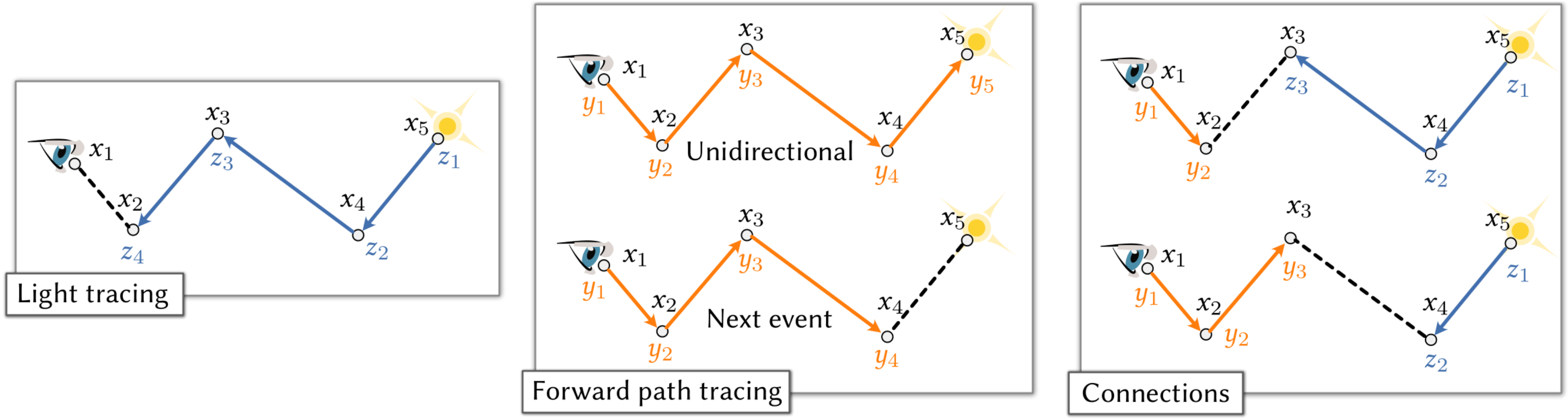


Bidirectional path tracing

It‘s difficult to find small lights or the camera. So why not start from both sides?

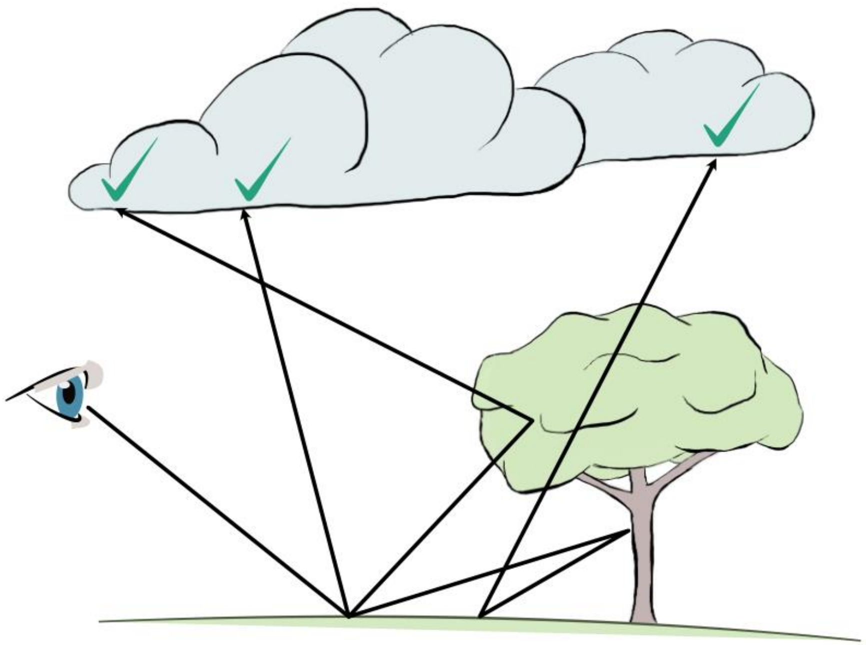
Pascal Grittmann, Corentin Salaün

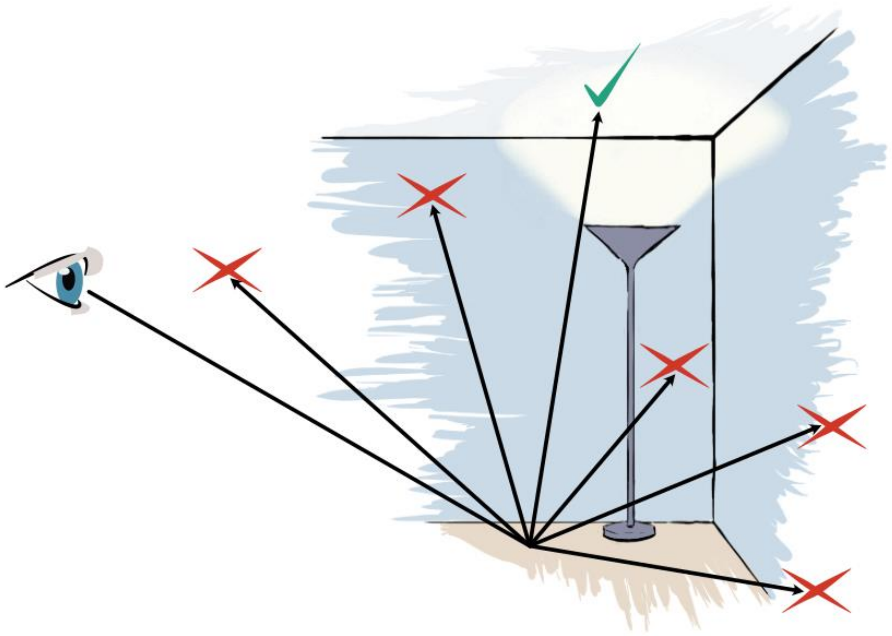
Motivation

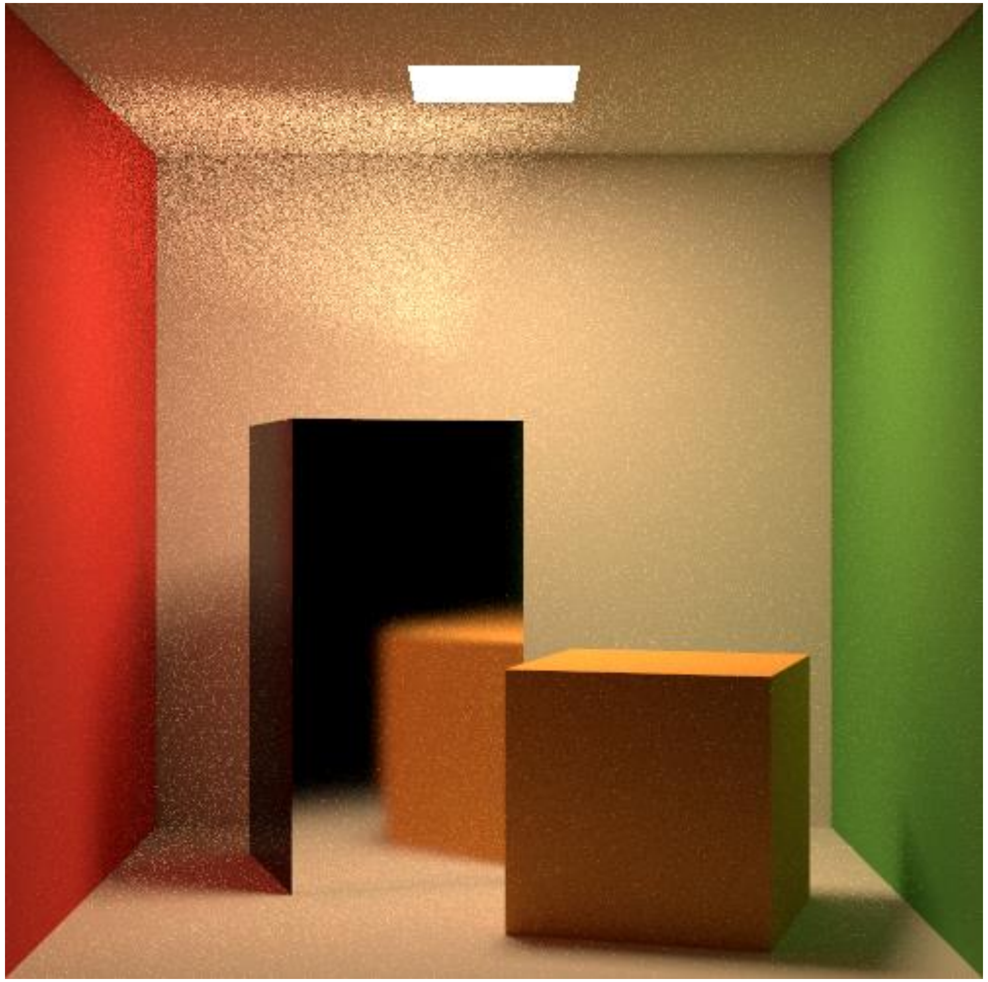
Why is it necessary / beneficial to trace bidirectionally?

