SIMULATING LIGHT IN LARGE VOLUME DETECTORS USING METROPOLIS LIGHT TRANSPORT

Gabriel Collin

MIT INSTITUTE FOR DATA, SYSTEMS, AND SOCIETY

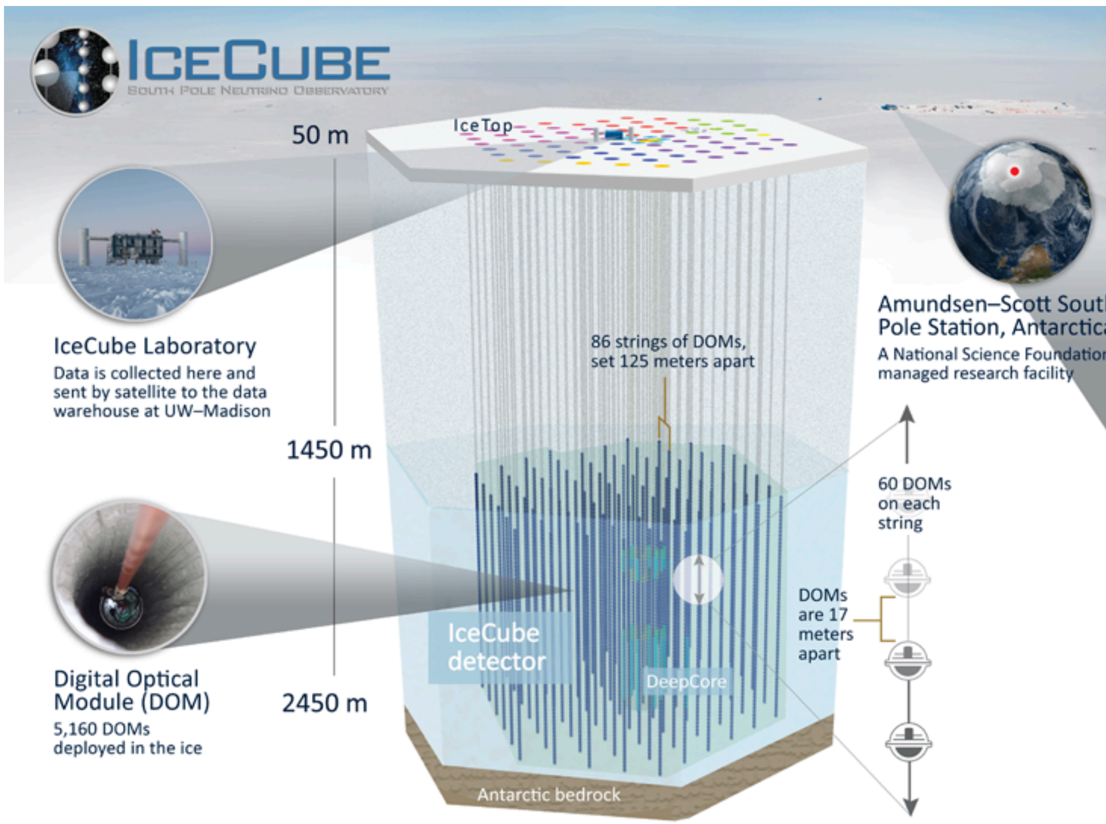
MIT Physics

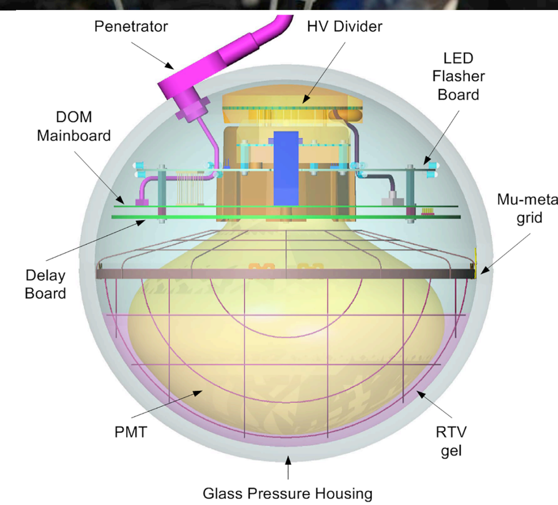
Outline

- Motivation
- Posing the problem as a path integral
- Sampling from the path integrand
    - Trans-dimensional sampling
- Results and performance comparison

