SIMULATING LIGHT IN LARGE VOLUME DETECTORS USING METROPOLIS LIGHT TRANSPORT

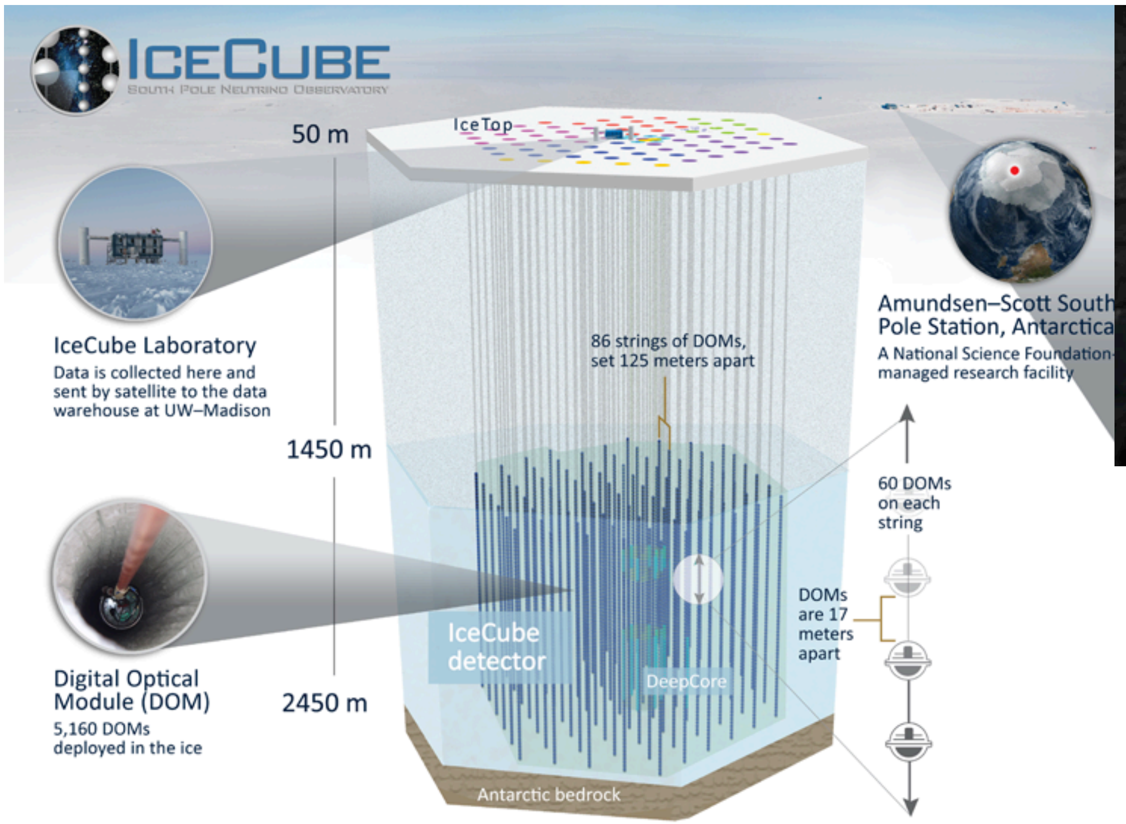
Gabriel Collin
MIT

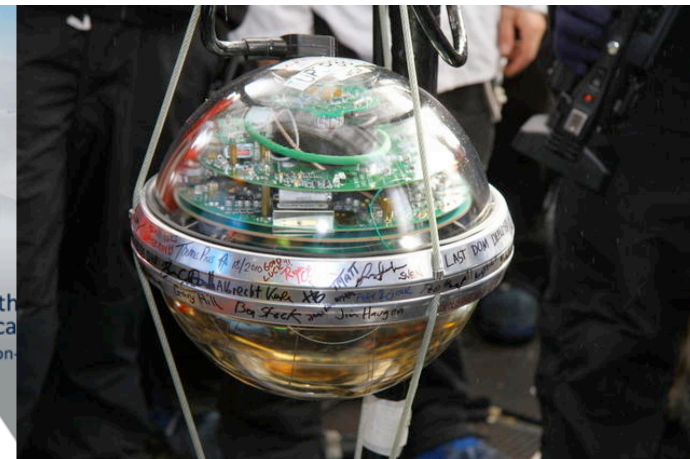
MIT INSTITUTE FOR DATA, SYSTEMS, AND SOCIETY

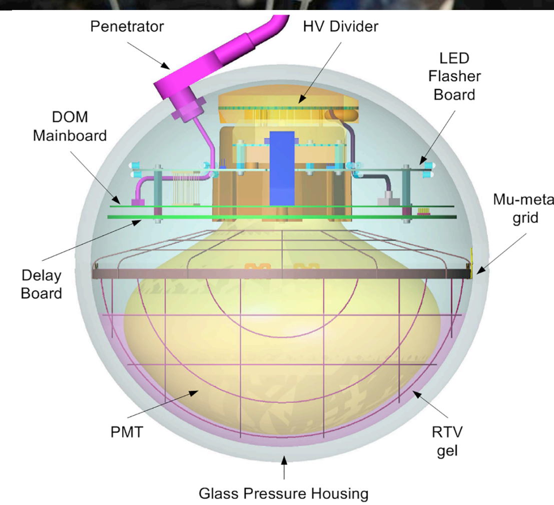
MIT Physics

IceCube

- Gigaton neutrino detector located at the south pole.