Forward path tracing

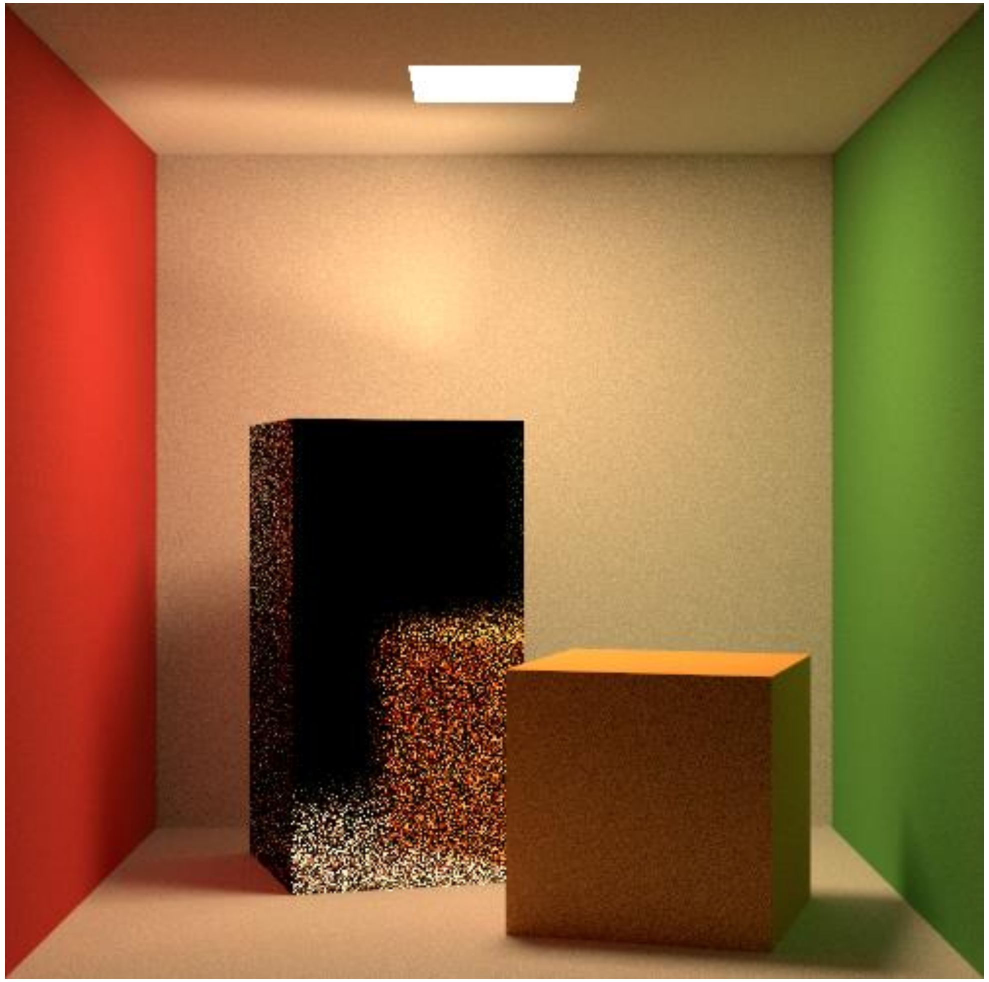
- Efficient at uniform / diffuse illumination and large scenes
- Very bad at focused indirect illumination







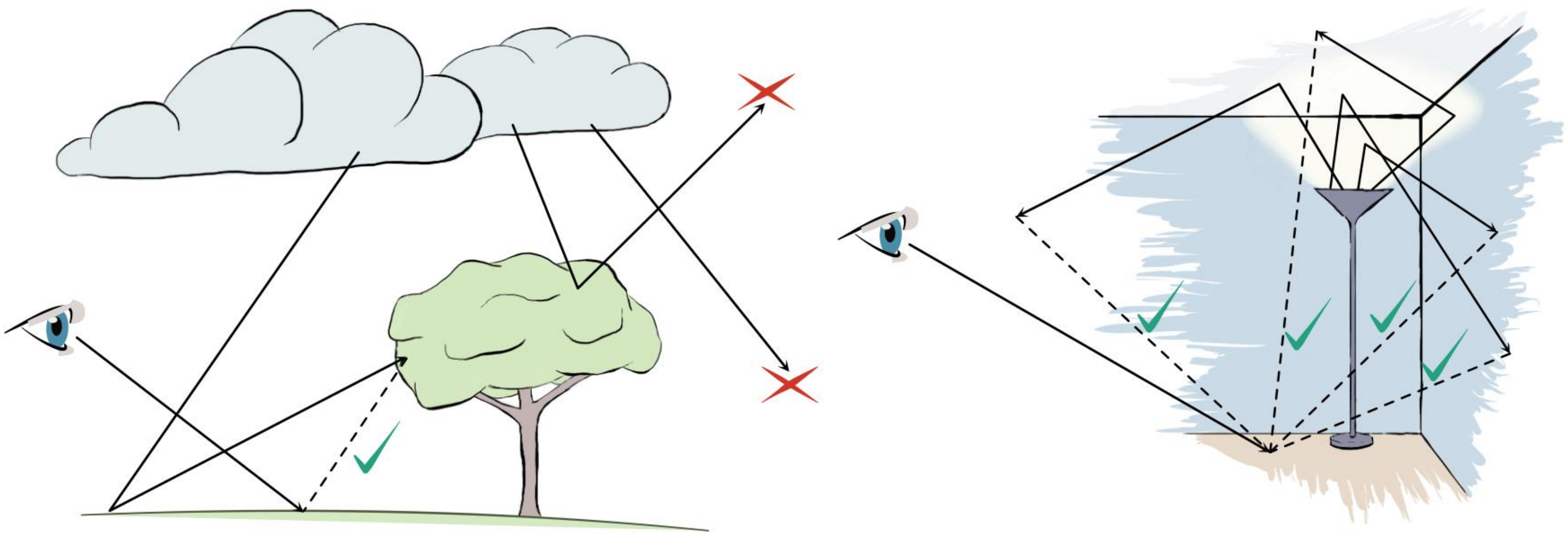
Path Tracing



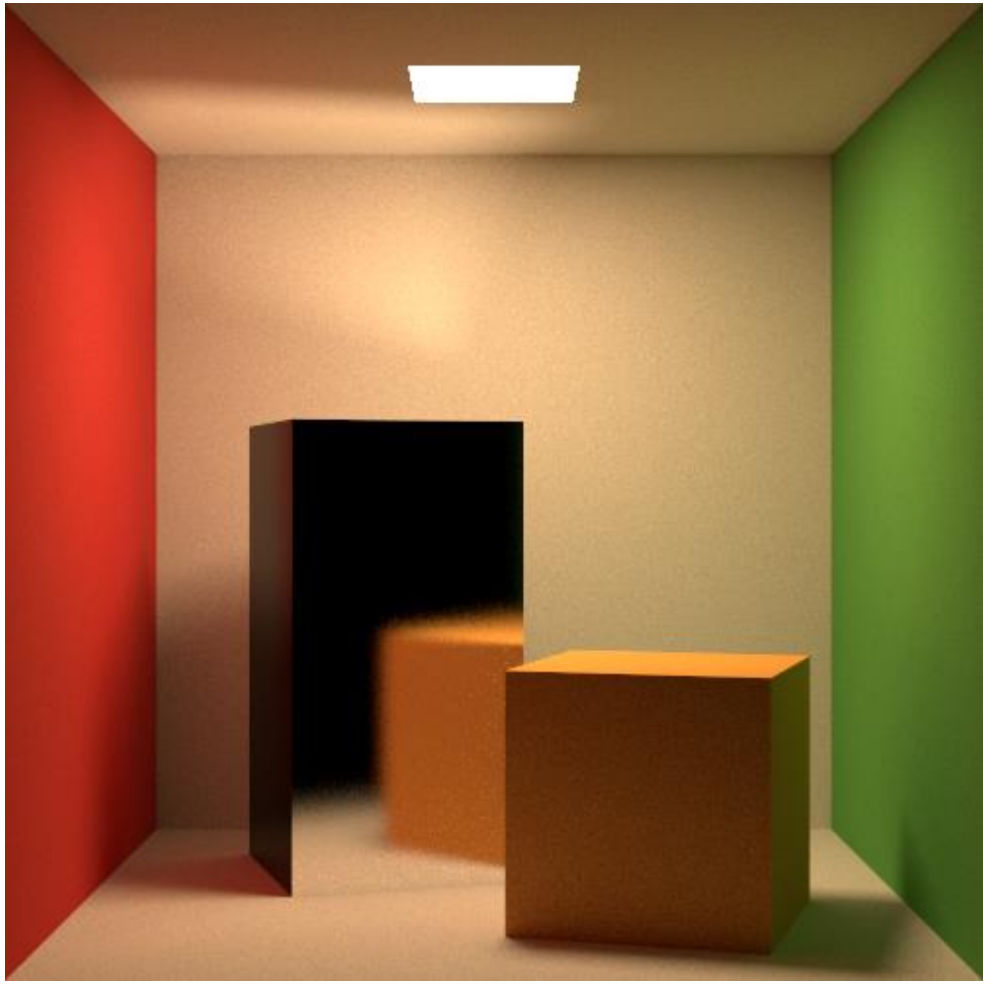
Light Tracing

Bidirectional path tracing

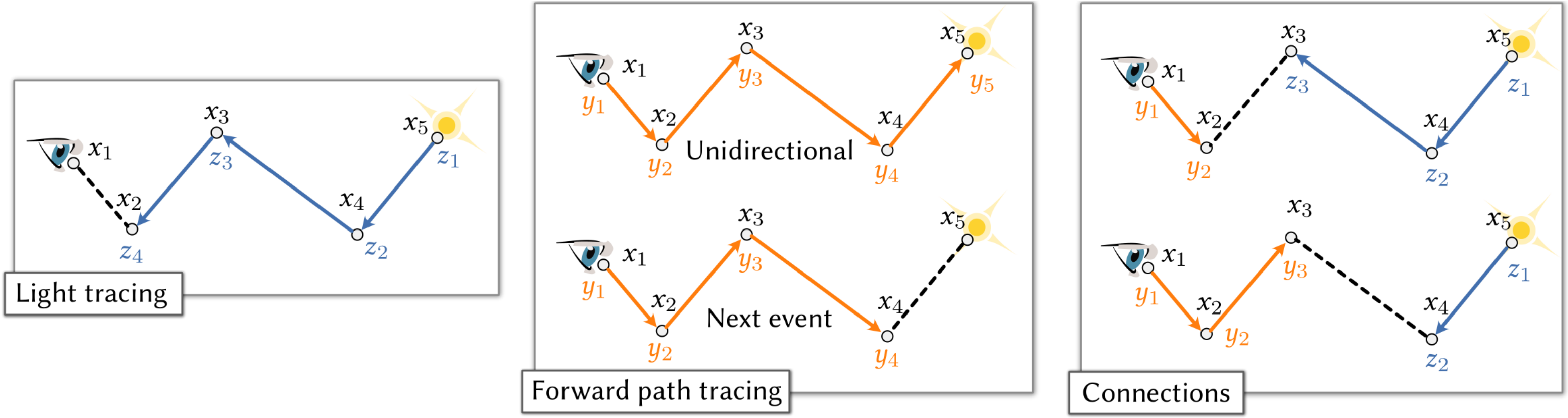
- Exceptional performance on focused indirect illumination
- But: Not great for large scenes