IceCube

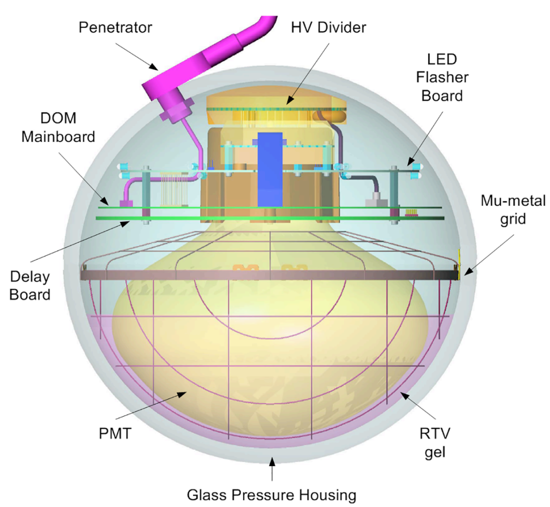
- Gigaton neutrino detector located at the south pole.

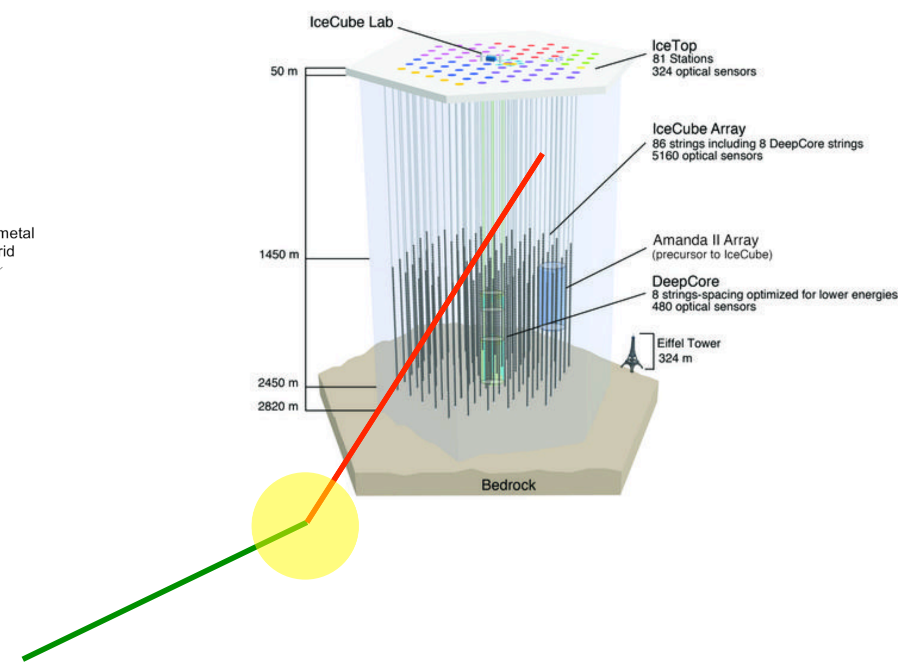




IceCube

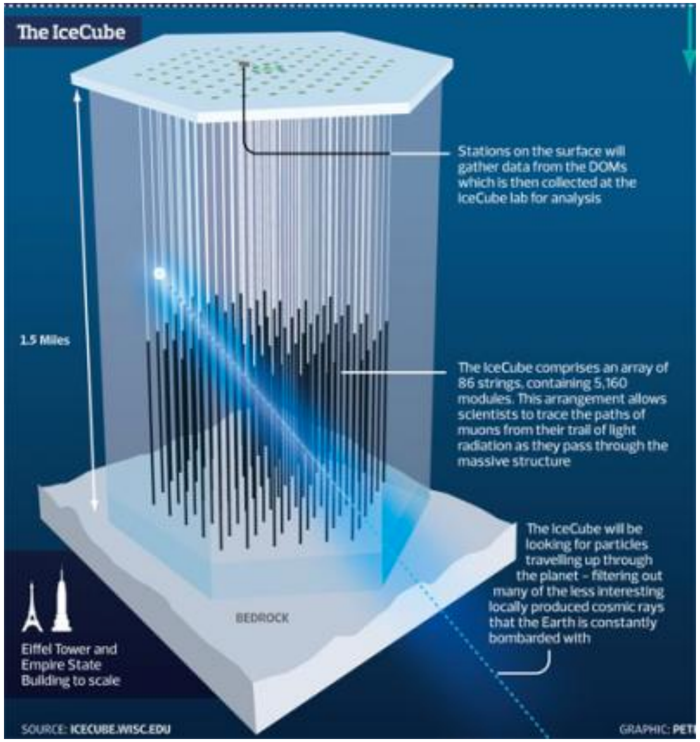
- Muon neutrinos interact with the surrounding ice/rock and produce muons that travel through the detector.
  - The muon travels a large distance before it decays.