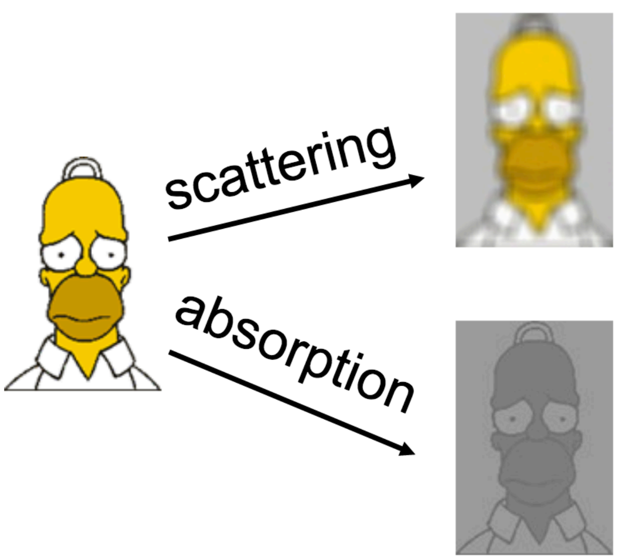






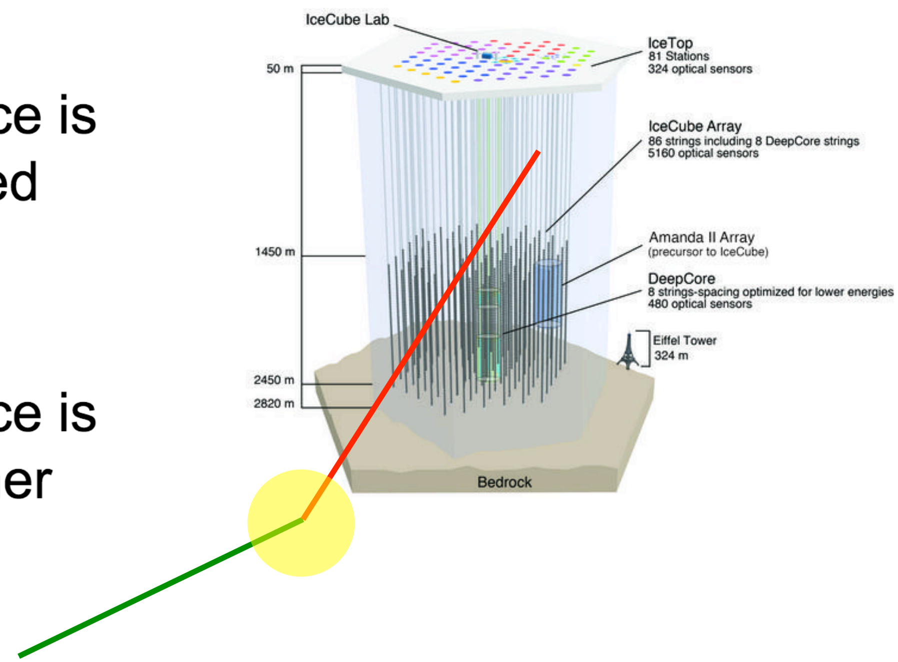
IceCube

- Muon neutrinos interact with the surrounding ice/rock and produce muons that travel through the detector.
  - Produces Cherenkov light as it travels.
- Cherenkov light is scattered and absorbed
  - Effects the angular and energy resolution.



Source is blurred

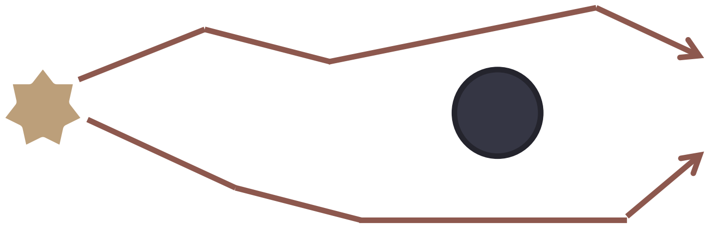
Source is dimmer



Example by Dmitry Chirkin

Motivation

- Currently, IceCube uses ray tracing to propagate light in the ice.
- However, most rays never reach a DOM.



- Ray tracers can be run *backwards* in time, but then most rays will never reach a light source.
    - Ray tracers can’t constrain both the starting and ending location of the rays.

- The fundamental problem is that the interesting paths are highly constrained.
    - Is there another way to approach this?

Path integration

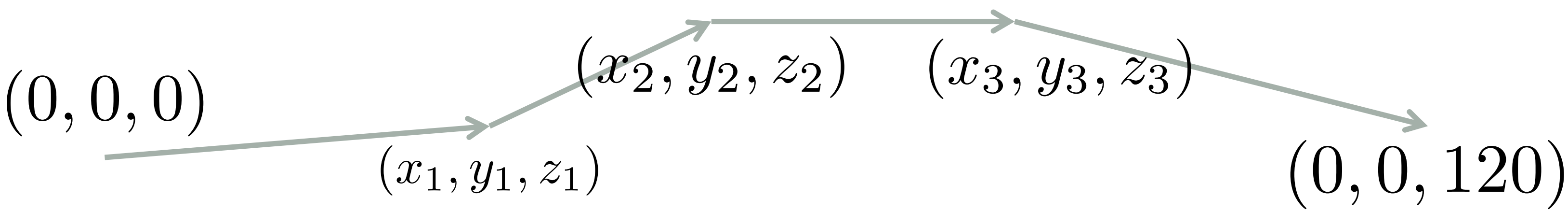
- The start and end locations of the ray can be constrained if the problem is specified in terms of a classical path integral.

$$\int_{f \in \Omega} e^{-S[f]} Df$$

Space of all paths

Probability of path 'f'

- Eg: $f = \{(0,0,0), (x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (0,0,120)\}$