Bidirectional path tracing



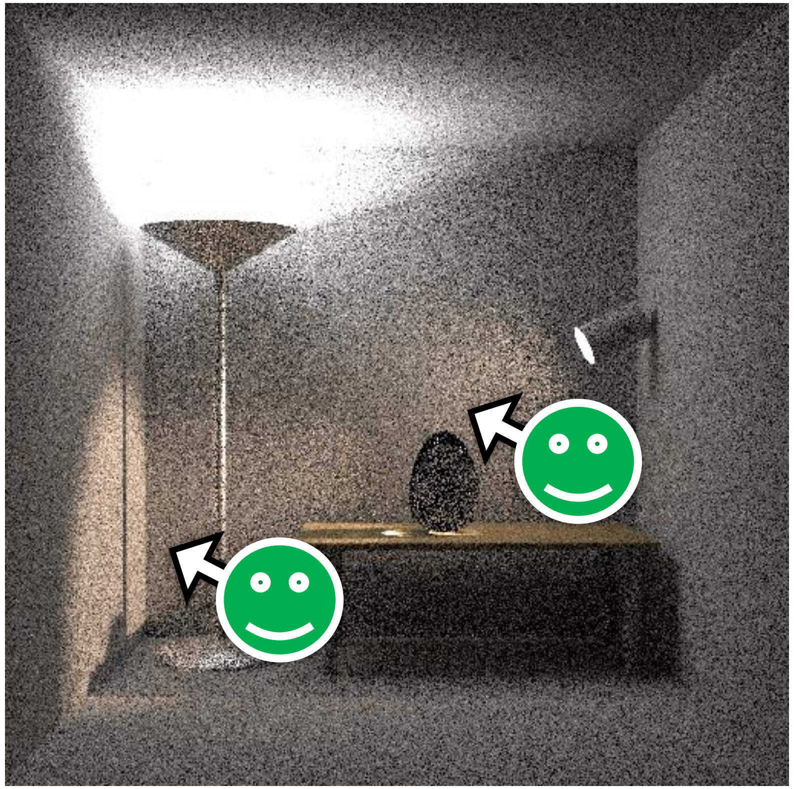
Bidirectional sampling techniques

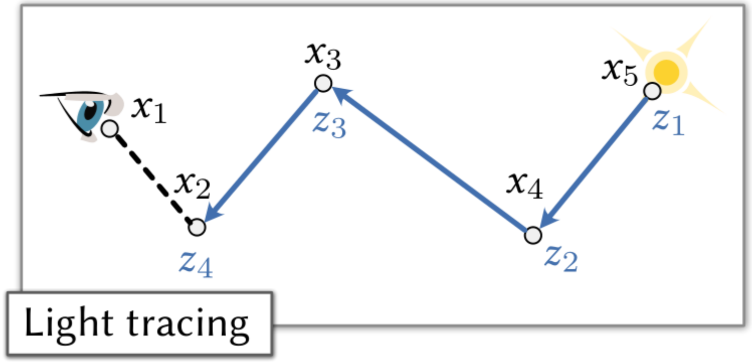


Lafortune et al., Bidirectional Path Tracing, [CompuGraphics`93]
Veach & Guibas, Bidirectional Estimators for Light Transport, [EGRW´94, Siggraph´95]

For (almost) every effect there is a well-suited technique

So, bidirectional path tracing combines them all



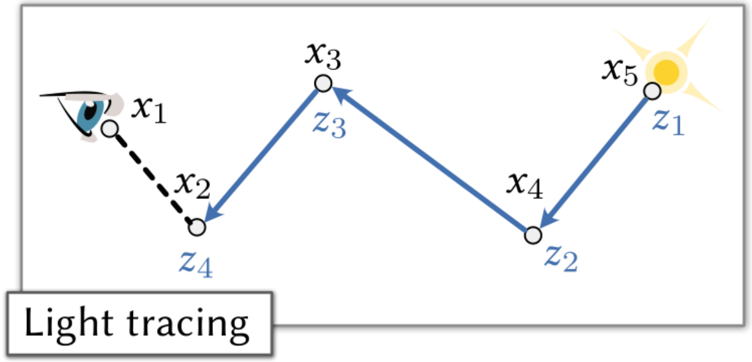


Light tracing

The simplest bidirectional technique

What do we compute?

- Let's assume we sampled a path from the light source into the scene
- What do we do with it? How do we get a (correct) pixel value estimate from it?



vs

$$L_o = L_e + \int_{\Omega} L_i \, f \cos\theta_i \, d\omega_i$$

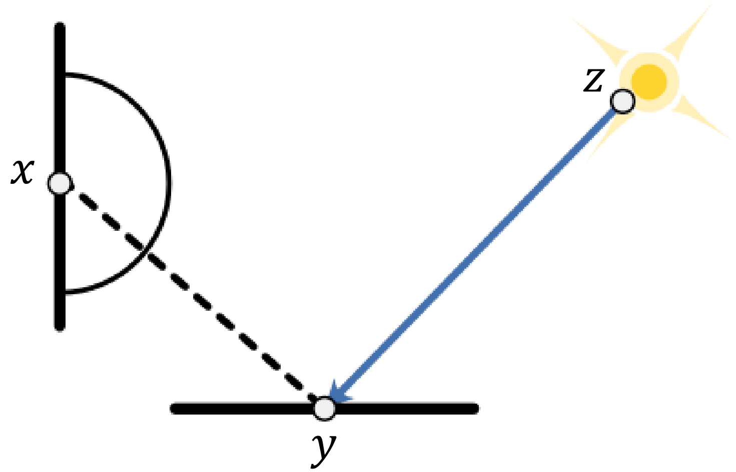
Starting simple: 2-bounce irradiance

- Irradiance at a point x in the scene:

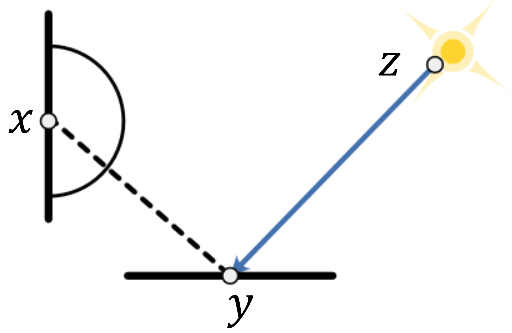
$$E(x) = \int_{\Omega} L_i(x, \omega_x) \cos\theta_x \, d\omega_x$$

- For paths of length 2:

$$E(x) = \int_{\Omega} \left(\int_{\Omega} L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, d\omega_y \right) \cos\theta_x \, d\omega_x$$



First, write it as a surface integral

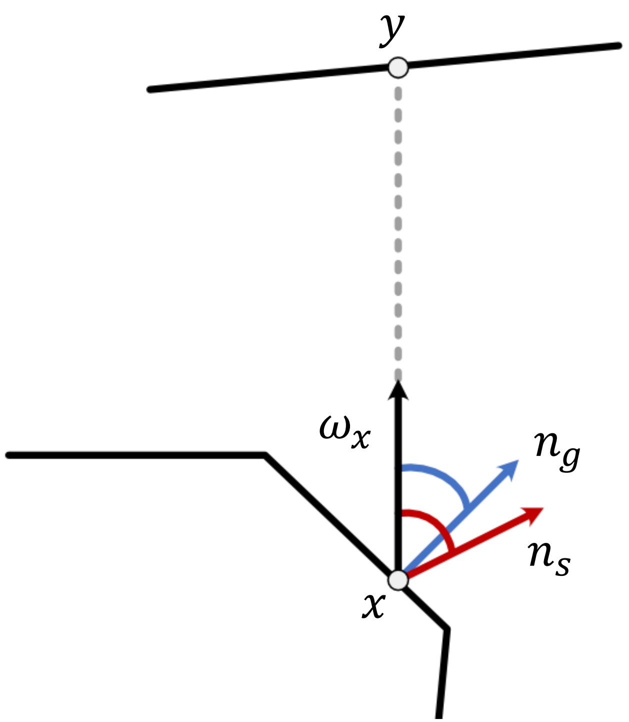


$$E(x) = \int_\Omega \left(\int_\Omega L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, d\omega_y \right) \cos\theta_x \, d\omega_x$$

$$E(x) = \int_A \left(\int_A L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, V(y,z) \frac{\cos\theta(z \to y)}{\|z - y\|^2} dz \right) \cos\theta_x \, V(x,y) \frac{\cos\theta(y \to x)}{\|y - x\|^2} dy$$