  - Produces Cherenkov light as it travels.
  - Light is detected by Digital Optical Modules (DOMs)

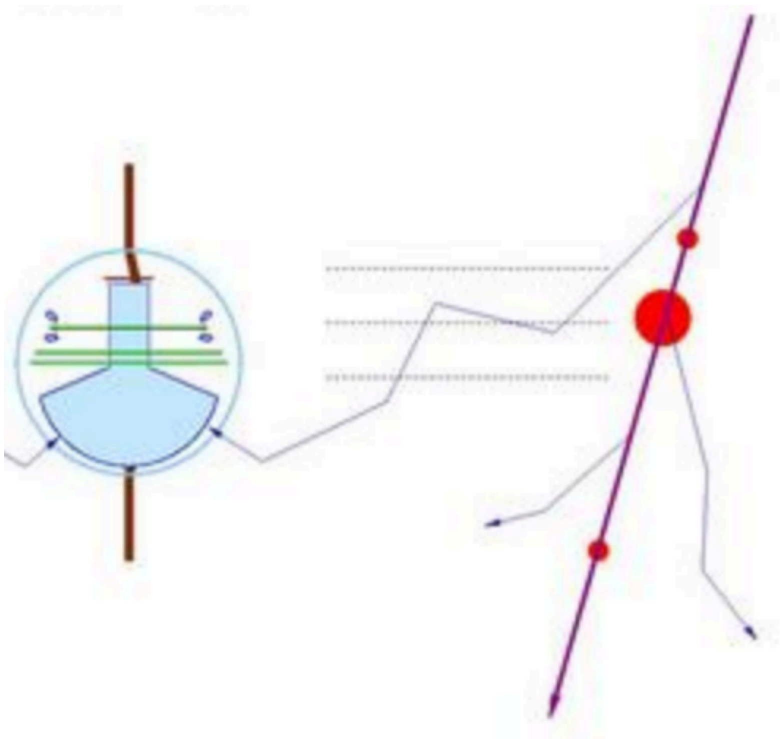




IceCube

- As light travels through the ice, it can be scattered or absorbed.





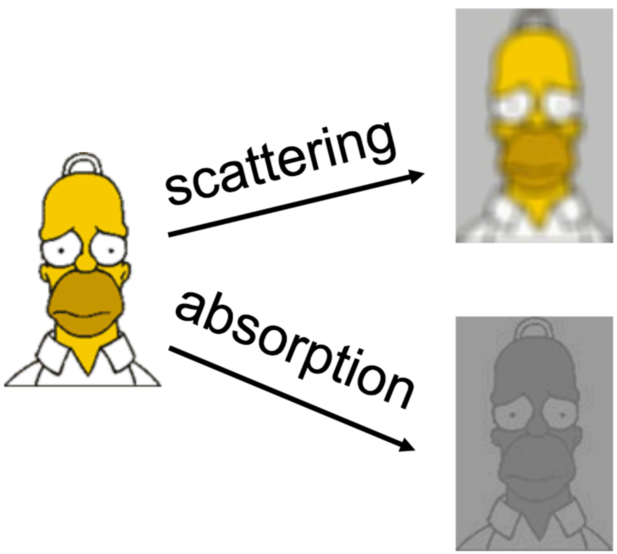
Example by Dmitry Chirkin

Light in ice

- Absorption is governed by the absorption parameter a.