Evaluation of the integral

- Information can be extracted about the light propagation by framing the integrand as a probability distribution:

$$e^{-S[f]} \rightarrow p[f] = p(x_1, y_1, z_1, x_2, y_2, z_2, \ldots)$$

- This distribution can be sampled with an MCMC.
- More details on the construction of p[f] in the paper:

arXiv.org > hep-ex > arXiv:1811.04156

High Energy Physics – Experiment

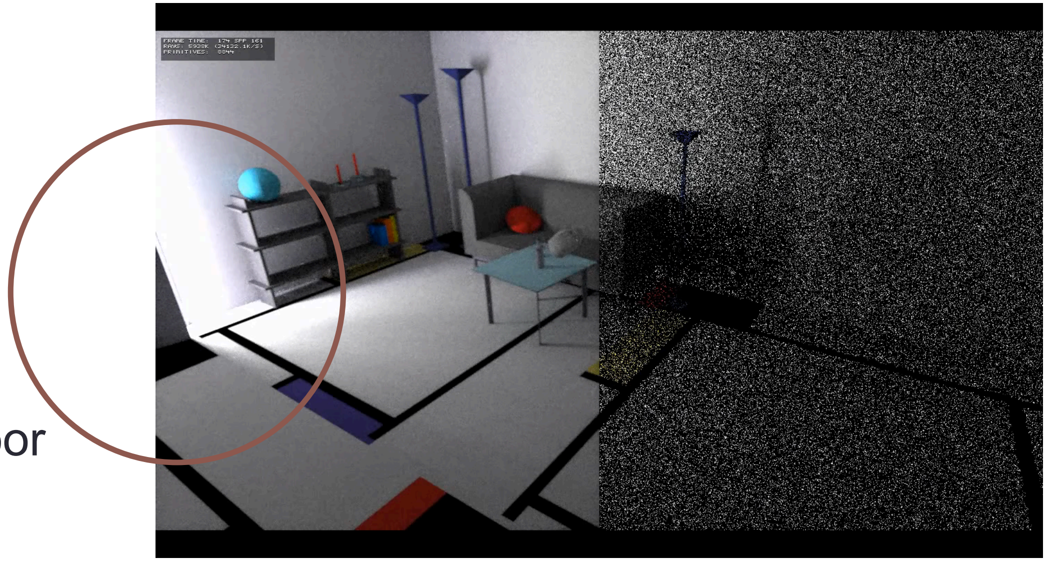
Using path integrals for the propagation of light in a scattering dominated medium

Gabriel H. Collin

Industry use

- This idea inspired by a CGI rendering technique called Metropolis light transport.
  - Computer animation often runs into a similar problem to us, where only a small fraction of light paths are detectable.
  - Canonical example is a light source in another room that shines through a door that is only slightly cracked open.

Slightly open door



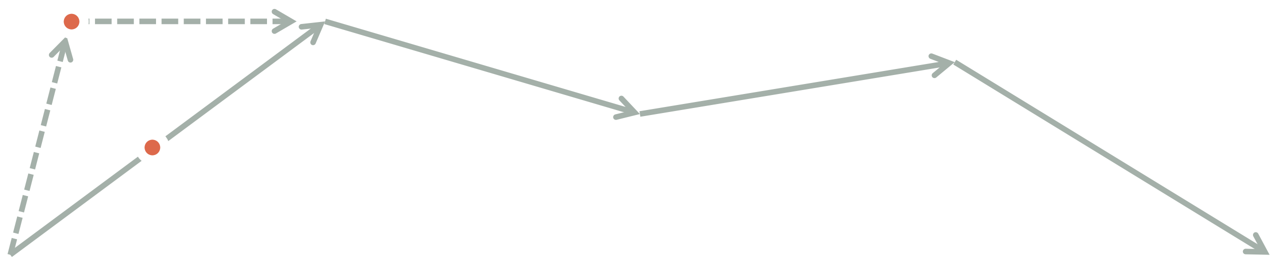
Left: Rendering algorithm similar to Metropolis light transport.
Right: Standard path tracing algorithm.

- CGI industry mainly renders scenes that are dominated by reflections.
  - In IceCube, light transport is entirely scattering.

http://raytracey.blogspot.com.es/2010/12/real-time-metropolis-light-transport-on.html

Reversible jump MCMC

- The number of places where light scatters is not fixed.
  - Thus the dimensionality of the probability distribution is variable.



- Reversible Jump Markov Chain Monte Carlo can change the number of dimensions in a probability distribution.

Reversible jump MCMC

- The acceptance probability is based on the following ratio:

$$\frac{p_1(\vec{\phi})}{p_0(\vec{\theta})q(\vec{q})} \frac{P_{1\to0}}{P_{0\to1}} \left| \frac{\partial g(\vec{\theta},\vec{q})}{\partial(\vec{\theta},\vec{q})} \right|$$

Proposal rates

Jacobian of proposal function.

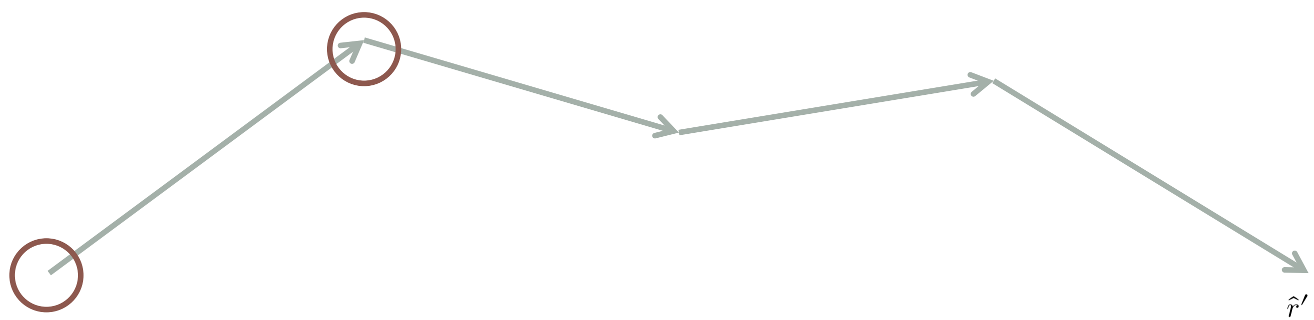
Padded probability distribution

- q can be marginalised out later for free.

