are **not the same as image gradients**

— Multiple positive and negative gradients within pixel can cancel out

Still, **sample density higher on edges**


Difficult transport through windows



Veach MLT (32.4 dB)

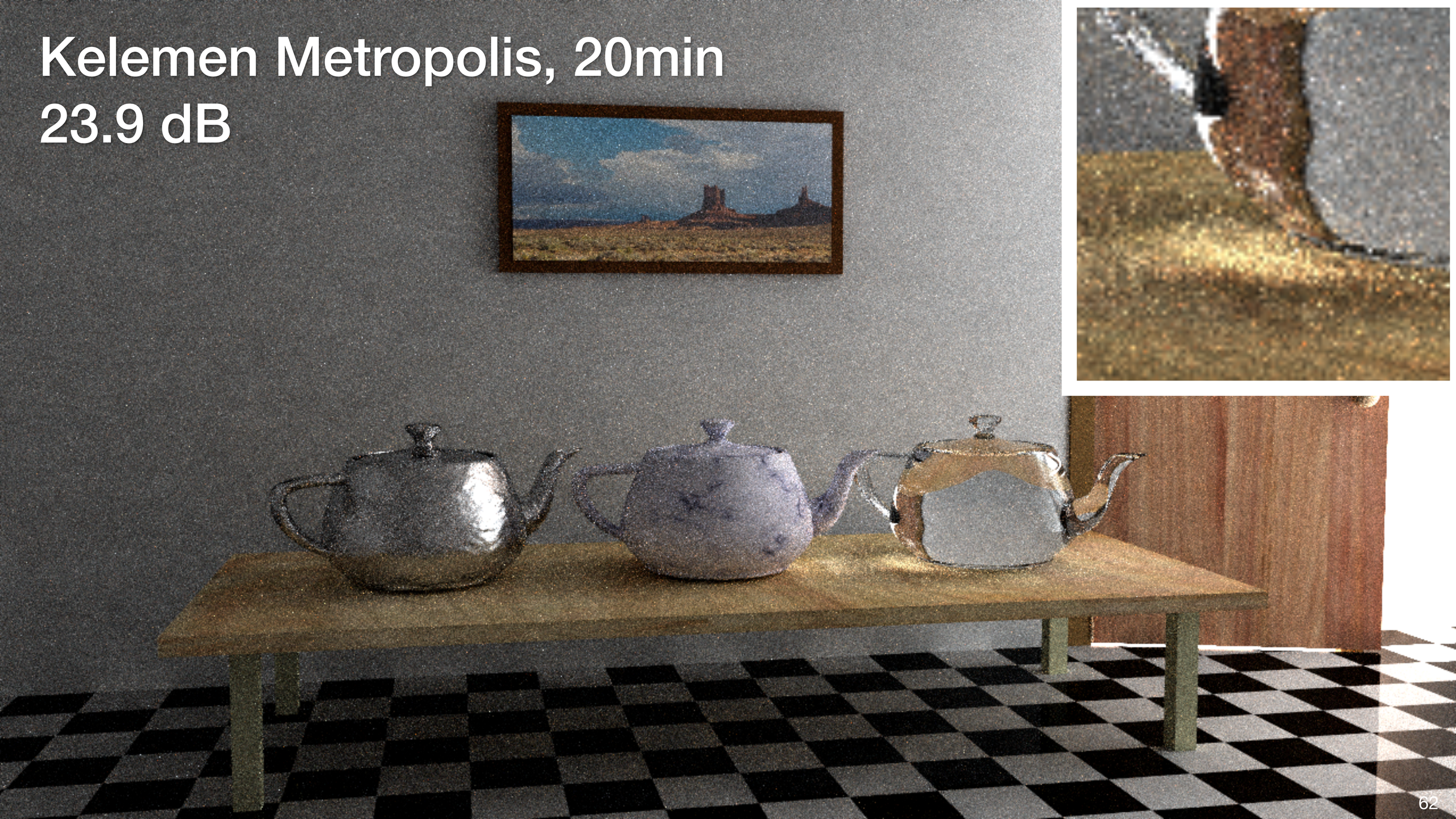
Our result (34.2 dB)