$$I(x) = I_0 e^{-ax}$$

- Scattering is governed by
  - The scattering parameter b $\quad P(x|\text{no scatter}) = e^{-bx}$
  - The angular scattering distribution. $\quad p(\cos\Delta\theta)$



Source is blurred

Poorer angular resolution

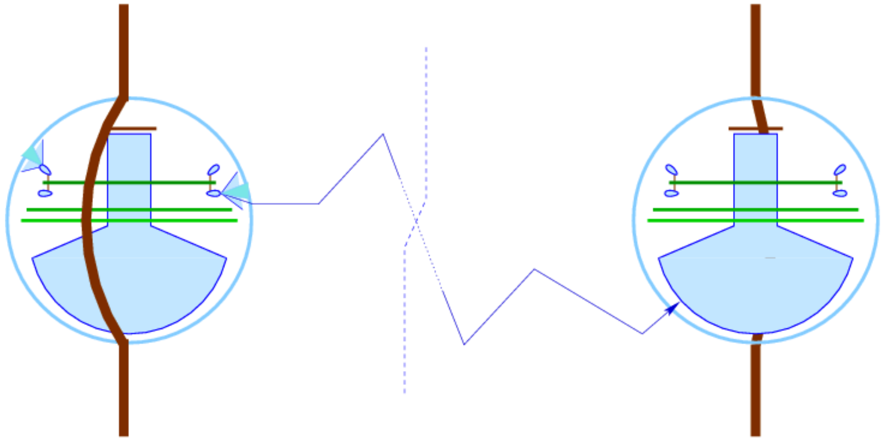
Source is dimmer

Poorer energy resolution

Example by Dmitry Chirkin

Calibrating IceCube

- Need to figure out how much absorption and scattering is in the ice.
  - DOMs have built in LEDs.
  - LED injects known amount of light into the ice.
  - Measure the light that makes it to other DOMs, and reconstruct the scattering and absorption.



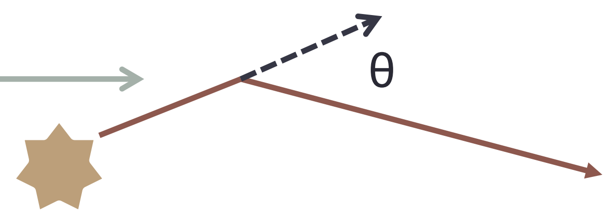
Ray tracing

- 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

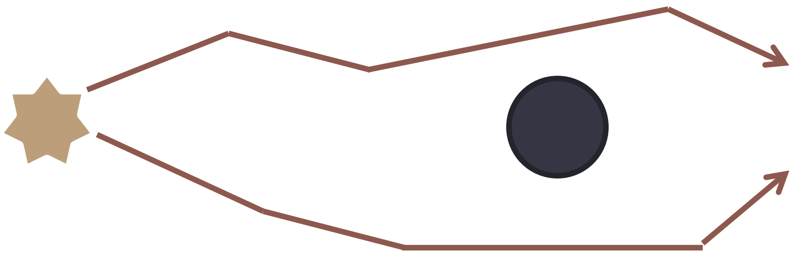
$$p(x) = be^{-bx}$$



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

Motivation

- Currently, IceCube uses ray tracing to propagate light in the ice.
- However, most rays never reach a DOM.
  - Collecting a significant number of rays on a far away DOM means simulating a huge number of rays that get lost somewhere in the ice.



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

Light propagation

- The fundamental problem is that the interesting paths are highly constrained.
- Both the starting and ending points have to be in ~10cm regions across distances of ~100m.
- Is there another way to approach this?