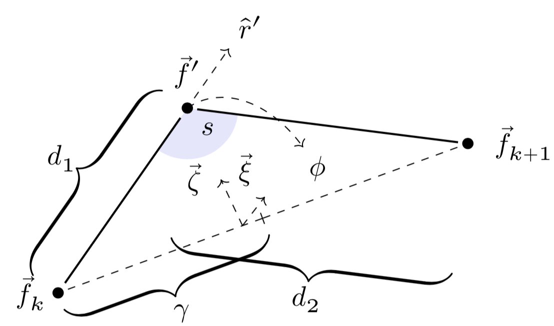
Reversible jump for light propagation

- A path with N vertices exists in $\mathbb{R}^{3N}$
  - We wish to propose a new path with N+1 vertices.
  - Requires a *q* with 3 parameters, and a choice of *g*.

- g selects a pair of vertices.



- Then inserts a new vertex between them.
  - Position of new vertex based on three random values from q



Path length distribution

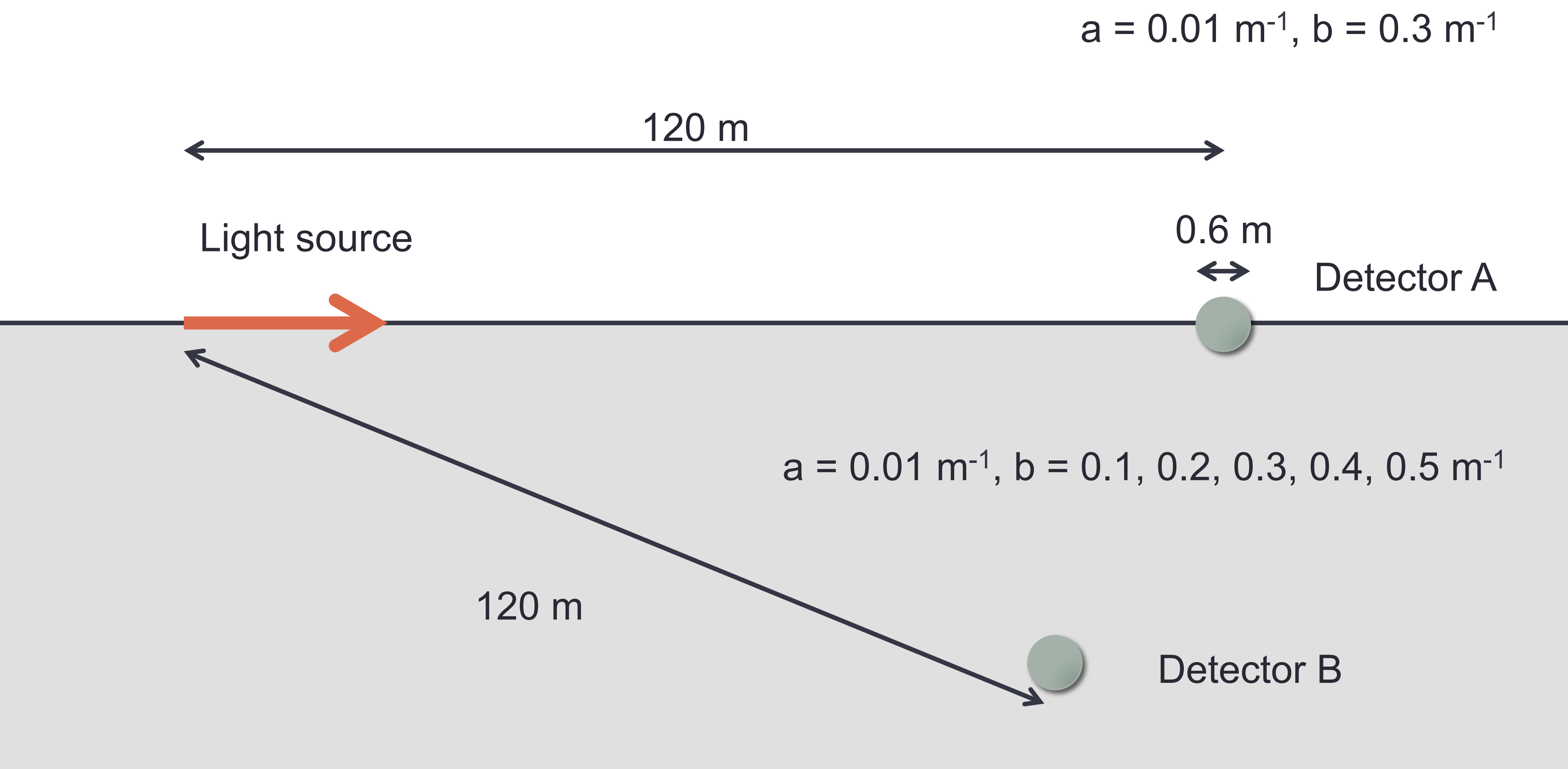
- From the samples created by the MCMC, the probability distribution for path length can be easily extracted.
  - IceCube measures photon arrival time, which is directly related to path length.
  - $P(L < X)$ = fraction of samples where the length of the path is less than X.

- To validate the method, the length distribution produced by the path sampler can be compared to one created using a ray tracer.

- An MCMC usually requires a burn-in period, however this can be partially avoided by seeding the MCMC with the ray-tracer.