Highly Complex Indirect Transport

Aittala feat. Laine, after Veach

Bidirectional Path Tracing, 20min
22.3 dB

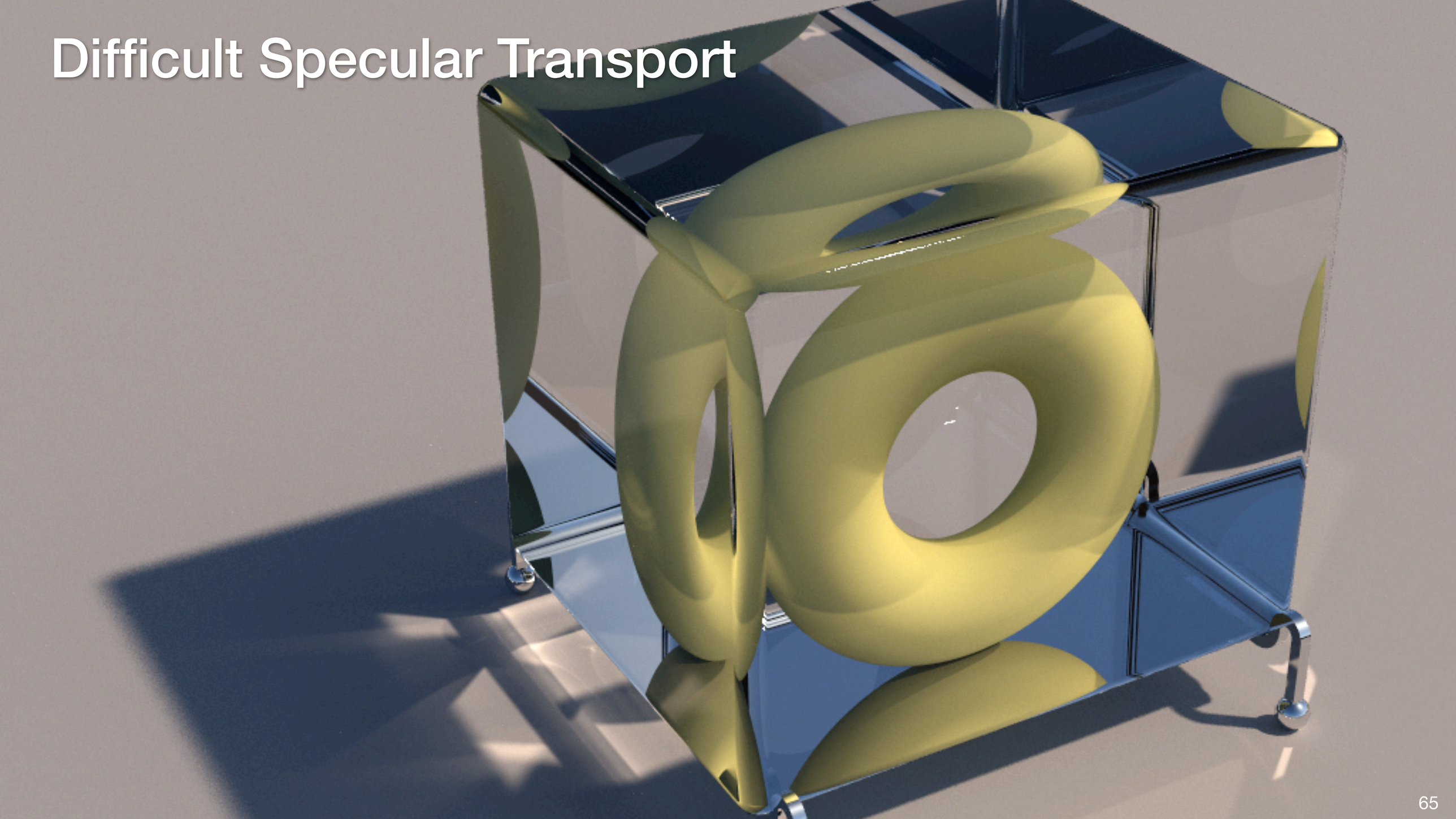
Kelemen Metropolis, 20min
23.9 dB



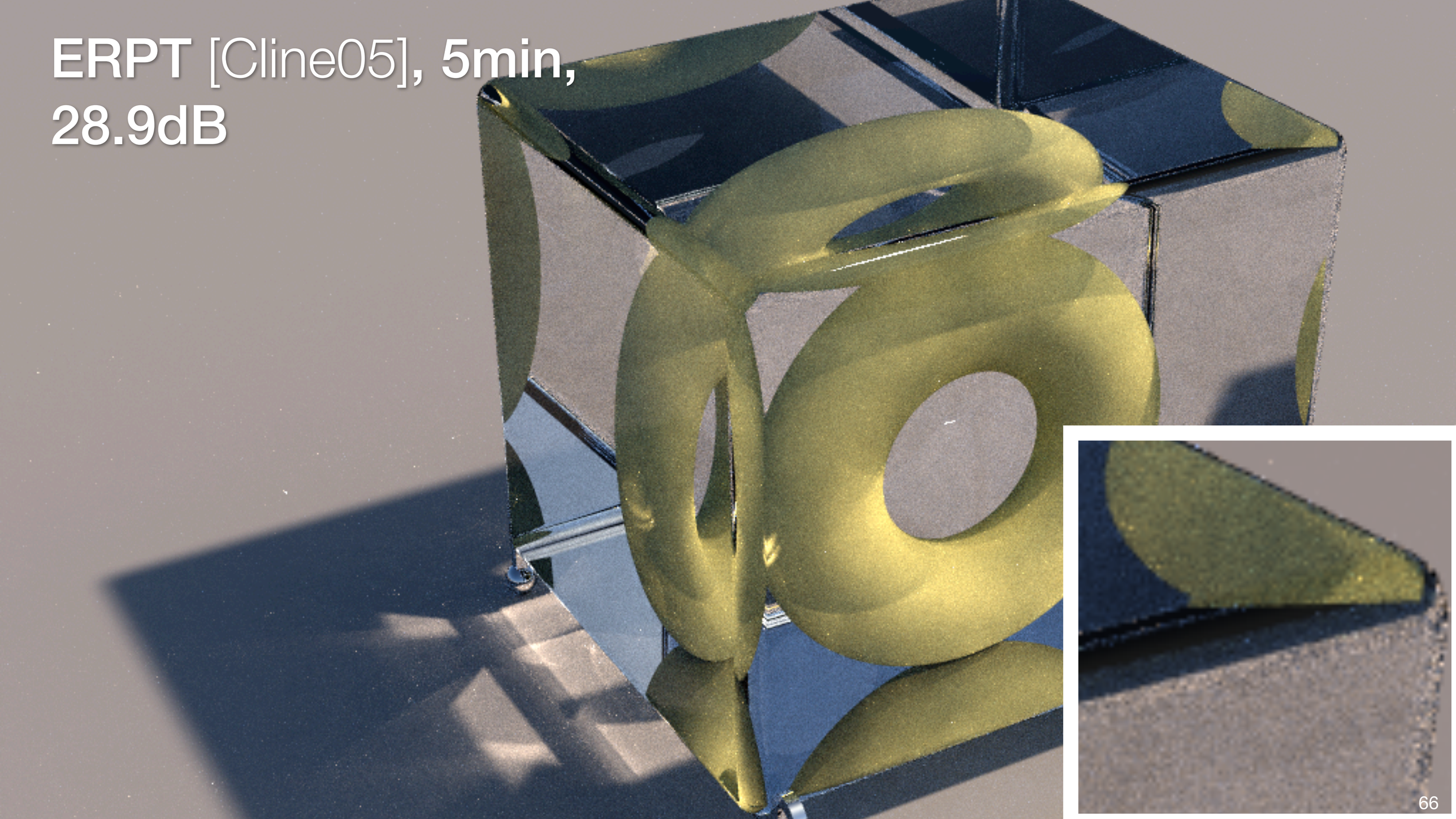
Veach Metropolis, 20min
34.2 dB

Gradient MLT (our), 20min
37.8 dB

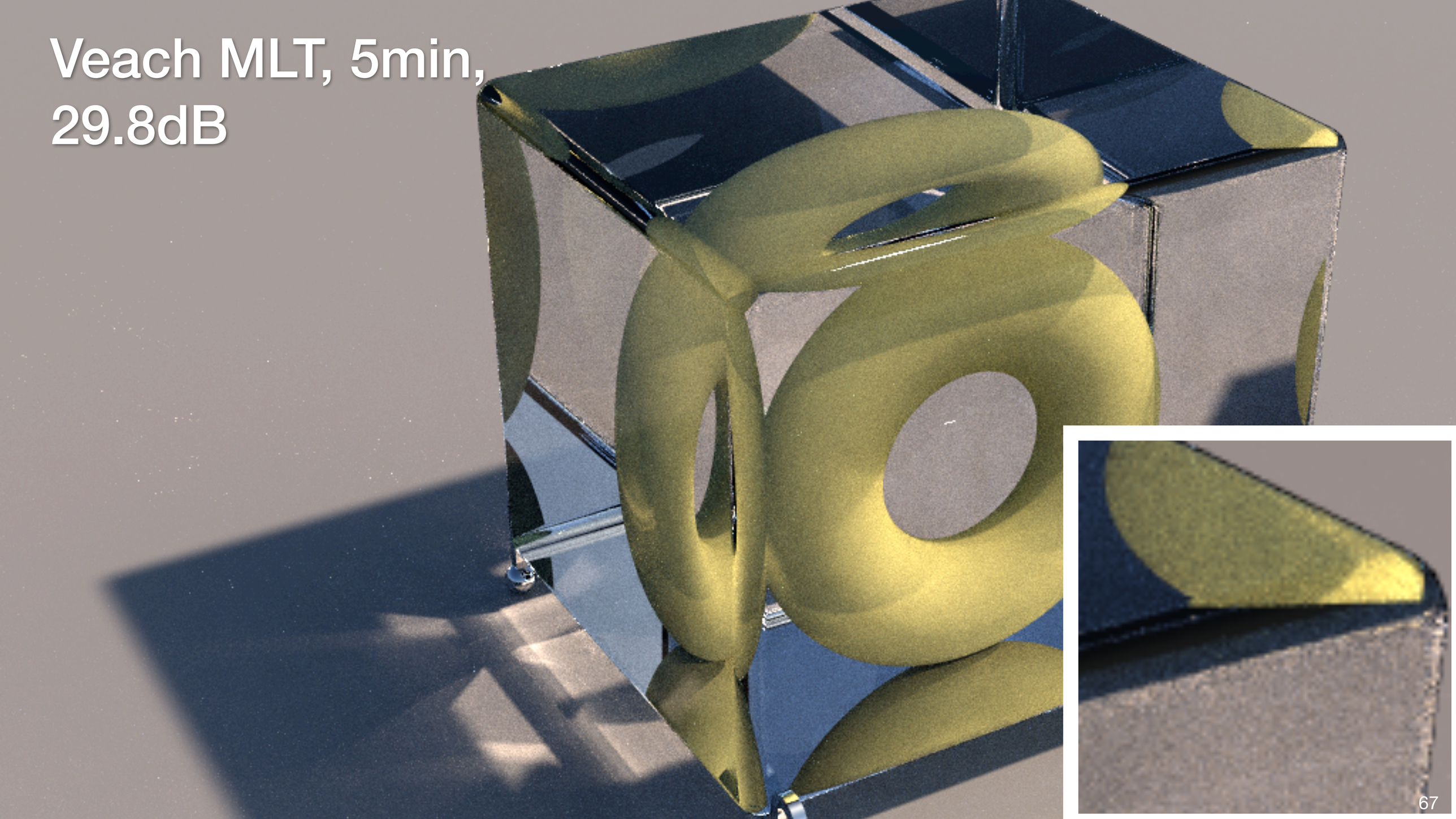
Difficult Specular Transport



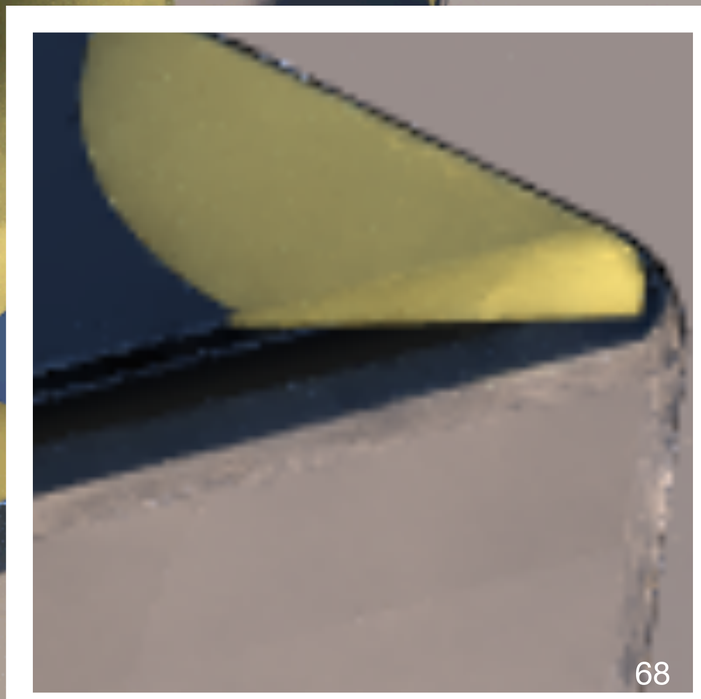
ERPT [Cline05]**, 5min, 28.9dB**



Veach MLT, 5min, 29.8dB



Gradient MLT (our), 5min, 29.8dB



Gradient MLT (our), 5min, 29.8dB

Gradient MLT works also for very difficult transport