Outline

- Motivation
- Posing the problem as a path integral
- Sampling from the path integrand
  - Trans-dimensional sampling
- Results and performance comparison

Path integration

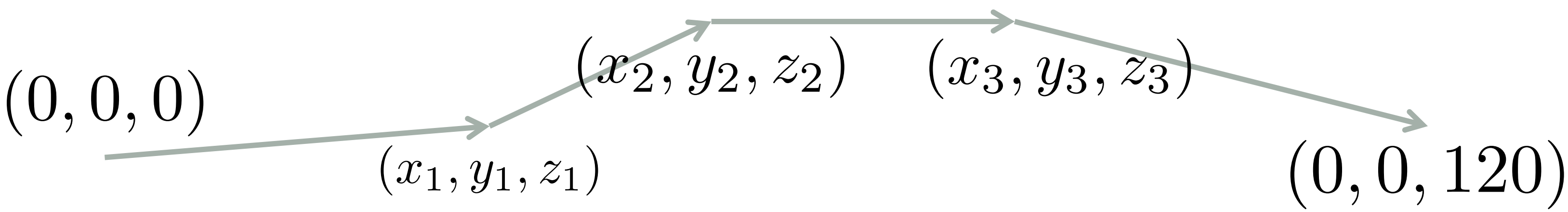
- The start and end locations of the ray can be constrained if the problem is specified in terms of a classical path integral.

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

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

- The scattering parameter, b, in IceCube is ~0.3 m^{-1}.
  - Over 120 meters, we expect at least 40 scatters.
  - Paths are thus 120 dimensional or more.

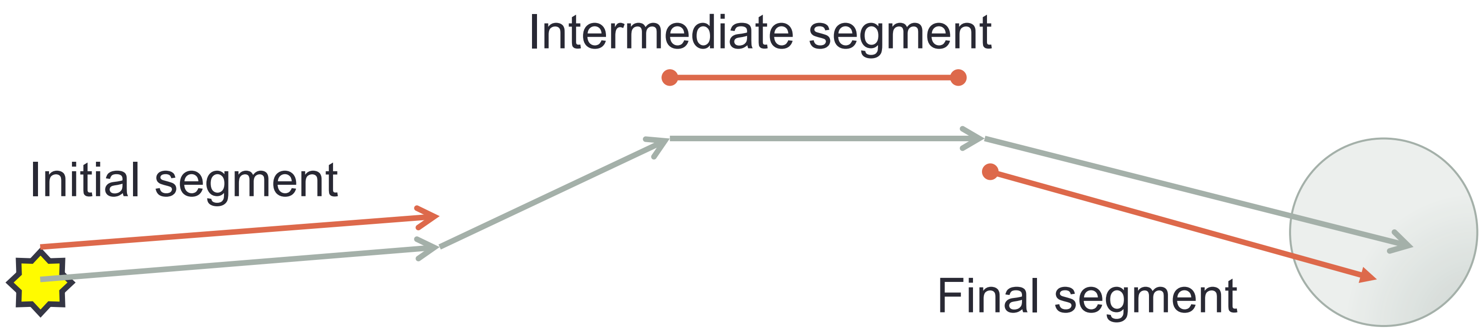
- Numerically integrating over 120 dimensions is not possible with standard techniques.
  - However, information can still be extracted about the light propagation by framing the integrand as a probability distribution:

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

  - This distribution can then be sampled.

Probability distribution

- The probability distribution has three main parts:
  - A factor for the initial vertex.
  - A factor for each intermediate vertex.
  - A factor for the last vertex, 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.

$$p(\vec{r}_0) = b(\vec{r}_0)e^{-\tau_0}\varepsilon(\hat{r}_0)$$

  - Here, emission probability distribution chosen to be a von Mises-Fisher distribution:

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

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

Probability distribution

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

$$p(\vec{r}_i | \vec{r}_{i-1}, \ldots) = \underline{b(\vec{x}_i) e^{-\tau_i}} \sigma \underline{(\cos \Delta\theta_i)}$$

Exponential distribution for scattering

Angular scattering distribution

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

$$\vec{x}_i = \sum_{j=0}^{i} \vec{r}_j$$

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.

$$p(\hat{r}_f|\vec{r}_{f-1},\ldots) = e^{-\tau_f}\rho(\vec{x}_f)(\hat{r}_f \cdot \hat{n})\sigma(\cos\Delta\theta_i)$$

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:

$$p(\{\vec{r}_i\}) = p(\vec{r}_0) \left[\prod_{i=1}^{n-2} p(\vec{r}_i | \vec{r}_{i-1}, \ldots) \right] p(\hat{r}_{n-1} | \vec{r}_{n-2}, \ldots)$$