Shading cosines and Jacobian cosines

- Note the distinction between the rendering equation cosines
  - $\cos\theta_x$ - w.r.t. the *shading normal* n_s
- And the geometry cosines from the Jacobians
  - $\cos\theta(x \to y)$ - w.r.t. the *actual surface normal* n_g



$$E(x) = \int_A \left(\int_A L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, V(y,z) \frac{\cos\theta(z \to y)}{\|z - y\|^2} dz \right) \cos\theta_x \, V(x,y) \frac{\cos\theta(y \to x)}{\|y - x\|^2} dy$$

Reverse order of integrals

- Fubini's theorem:
  - Our integrand is positive, integral value is finite ⇒ integration order arbitrary

$$E(x) = \int_A \left(\int_A L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, V(y,z) \frac{\cos\theta(z \to y)}{\|z-y\|^2} dz \right) \cos\theta_x \, V(x,y) \frac{\cos\theta(y \to x)}{\|y-x\|^2} dy$$

$$\Leftrightarrow \quad E(x) = \int_A \int_A L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, V(y,z) \frac{\cos\theta(z \to y)}{\|z-y\|^2} \cos\theta_x \, V(x,y) \frac{\cos\theta(y \to x)}{\|y-x\|^2} dy \, dz$$

Integral formulation for light tracing

$$E(x) = \int_A \int_A L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, V(y,z) \frac{\cos\theta(z \to y)}{\|z-y\|^2} \cos\theta_x \, V(x,y) \frac{\cos\theta(y \to x)}{\|y-x\|^2} \boxed{dy} \, dz$$

- Back to the hemisphere, but at the other end

$$E(x) = \int_A \int_\Omega L_e(z, \omega_y) f(y, \omega_x, \omega_y) \cos\theta_y \, \cancel{V(y,z)} \frac{\cos\theta(z \to y)}{\cancel{\|z-y\|^2}} \cos\theta_x \, V(x,y) \frac{\cos\theta(y \to x)}{\|y-x\|^2} \boxed{\frac{\cancel{\|z-y\|^2}}{\cos\theta(y \to z)} d\overleftarrow{\omega}_z} \, dz$$

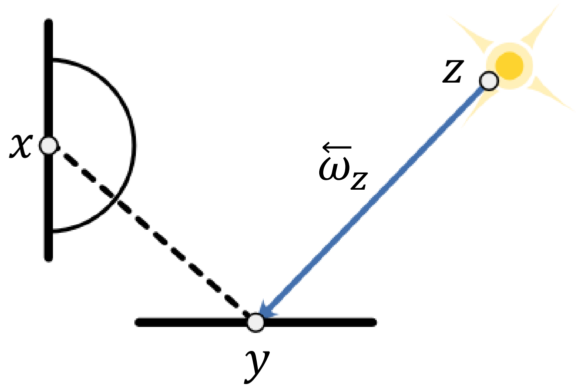
- Simplified & reordered

$$E(x) = \int_A \int_\Omega L_e(z, \omega_y) \cos\theta(z \to y) \; f(y, \omega_x, \omega_y) \frac{\cos\theta_y}{\cos\theta(y \to z)} \cos\theta(y \to x) \, V(x,y) \frac{\cos\theta_x}{\|y-x\|^2} d\overleftarrow{\omega}_z \, dz$$

Understanding the result

- Sample z on light and direction $\overleftarrow{\omega}_z = -\omega_y$, evaluate emission and Jacobian
- Trace ray to find y, evaluate BSDF, correct discrepancy between shading normal and geometry
- Connect to x (deterministic), evaluate visibility, geometry term

$$E(x) = \int_A \int_\Omega L_e(z, \omega_y) \cos\theta(z \to y)\ f(y, \omega_x, \omega_y) \frac{\cos\theta_y}{\cos\theta(y \to z)} \cos\theta(y \to x)\, V(x, y) \frac{\cos\theta_x}{\|y - x\|^2}\, d\overleftarrow{\omega}_z\, dz$$

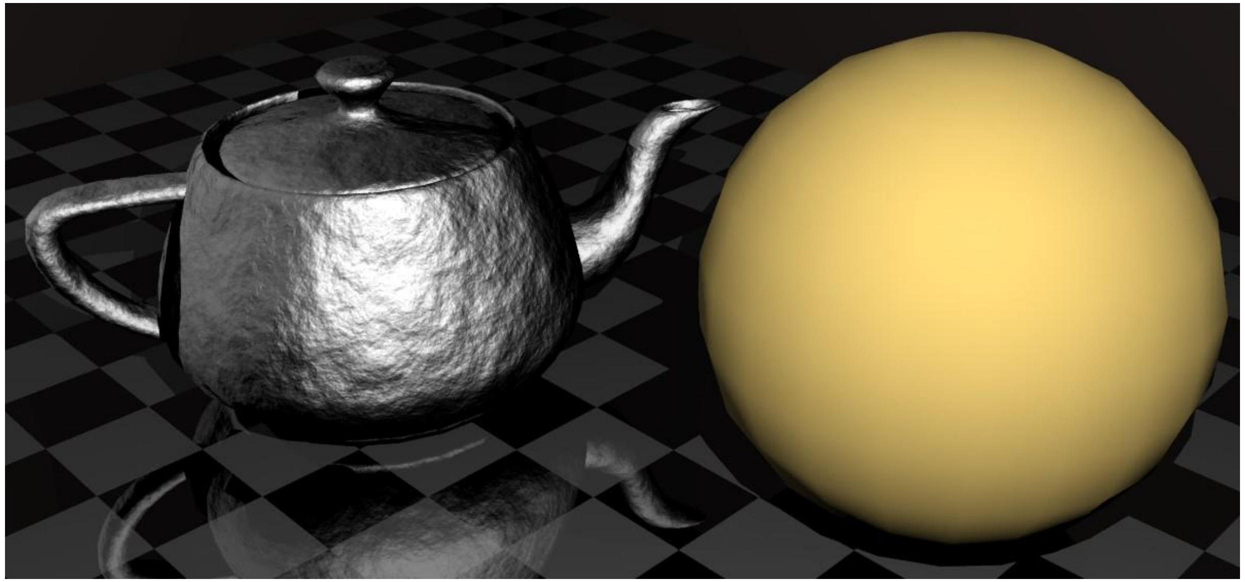


Including the correct cosines

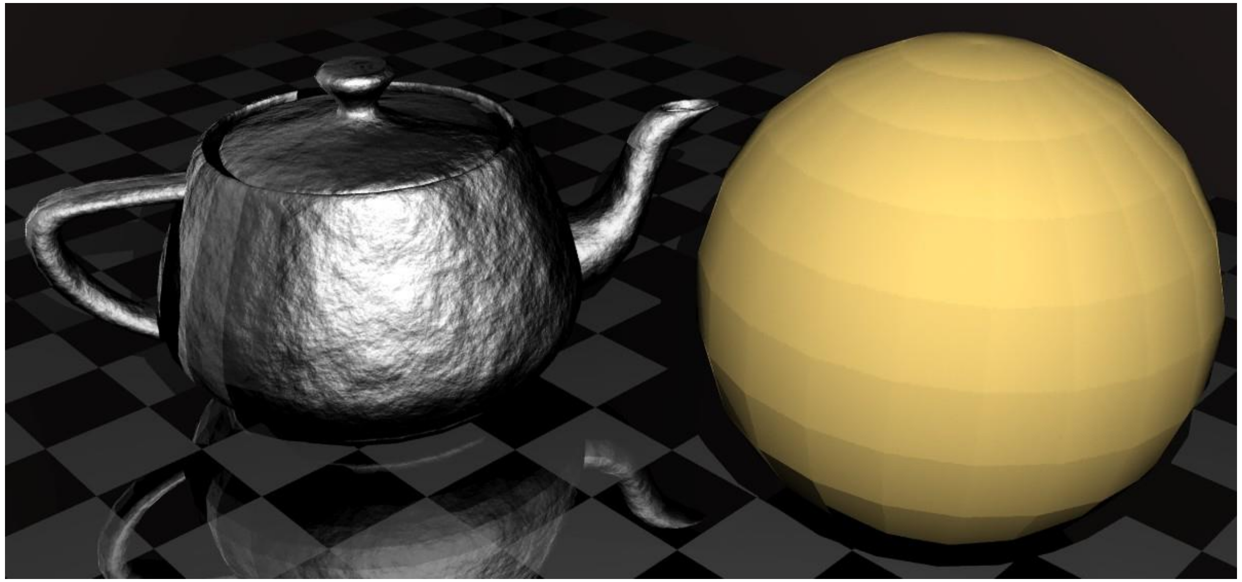
- If the shading normal and geometry normal align, some cosines cancel out

$$E(x) = \int_A \int_\Omega L_e(z, \omega_y) \cos\theta(z \to y)\ f(y, \omega_x, \omega_y) \cancel{\frac{\cos\theta_y}{\cos\theta(y \to z)}} \cos\theta(y \to x)\, V(x, y) \frac{\cos\theta_x}{\|y - x\|^2} d\overleftarrow{\omega}_z\, dz$$