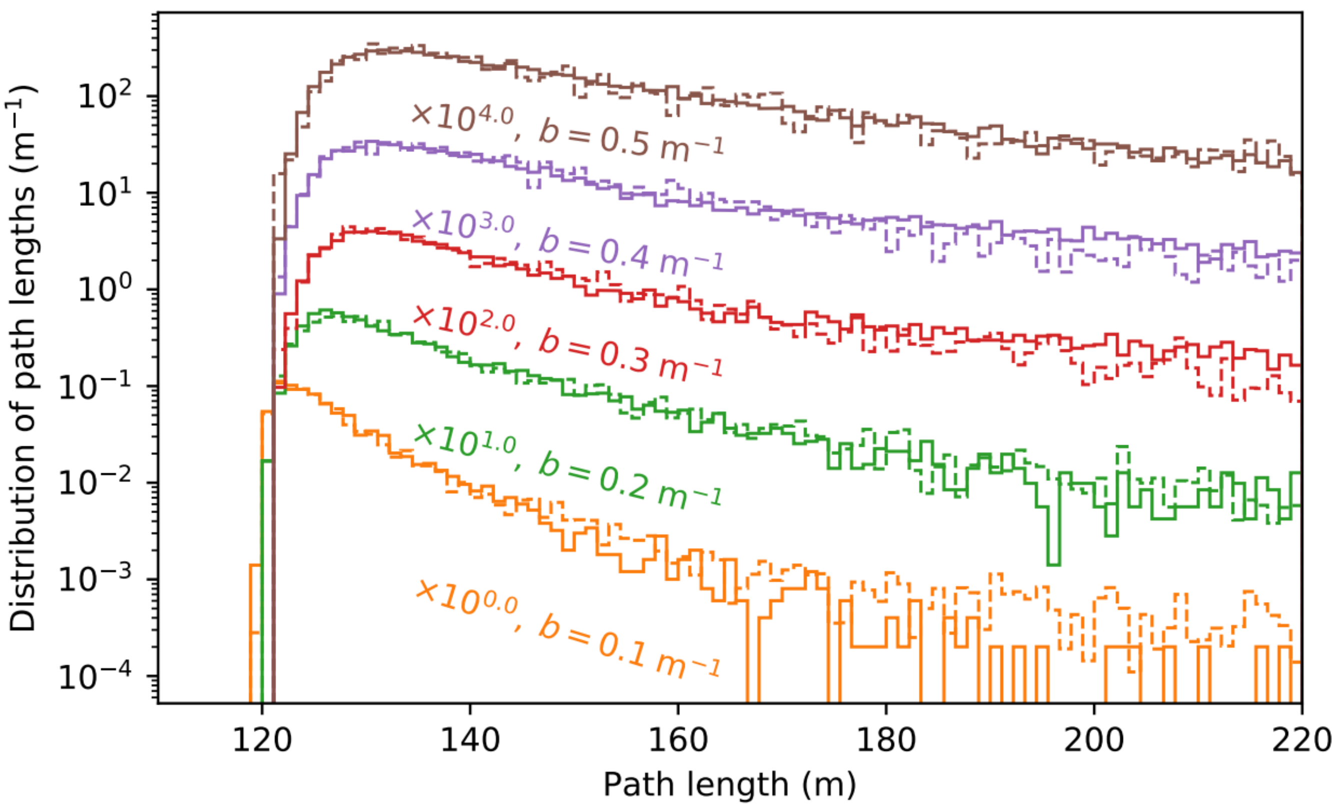
Synthetic test case

- One light source, with two detectors



Path length distribution

- Solid: path sampler. Dashed: reference ray tracer
  - Ray tracer was run until 5000 samples collected.
  - Path sampler was run until results matched the path sampler.



Detector A

Acceptance rate ~20%

Lines offset for readability

Performance

- Ray tracer is also CPU based to allow a performance comparison.

b	Ray tracer	Path sampler
0.1 m^{-1}	~46000 s	~23 s
0.2 m^{-1}	~78000 s	~74 s
0.3 m^{-1}	~99000 s	~232 s
0.4 m^{-1}	~122000 s	~373 s
0.5 m^{-1}	~156000 s	~416 s

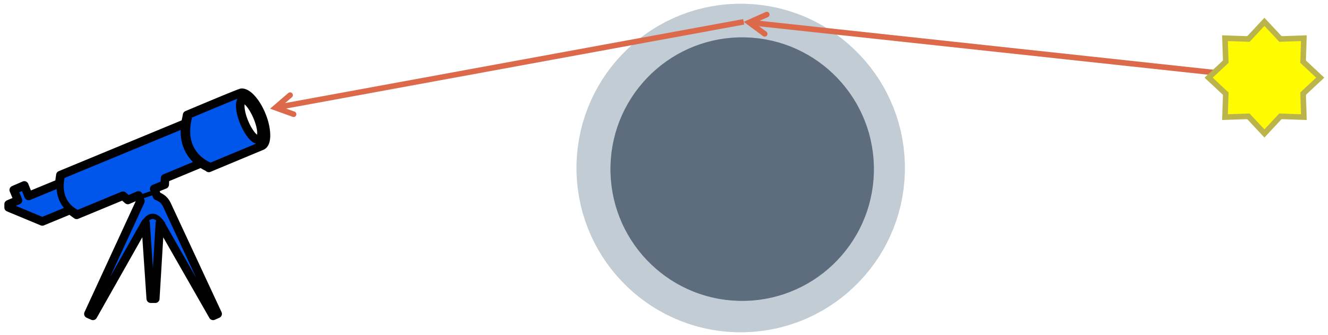
- Performance improvement of 300 to 1000 times faster.
  - The b = 0.3 to 0.5 m^{-1} cases are probably most comparable to conditions in IceCube.

Other applications

- This approach to simulation is useful when initial and final states are highly constrained.

- Litmus test:
  - Are you throwing out the vast majority of your events (99.9%+) due to them not meeting one of these constraints?

- Constraints do not have to just be in position.
  - Eg: initial and final angle for light passing through a planetary atmosphere.



Other applications

- Path does not just have to describe light.
  - Eg: Simulation of transport of neutrons.
- Constraints could be discrete parameters.
  - Eg: Simulation of atmospheric showers.
    - Initial condition: particle must be a nucleus.
    - Final condition: shower products must reach underground detector.
- May also be possible to incorporate selection cuts into the constraint.

Conclusion

- Simulation of light can be posed as a path integral from which samples can be drawn.