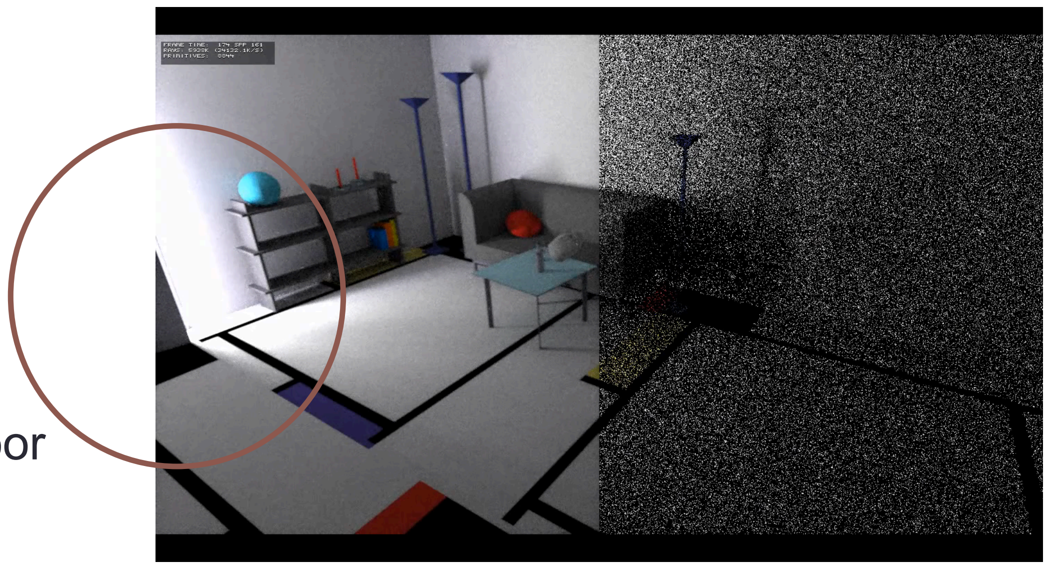
- This PDF is analytic.
  - But the CDF is not.
  - Inversion sampling cannot be used to draw samples.
  - Instead, we can use a Markov Chain Monte-Carlo.

Outline

- Motivation
- Posing the problem as a path integral
- Sampling from the path integrand
    - Trans-dimensional sampling
- Results and performance comparison

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.

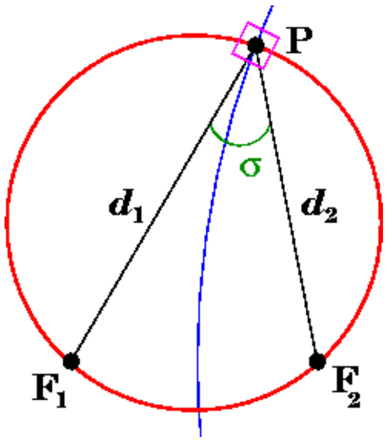
http://raytracey.blogspot.com.es/2010/12/real-time-metropolis-light-transport-on.html

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.

Choice of coordinates

- One solution is to de-correlate through a good choice of coordinates.
- Partial de-correlation can be achieved with bi-spherical coordinates.
- Has two fixed focii (much like our paths have a fixed start and end points).
- Angle between vertices is naturally one of the coordinates.