- But careful! Doesn't hold with normal mapping, smooth shading, …



Correct result

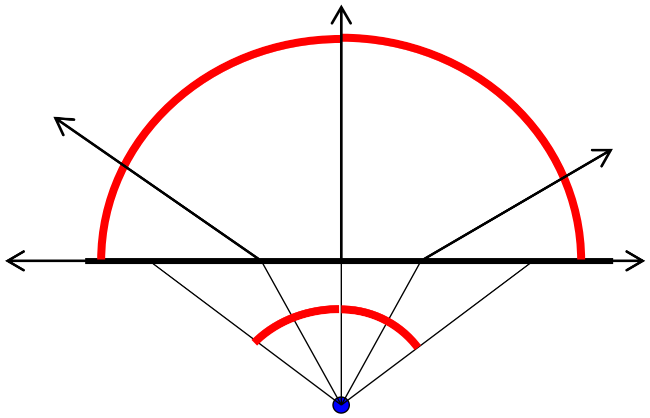


Erroneously canceled cosine terms

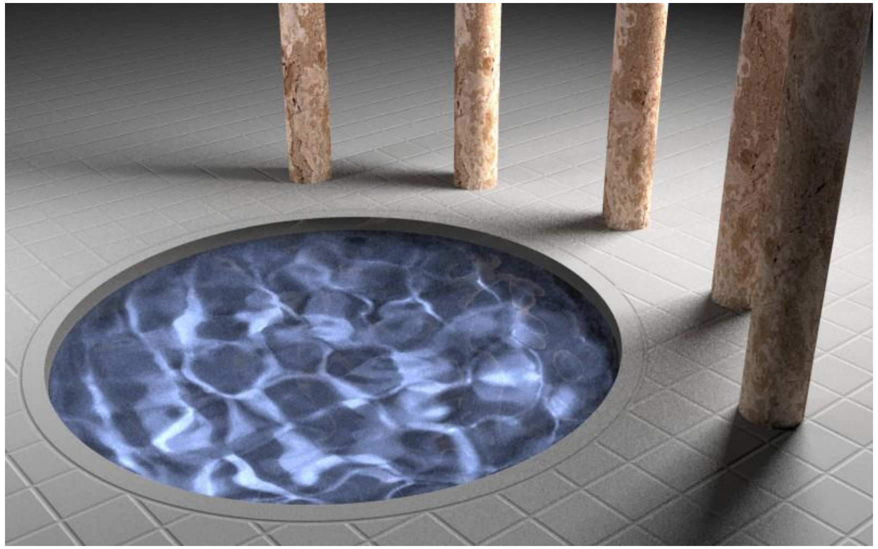
Non-symmetric BSDFs

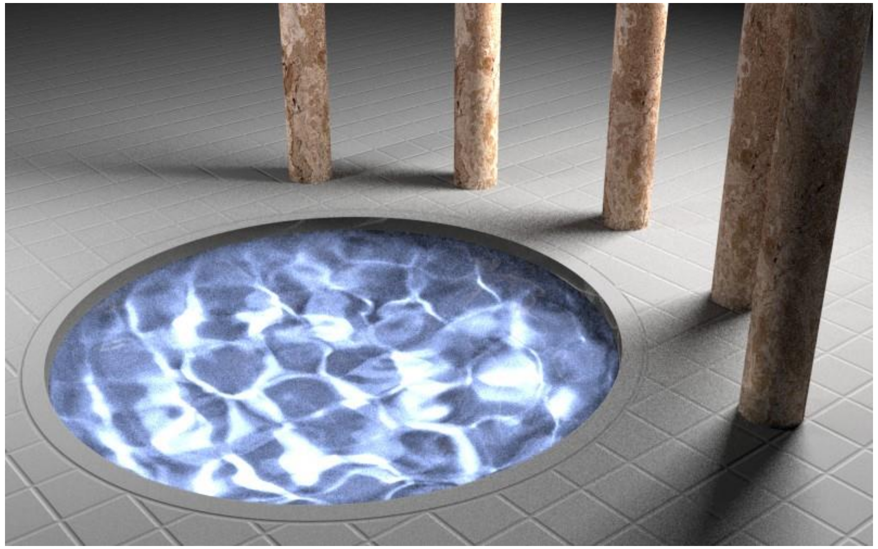
- Refraction is not symmetric
- Must be careful to compute the correct terms!
- See Veach 1997, Chapter 5

$$L_t = \frac{\eta_t^2}{\eta_i^2} L_i$$



Correct image





Erroneously assuming symmetric BSDF

Extension to paths of length k

- Same steps, just more integrals $\int_A \int_\Omega \dots \int_\Omega f(x_1 \dots x_k)\, d\overleftarrow{\omega}_2 \dots d\overleftarrow{\omega}_k\, dx_k$

$$f(x_1 \dots x_k) = \underbrace{L_e \cos\theta(x_k \to x_{k-1})}\left(\prod_{i=2}^{k-1} \underbrace{f(x_{i+1}, x_i, x_{i-1}) \frac{\cos\theta_{x_i}}{\cos\theta(x_i \to x_{i+1})} \cos\theta(x_i \to x_{i-1})}\right) \underbrace{V(x_1, x_2) \frac{\cos\theta_{x_1}}{\|x_2 - x_1\|^2}}$$

Emission and cosine (Jacobian)
After sampling point + dir on light

BSDF, cosine, geometry cosines (Jacobian)
After sampling dir at next hit point

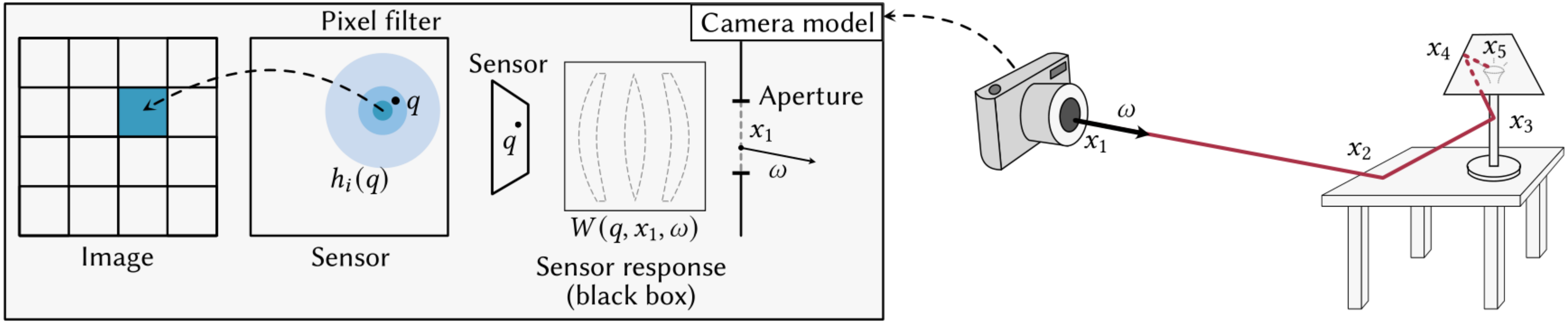
Visibility, cosine, and squared distance (from Jacobian) at shading point
After connecting x_2 to x_1

The camera model

- We’re not computing irradiance at a point, but a pixel value

$$I_i = \int_S \int_L h_i(q)\, W(q, x_1, \omega)\, L_i(x_1, \omega)\, dx_1 dq$$

Sensor — Aperture — Direction (usually implicitly defined by combination of q and x_1)



Surface integral for the camera model

- With light tracing, we sample the sensor point implicitly:
  - We “arrived” at x_2 in the scene
  - We connect to x_1 on the aperture (deterministic if pinhole, else sampled)

  ➔ ω, q are implicit via this combination of surface points

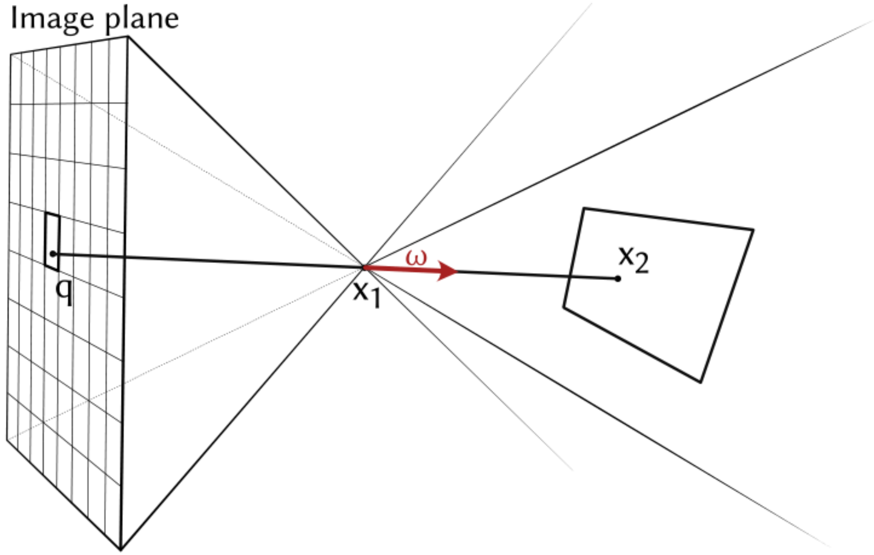
$$I_i = \int_S \int_L h_i(q)\, W(q, x_1, \omega)\, L_i(x_1, \omega)\, dx_1 dq$$

- Nothing special: change of variables → multiply by Jacobian

$$I_i = \int_A \int_L h_i(q)\, W(q, x_1, \omega)\, L_i(x_1, \omega) \frac{dq}{dx_2}\, dx_1 dx_2$$

Jacobian for a pinhole camera

- We derive it in two steps, by separating into product:



$$\frac{dq}{dx_2} = \frac{dq}{d\omega}\frac{d\omega}{dx_2}$$

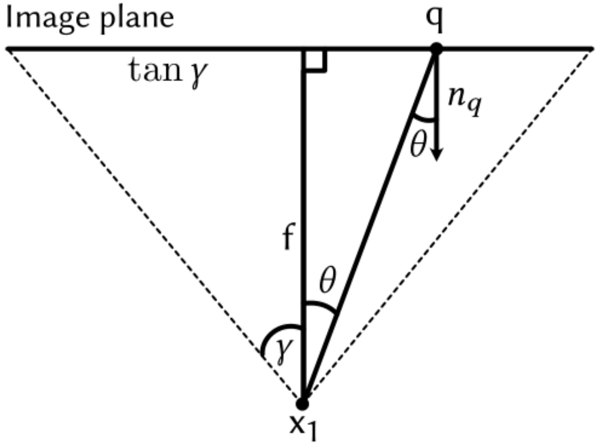
Hemisphere → surface Jacobian

$$\frac{dq}{d\omega} = \frac{\cos\theta}{\|q - x_1\|^2}$$

Good old surface → hemisphere Jacobian $\frac{d\omega}{dx_2} = \frac{\|x_2 - x_1\|^2}{\cos\theta(x_2 \to x_1)}$