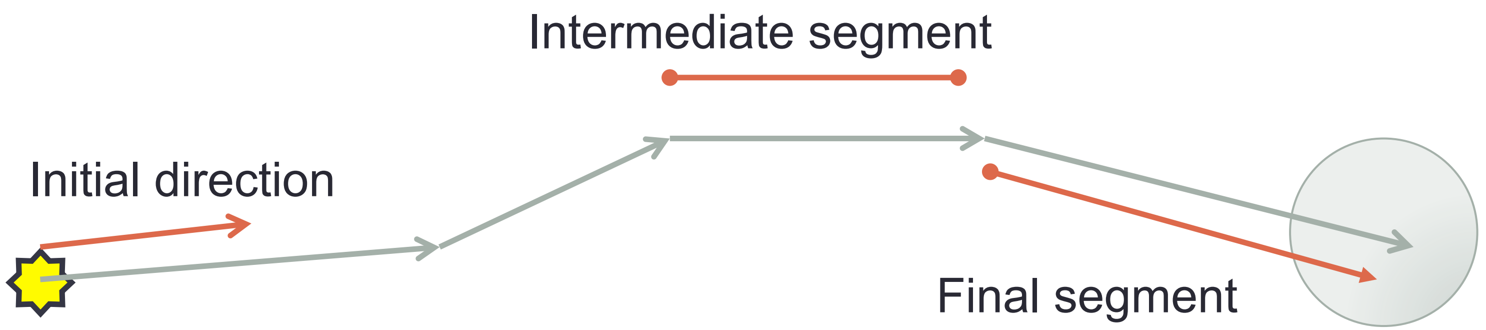
- Reproduces the timing distribution of light incident on a detector.
  - Up to 1000x faster than a ray tracer in synthetic test case.

- Method is generally applicable to a wide range of problems.
  - When initial and final states are highly constrained.

BACKUP

Probability distribution

- The probability distribution has three main parts:
  - A factor for the initial direction (probability of emission).
  - A factor for each segment in the path (except the last).
  - A factor for the last segment, including the probability of detection.



Probability distribution

- The first factor:
  - Determined by the light source.
    - Here the light source is assumed to be a point.
    - Can be extended to line or spherical sources.
  - Here, probability distribution chosen to be a von Mises-Fisher distribution:

$$\varepsilon(\hat{r}_0) = \frac{\kappa e^{-\kappa \hat{r}_0 \cdot \hat{\varepsilon}}}{4\pi \sinh \kappa}$$

Probability distribution

- The second factor:
  - Repeated for each segment (except the last).
  - Is the probability of:
    - Light scattering at x_i after traveling along the line segment, and
    - Light changing direction according to the next segment.

$$p_i = \underbrace{b(x_i)e^{-\tau_i}}_{\text{Exponential distribution for scattering}} \underbrace{\sigma(\cos\Delta\theta_i)}_{\text{Angular scattering distribution}}$$

Exponential distribution for scattering

Angular scattering distribution

Optical depth: $$\tau_i = \int [a(x(s)) + b(x(s))]\, ds$$

Probability distribution

- The third factor:

  - Is the probability of:
    - Light traveling along the last segment **without** scattering, and
    - The detection efficiency where the light ends on the sphere.

$$p_f = e^{-\tau_f} \rho(x_f)(\hat{x}_f \cdot \hat{n})$$

Exponential CDF for the survival of light

Detection efficiency

2D constraint term

  - The constraint that the final vertex of the path must lie on a 2D spherical surface introduces an extra factor of cos(θ).

Probability distribution

- The total probability is the product of these factors:

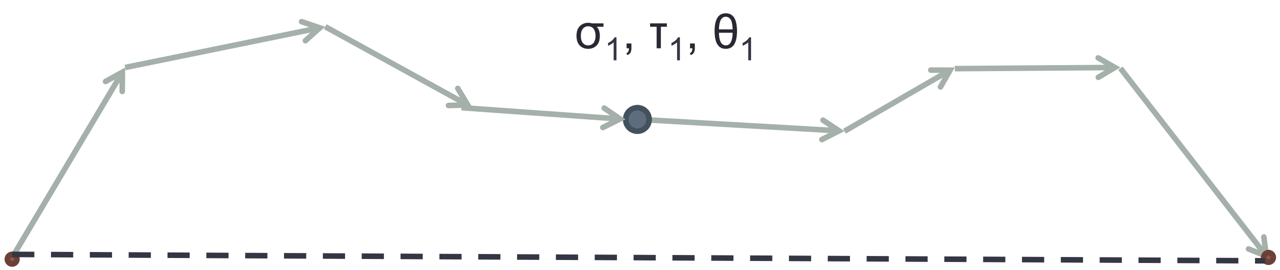
$$p = \epsilon(\hat{r}_1) \left[\prod_{i=1}^{n-1} p_i \right] p_f$$

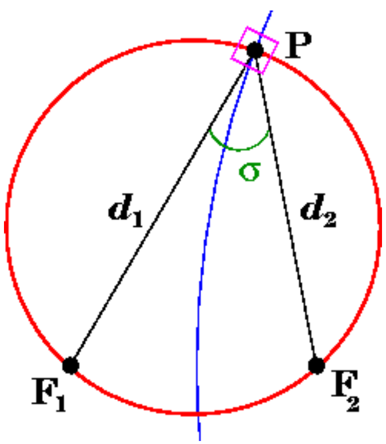
Choice of coordinates

- The method of proposing new coordinates has a large impact on the efficiency of the sampler.
- As the angular probability distribution for scattering in ice is very forward focused, the coordinates are highly correlated with each other.
- In addition, the length scales of the probability distribution is a function of the distance between vertices.
  - A simple normal distribution based proposal function results in very poor performance.
- One solution is to de-correlate through a good choice of coordinates.

Tree based coordinates

- System is defined in a nested form like a tree.