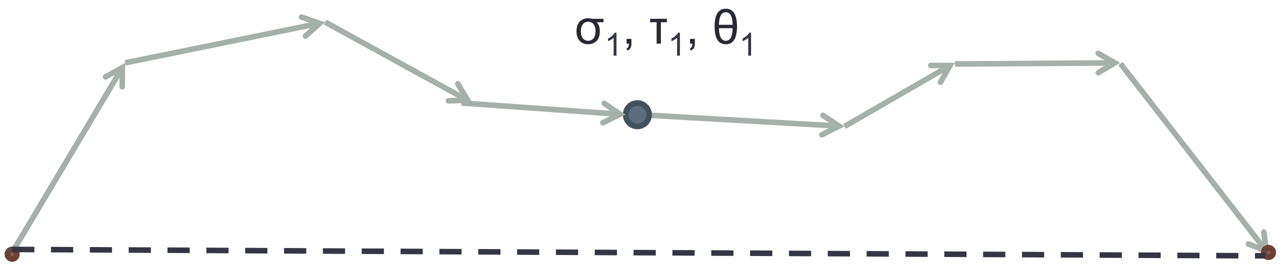


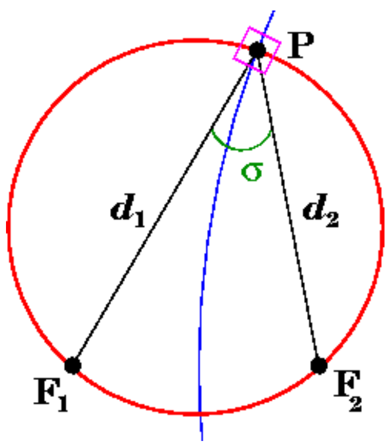
$$\tau = \ln \frac{d_1}{d_2}$$

Bi-polar coordinates. Bi-spherical coordinates include an additional rotation around the $F_1 - F_2$ axis.