— No large differences between Metropolis algorithms

— Gradient MLT artifacts have a different visual characteristic

Why is Gradient Sampling Beneficial?

Gradients contain **way less energy than actual image**

— I.e., more succinct representation

— But they are **harder to sample**, partially canceling benefit

Still **net win** in our test cases

Screened Poisson Analysis

Image is reconstructed from gradients and a coarse primal image

— L2 reconstruction is unbiased

Fourier analysis shows the **solver takes low frequencies from primal image, high frequencies from integrated gradients**

— Cutoff frequency controlled by weighting

Explicit details in paper

Screened Poisson Solver, high gradient weight

Result

=

Low freq. (primal)

+

High freq. (gradients)

Screened Poisson Solver, small gradient weight

Result

=

Low freq. (primal)

+

High freq. (gradients)

Limitations and Future Work

MLT algorithms have trouble finding all light transport “modes”

— Results in uneven, “clunky” converge

— Our algorithm shares this problem

— Can we do something about it?

We use only mutators designed for regular MLT

— Interesting to see if we can specialize for gradients

Conclusion

Fundamentally novel unbiased sampling for light transport?

— Drives computation towards “where things happen”

Conclusion

Fundamentally novel unbiased sampling for light transport?

— Drives computation towards “where things happen”

Interesting connections to adaptive sampling,
such as **MDAS** [Hachisuka08] and others

— Both algorithms concentrate samples on boundaries in path space

Thank You!

Acknowledgments

Wenzel Jakob for Mitsuba

HeCSE Graduate School, MIDE/UI-ART/Aalto University

NSF CGV 1116303

David Luebke

Anonymous reviewers for constructive and thorough feedback