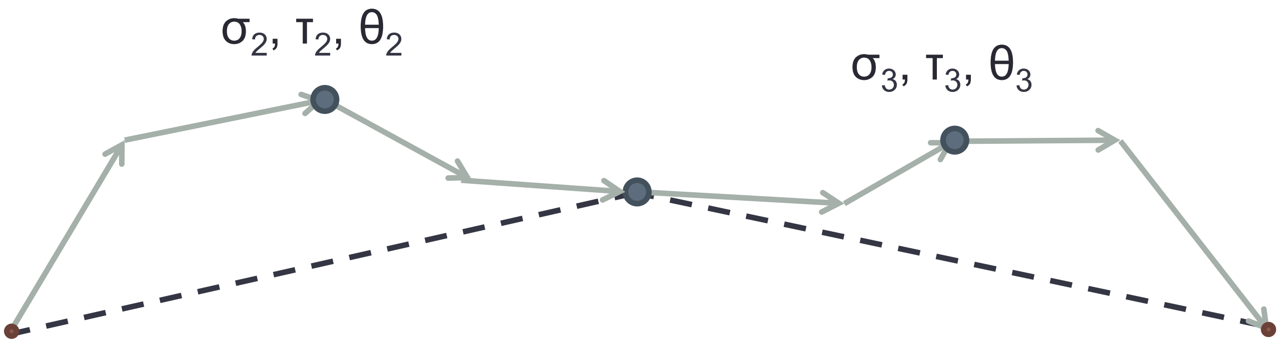


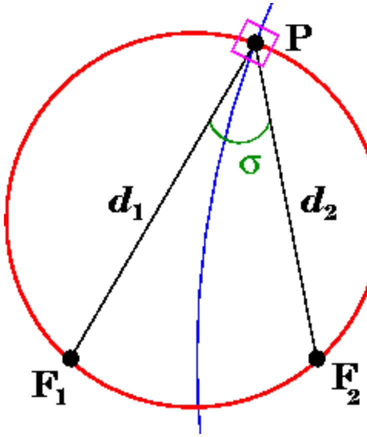
$$\tau = \ln \frac{d_1}{d_2}$$

σ_1, τ_1, θ_1

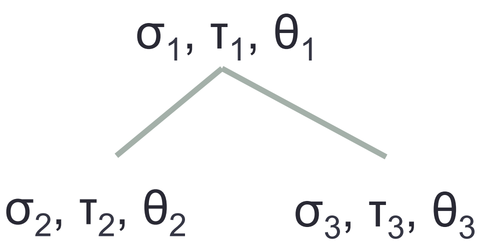
Tree based coordinates

- System is defined in a nested form like a tree.



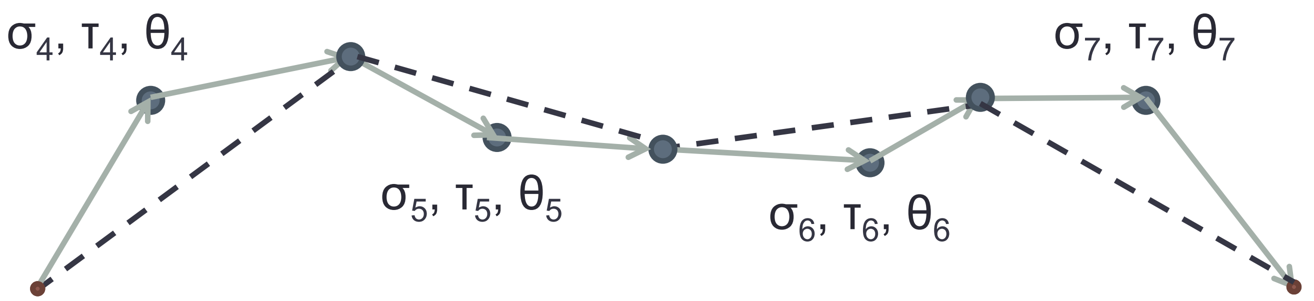


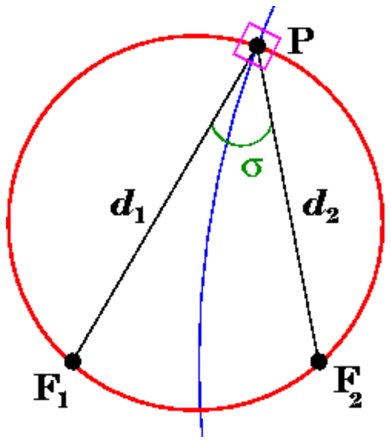
$$\tau = \ln \frac{d_1}{d_2}$$



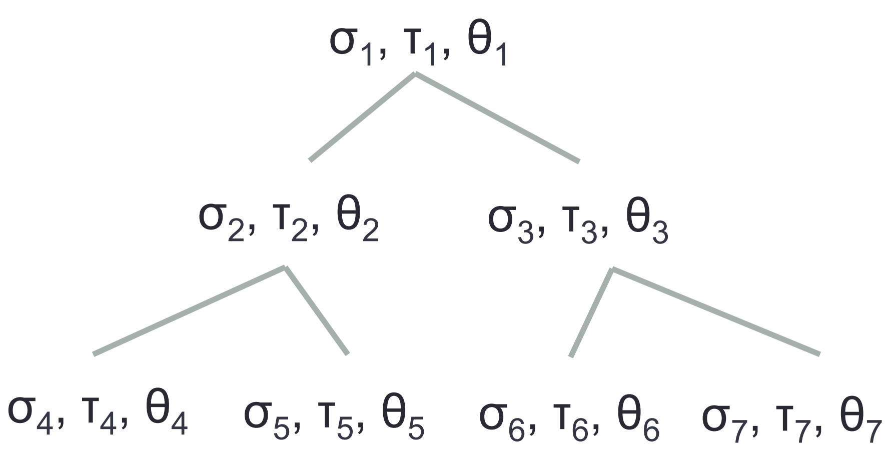
Tree based coordinates

- System is defined in a nested form like a tree.