Substituting $\cos\theta = \frac{f}{\|q - x_1\|}$

$$\Rightarrow \frac{dq}{d\omega} = \frac{\cos^3\theta}{f}$$



$$\frac{dq}{dx_2} = \frac{\cos^3\theta}{f}\frac{\|x_2 - x_1\|^2}{\cos\theta(x_2 \to x_1)}$$

Integral formulation for a light tracer

- Same as for irradiance, except:
  - Integrate over aperture L (if not a pinhole)
  - Multiply by Jacobian for hemisphere → sensor: $\frac{dq}{d\omega}$

$$\int_A \int_\Omega \dots \int_\Omega \int_L h_i(q)\, W(q, x_1, \omega)\, f(x_1 \dots x_k) \frac{\cos^3 \theta}{f} dx_1\, d\overleftarrow{\omega}_2 \dots d\overleftarrow{\omega}_k\, dx_k$$

Same as with irradiance before

Pixel filter and sensor response (camera model)

The light tracer estimator for path of length k

- Sample x_k on the light, sample direction
- Trace ray to find x_{k-1}, sample direction from there
- Repeat until x_2 was found
- Sample x_1 on the camera aperture, connect with shadow ray (to compute $V(x_1, x_2)$)
- Log contribution in whatever pixel(s) i contain(s) the sensor point q

$$\frac{L_e \cos\theta(x_k \to x_{k-1})}{p(x_k)p(x_k \to x_{k-1})} \left(\prod_{i=2}^{k-1} \frac{f(x_{i+1}, x_i, x_{i-1}) \frac{\cos\theta_{x_i}}{\cos\theta(x_i \to x_{i+1})} \cos\theta(x_i \to x_{i-1})}{p(x_i \to x_{i-1})} \right) \frac{V(x_1, x_2) \frac{\cos\theta_{x_1}}{\|x_2 - x_1\|^2} h_i(q) W(q, x_1, \omega) \frac{\cos^3\theta}{f}}{p(x_1)}$$

Now we can render with pure light tracing

Light tracing

Path tracing

The path integral

A common ground for bidirectional techniques

We want to combine forward and backward sampling

- Need a common integral formulation / integration domain
- Why?
  - MIS is only possible if we have a consistent domain!
  - Also: notational convenience

The path integral

Value of pixel i

Surface integrals (one per vertex)

$$I_i = \sum_{k=2}^{\infty} \int_A \dots \int_A f_i(x)\, dx_1 \dots dx_k$$

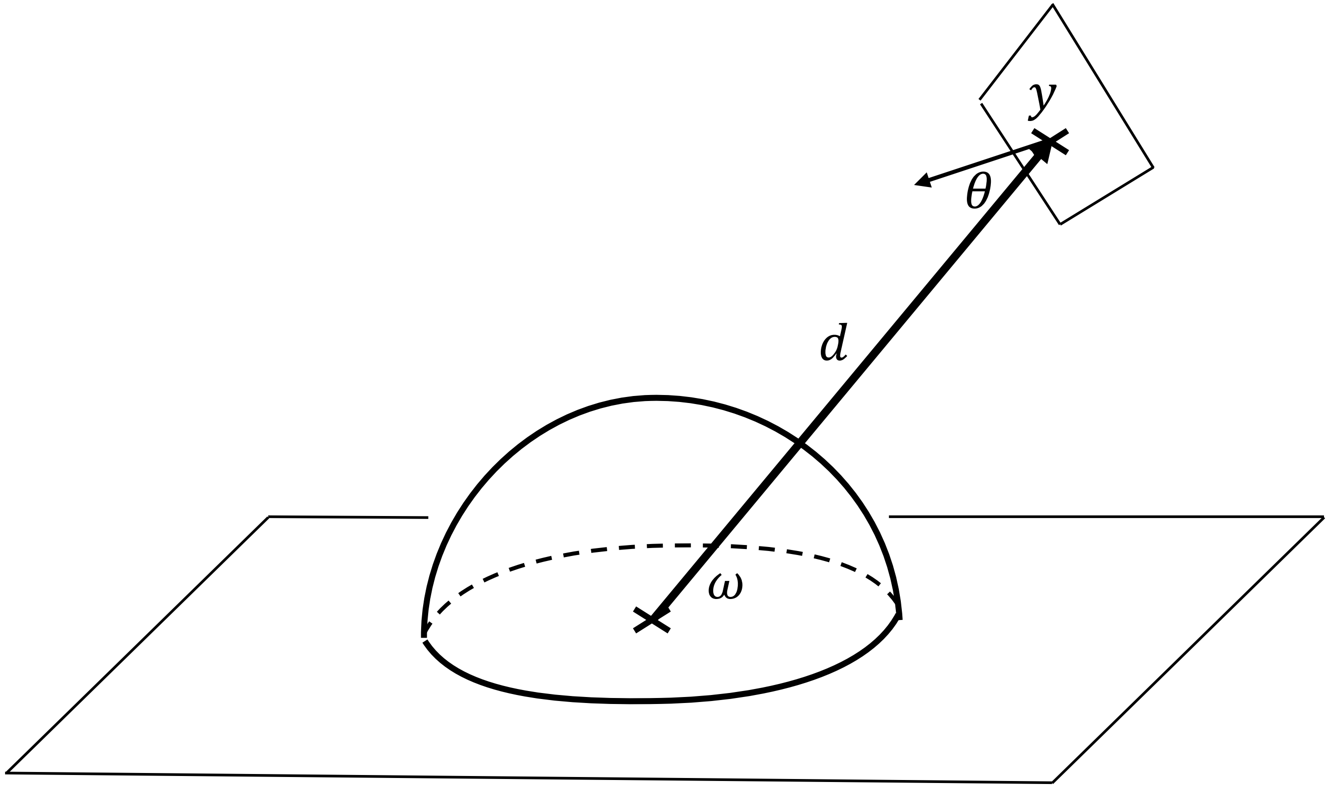
Sum over all path lengths

$$f_i(x) = \underbrace{\overbrace{h_i(q) W(q, x_1, \omega)}^{\text{Camera model}} \overbrace{\frac{\cos^3\theta}{f} \frac{\|x_2 - x_1\|^2}{\cos\theta(x_2 \to x_1)}}^{\text{Camera Jacobian}} V(x_1, x_2)}_{\text{“Sensor response”}} \underbrace{\left(\prod_{j=2}^{k-1} \overbrace{f(x_{j-1}, x_j, x_{j+1}) \cos\theta_{x_j}}^{\text{BSDF, cosine from RE}} V(x_j, x_{j+1}) \overbrace{\frac{\cos\theta(x_{j+1} \to x_j)}{\|x_j - x_{j+1}\|^2}}^{\text{Local Jacobian}} \right) L_e}_{\text{“Throughput”}}$$

Sampling directions via points on a surface

1. Sample a point y on the surface (e.g., light source)
2. Compute

$$p(\omega) = p(y)\frac{\|y - x\|^2}{\cos\theta}$$



Previously on RIS

The light tracer as a path integral estimator

- Light tracing samples a path of length k with the product PDF

$$p(x_k)\left(\prod_{i=2}^{k} p(x_i \to x_{i-1})\right)p(x_1)$$

Surface PDF — Direction PDFs — Surface PDF

- By mapping the directions $x_i \to x_{i-1}$ to surface points, we can define a product surface density

$$p_{LT}(x) = p(x_k)\left(\prod_{i=2}^{k} p(x_i \to x_{i-1})\frac{\cos\theta(x_{i-1} \to x_i)}{\|x_{i-1} - x_i\|^2}\right)p(x_1)$$

Forward path tracing as a path integral estimator

- Similarly, forward path tracing generates path samples with PDF

$$p_{PT}(x) = p(x_1)p(q)|J_{cam}(x_2)| \prod_{i=2}^{k-1} p(x_i \to x_{i+1}) \frac{\cos\theta(x_{i+1} \to x_i)}{\|x_i - x_{i+1}\|^2}$$

Integral transformation = sample transformation

- The path integral estimator with that forward path PDF is:

$$\frac{f_i(x) = h_i(q)W(q, x_1, \omega)|J_{cam}(x_2)|V(x_1, x_2)\left(\prod_{j=2}^{k-1} f(x_{j-1}, x_j, x_{j+1}) \cos\theta_{x_j} V(x_j, x_{j+1}) \frac{\cos\theta(x_{j+1} \to x_j)}{\|x_j - x_{j+1}\|^2}\right) L_e}{p_{PT}(x) = p(x_1)p(q)|J_{cam}(x_2)| \prod_{i=2}^{k-1} p(x_i \to x_{i+1}) \frac{\cos\theta(x_{i+1} \to x_i)}{\|x_i - x_{i+1}\|^2}}$$

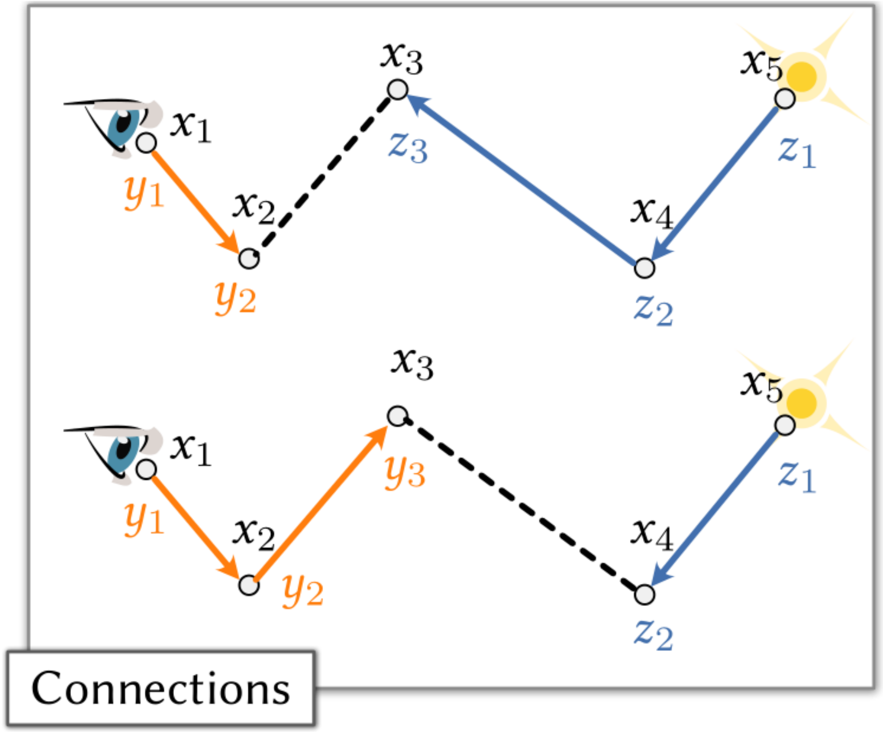
- Same as a rendering equation estimator with direction PDFs:

$$\frac{f_i(x)}{p_{PT}(x)} = \frac{h_i(q)W(q, x_1, \omega)\left(\prod_{j=2}^{k-1} f(x_{j-1}, x_j, x_{j+1}) \cos\theta_{x_j}\right) L_e}{p(x_1)p(q) \prod_{i=2}^{k-1} p(x_i \to x_{i+1})}$$

Bidirectional connections

Tracing and connecting paths from both ends

- Multiple techniques possible
- Path of length k is sampled by tracing t rays from the camera and $k - t$ from the light

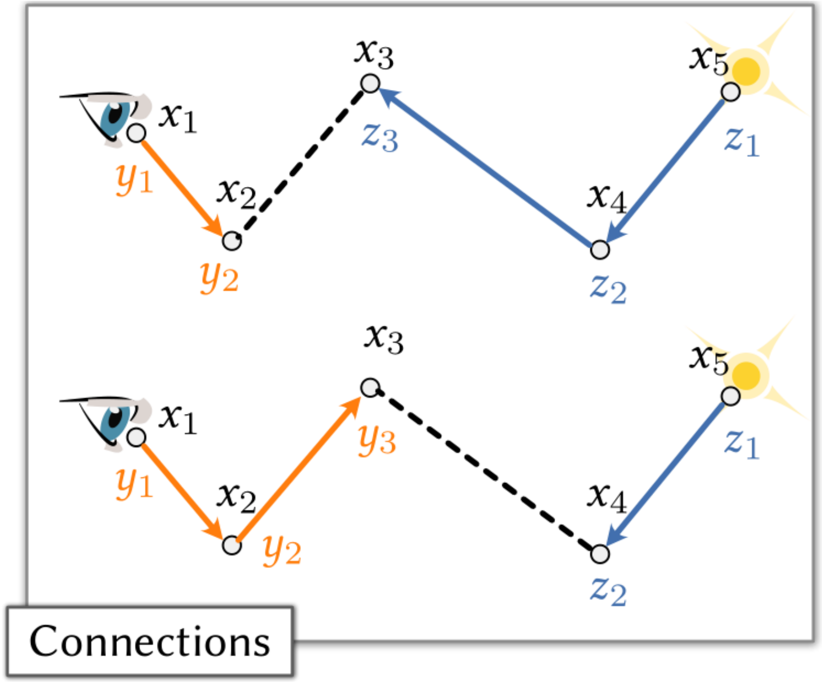


The path PDF for connections

$$p_{c,t}(\bar{x}) = p(\bar{y}_t)p(\bar{z}_s)$$

$$p(\bar{y}_t) = p(y_1)p(q)|J_{\text{cam}}|\left(\prod_{i=2}^{t-1} p(y_i \to y_{i+1})\frac{\cos\theta(y_{i+i} \to y_i)}{\|y_i - y_{i+1}\|^2}\right)$$

$$p(\bar{z}_s) = p(z_1)\left(\prod_{i=1}^{s-1} p(z_i \to z_{i+1})\frac{\cos\theta(z_{i+1} \to z_i)}{\|z_i - z_{i+1}\|^2}\right).$$



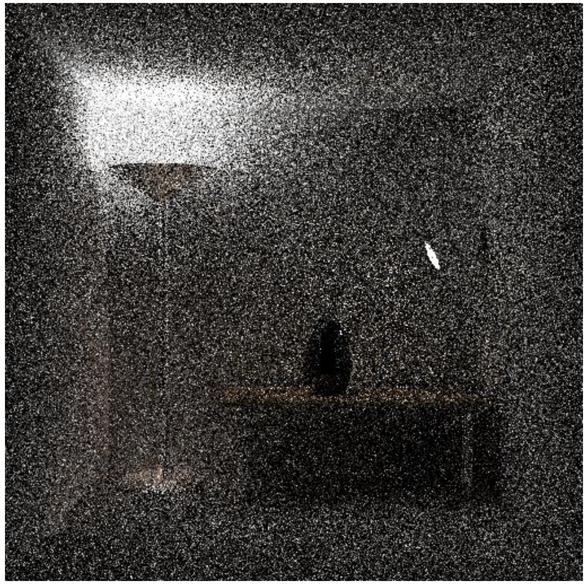
MIS

Let’s combine these techniques

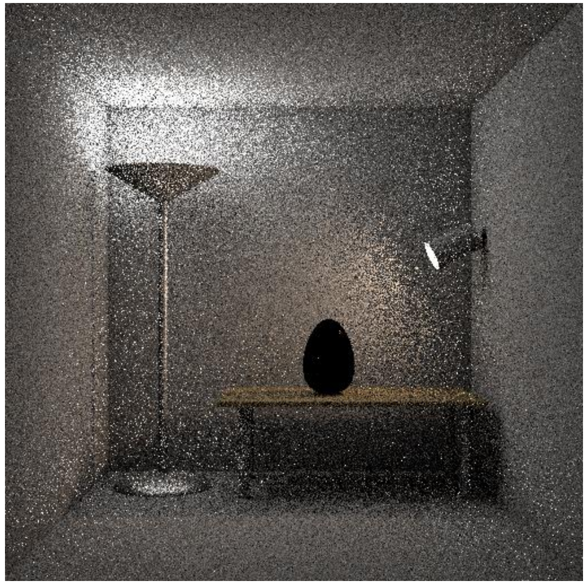
Each technique estimates the full integral

Light tracing

Forward path tracing

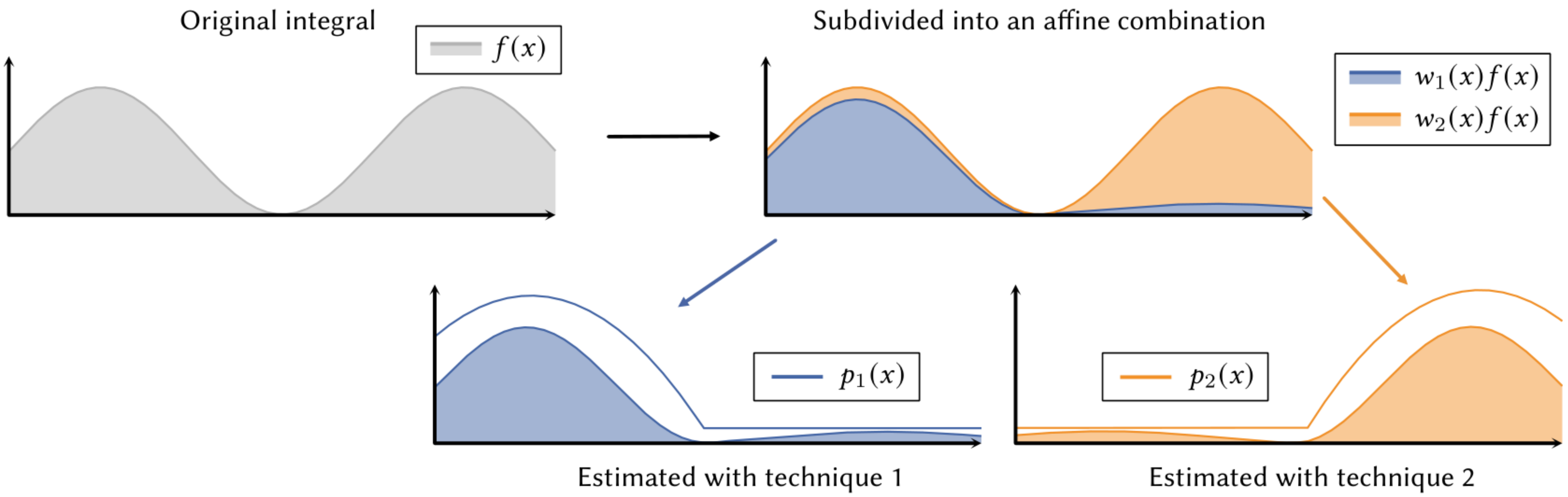


Connect first



Connect second

Recap: MIS



$$F_{MIS} = \sum_t \frac{1}{n_t} \sum_{i=1}^{n_t} w_t(x_{t,i}) \frac{f(x_{t,i})}{p_t(x_{t,i})} \approx \sum_t \int w_t(x) f(x)\, dx = \int f(x)\, dx \quad (\text{if } \textstyle\sum_t w_t(x) = 1)$$