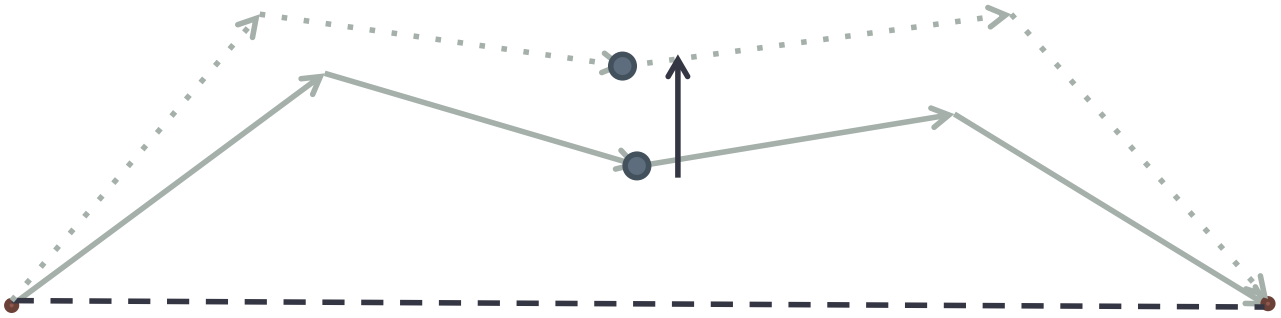


$$\tau = \ln \frac{d_1}{d_2}$$



Tree based coordinates

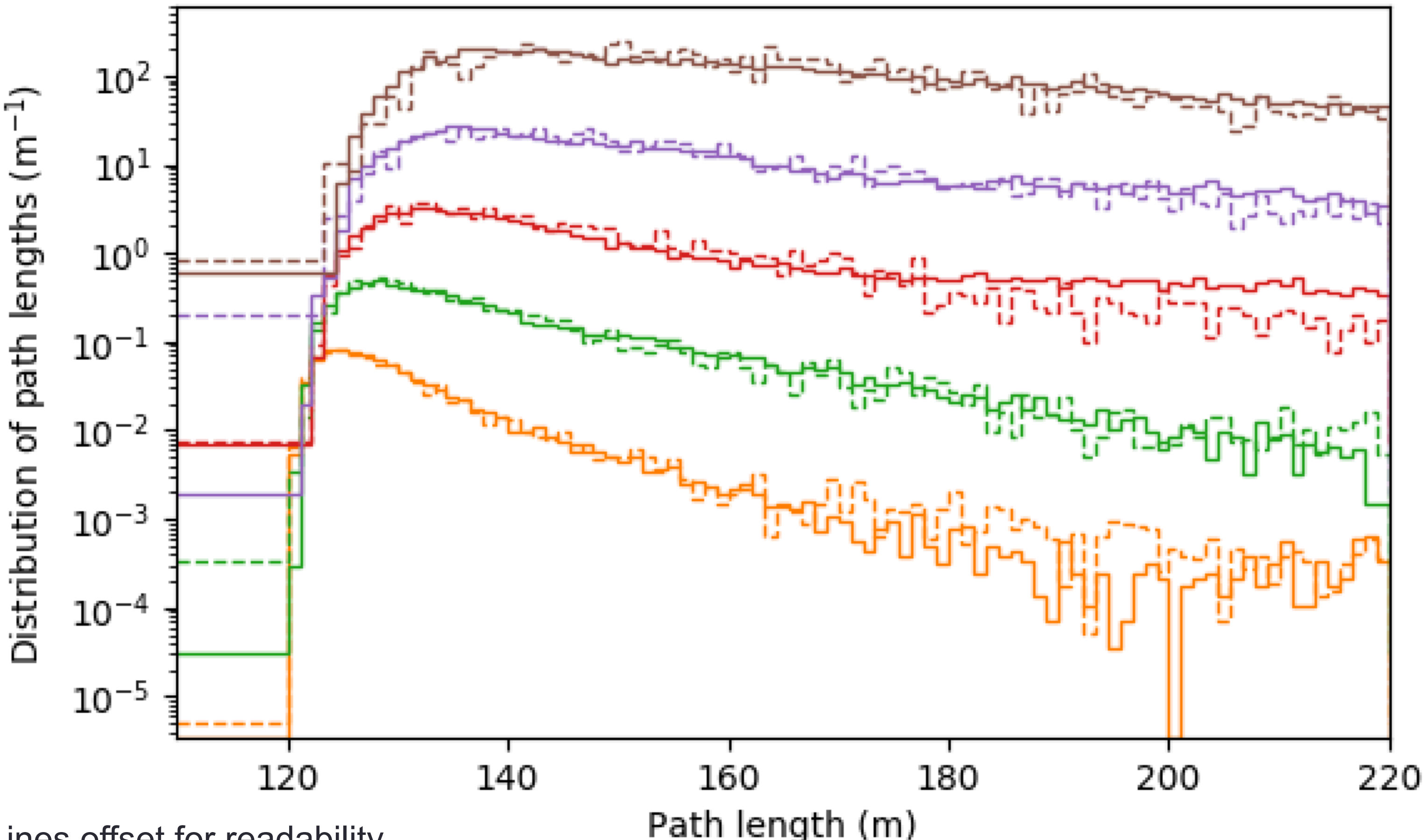
- Specified in terms of only dimensionless quantities, this system has a natural length scale independence.
- Also has a nice side-effect of correlated movements in the vertices.



- Sampling happens in this coordinate space, so an appropriate Jacobian factor is also needed.

Path length distribution

- Acceptance rate ~ 20%



b = 0.5 m^{-1}

b = 0.4 m^{-1}

b = 0.3 m^{-1}

b = 0.2 m^{-1}

b = 0.1 m^{-1}

Detector B

Lines offset for readability

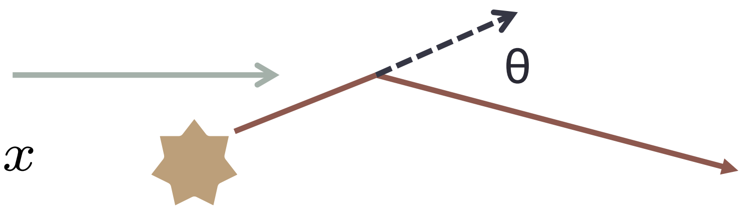
Ray tracing

- IceCube uses ray tracing to simulate light.
  - Equivalent to solving the equations of motion for photons.

- Light ray is propagated a random distance
- Direction is changed by a random amount according to the angular scattering distribution: p(cos θ)
- Ray thrown out (or re-weighted) according to the absorption probability: $e^{-a\,x}$

Length of propagation drawn from

$$p(x) = be^{-bx}$$



- IceCube generates millions of rays for each one that finds its way to a DOM.