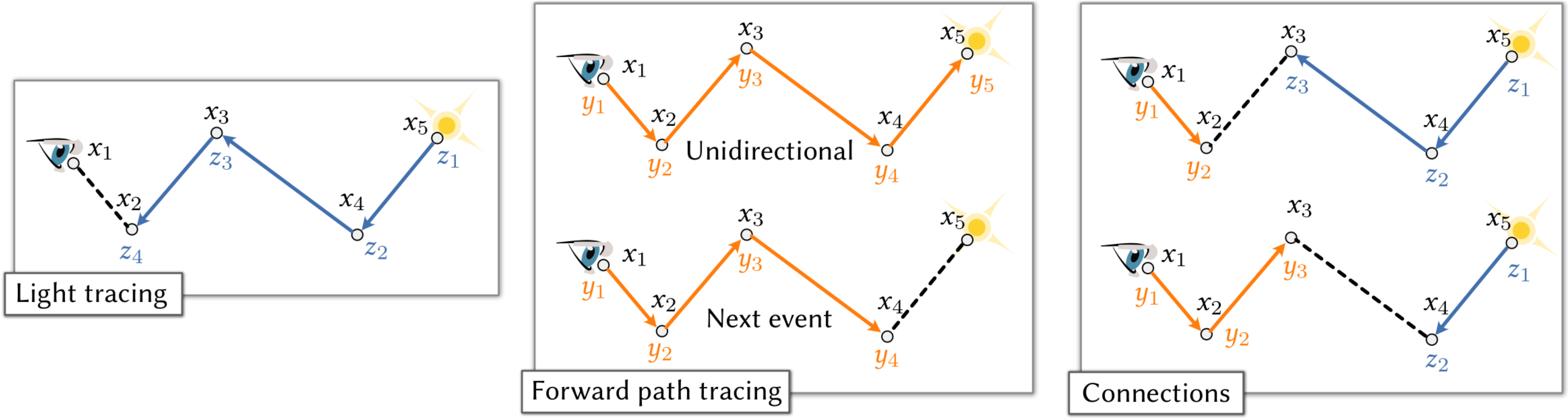
Recap: Balance heuristic

$$w_t(x) = \frac{n_t p_t(x)}{\sum_{t'} n_{t'} p_{t'}(x)}$$

All PDFs **must** be in the **same domain**

Evaluating the balance heuristic

- The biggest challenge of a BDPT implementation
- Need to gather all PDFs and make sure that all are correctly converted to compatible domains



Example: Weight of LT for path of length 3 (i.e., direct illum.)

$$w_{LT} = \frac{p_{LT}}{p_{LT} + p_{PT} + p_{NEE}}$$

PDFs are products of many local sampling decisions with different Jacobians. Challenging to get right.

$$p_{LT}(x) = p(x_3)p(x_3 \to x_2)\frac{\cos\theta(x_2 \to x_3)}{\|x_2 - x_3\|^2}p(x_1)$$

$$p_{PT}(x) = p(x_1)p(q)|J_{cam}(x_2)|\, p(x_2 \to x_3)\frac{\cos\theta(x_3 \to x_2)}{\|x_2 - x_3\|^2}$$

$$p_{NEE}(x) = p(x_1)p(q)|J_{cam}(x_2)|\, p(x_3)$$

Products of PDFs are dangerous

$$p_{LT}(x) = p(x_3)p(x_3 \rightarrow x_2)\frac{\cos\theta(x_2 \rightarrow x_3)}{\|x_2 - x_3\|^2}p(x_1)$$

If these are high → $p_{LT}(x)$ can be extremely huge

E.g., series of highly glossy scattering events.

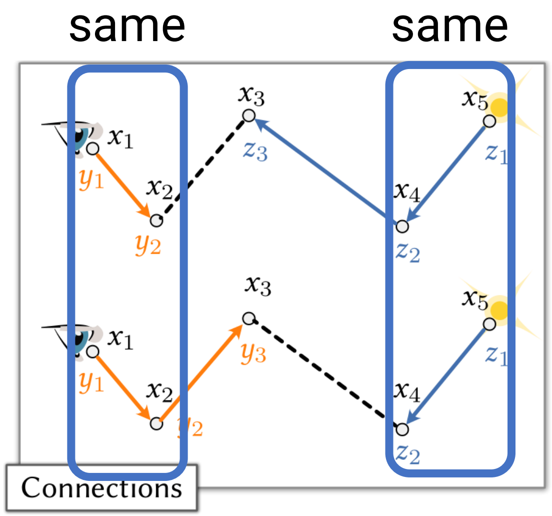
So we compute ratios instead

$$w_{LT} = \frac{p_{LT}}{p_{LT} + p_{PT} + p_{NEE}} = \left(1 + \frac{p_{PT}}{p_{LT}} + \frac{p_{NEE}}{p_{LT}}\right)^{-1}$$

$$\frac{p_{PT}}{p_{LT}} = \frac{p(q)|J_{cam}(x_2)|}{p(x_3 \to x_2)\frac{\cos\theta(x_2 \to x_3)}{\|x_2 - x_3\|^2}} \quad \frac{p(x_2 \to x_3)\frac{\cos\theta(x_3 \to x_2)}{\|x_2 - x_3\|^2}}{p(x_3)}$$

PDF to sample x_2 PDF to sample x_3

$$\frac{p_{PT}}{p_{LT}} = \frac{p(q)|J_{cam}(x_2)|}{p(x_3 \to x_2)\cos\theta(x_2 \to x_3)} \quad \frac{p(x_2 \to x_3)\cos\theta(x_3 \to x_2)}{p(x_3)}$$



Numerically much better. And many common ratios across techniques!

MIS weighted techniques

Light tracing

Forward path tracing

Connect first

Connect second

MIS weighted techniques

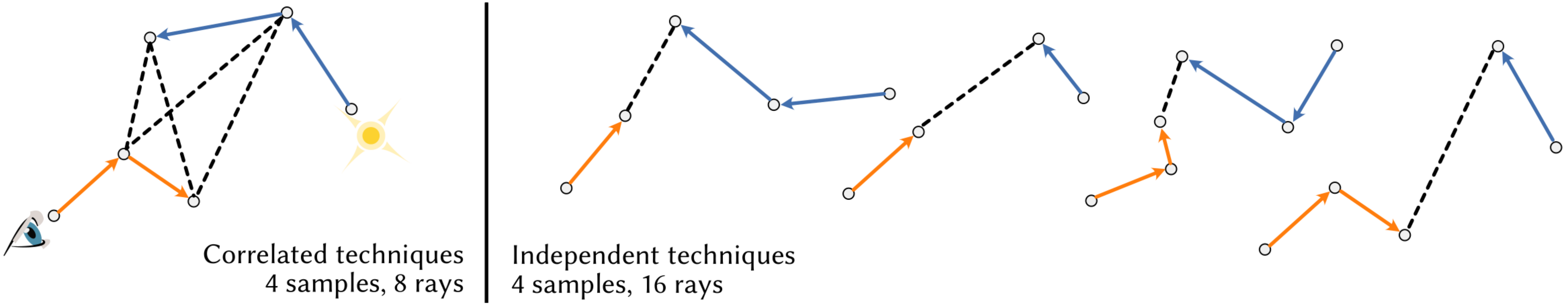
$$= \sum$$

Implementation

There are many ways of doing it – here's one

Reusing computations

- Tracing one full path per technique is expensive:
  - Number of techniques grows quadratically with the path length!
- Instead: trace one set of subpaths and reuse components
- Exception: next event uses a separate light sample (specialized techniques available)



Progressive rendering with two passes

- Pass 1: Trace n light paths and store them
  - At every vertex except the first:
    - Connect to the camera – aka light tracing
- Pass 2: Trace 1 camera path per pixel
  - At every vertex except the first:
    - Connect to one (or more) stored light vertices
    - Connect to one (or more) random point(s) on a light – aka next event estimation
    - Check if this is a light source, add contribution if so

Technical details

- Compute pdf of infinite light in surface area
  - Sampling a distant disc with radius as large as the scene and a Dirac direction distribution
- Efficiently compute the pdf for MIS without redundancy
  - Take advantage of prefix/suffix change between techniques
- Dirac delta distribution
  - Can be avoided in many case with analytic simplification
- Non-symmetrical scattering

- Many other technical details need to be implemented
  - PBRT book give some good insight on solutions (online book)

Much design freedom

- How to store light paths?
- Which paths to store?
- How many light paths?
- How many camera paths per pixel?
- How to select light vertices for connection? At random? By pairing deterministic paths?
- How many connections to use?
- Trace camera paths first and then light paths
- …

Limitations and challenges

How to improve this basic recipe and make it practical?

Specular – diffuse – specular (SDS) paths

- Photon Mapping / VCM to the rescue → upcoming lectures

BDPT

VCM

Wasted computation

- Light paths might not be visible
- Light paths might not be needed
- Redundancy in MIS techniques
- Possible solutions:
  - Path guiding → upcoming lecture
  - Markov chain Monte Carlo → upcoming lecture
  - Optimizing sample counts → https://graphics.cg.uni-saarland.de/publications/grittmann-sig2022.html

Summary

$$I_i = \sum_{k=2}^{\infty} \int_A \dots \int_A f_i(x)\, dx_1 \dots dx_k$$

- Trace paths from both ends
- Combine in various ways, reuse subpaths for efficiency
- Each combination technique: A correct estimator for the pixel values
- Combine techniques via MIS
- Path integral (aka iterated surface integral) as the common domain for MIS