Tree based coordinates

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



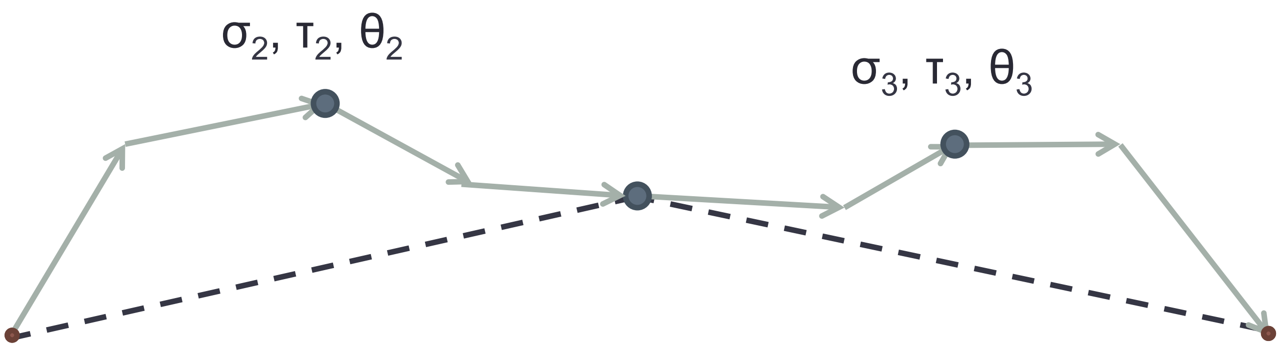


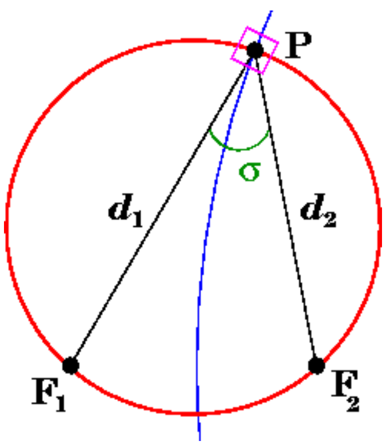
$$\tau = \ln \frac{d_1}{d_2}$$

$\sigma_1, \tau_1, \theta_1$

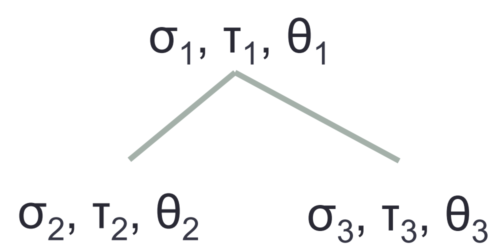
Tree based coordinates

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



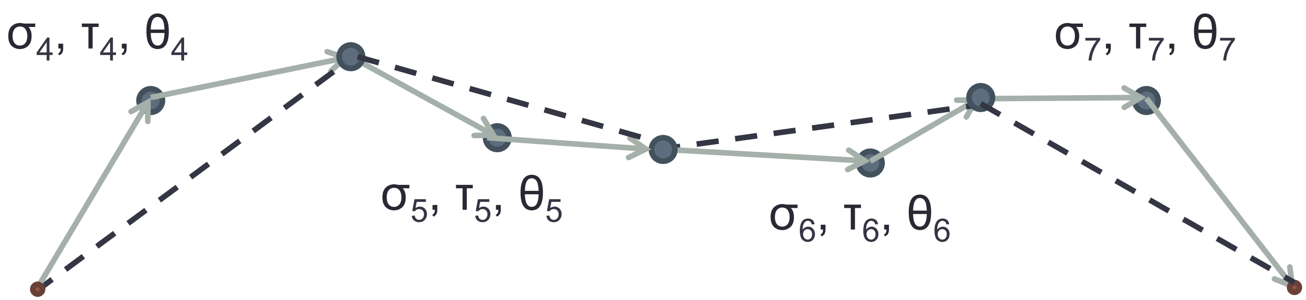


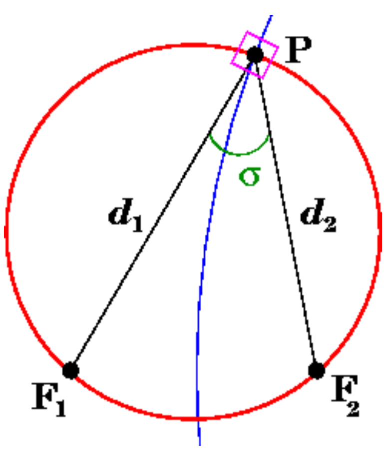
$$\tau = \ln \frac{d_1}{d_2}$$



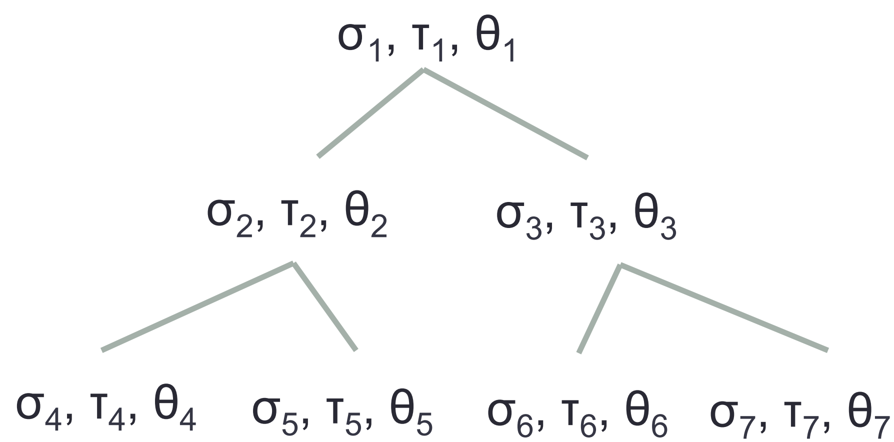
Tree based coordinates

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



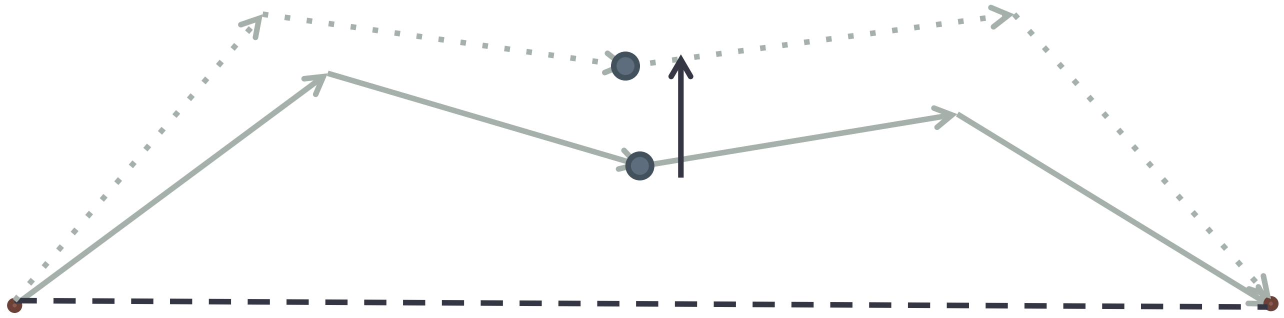


$$\tau = \ln \frac{d_1}{d_2}$$



Tree based coordinates

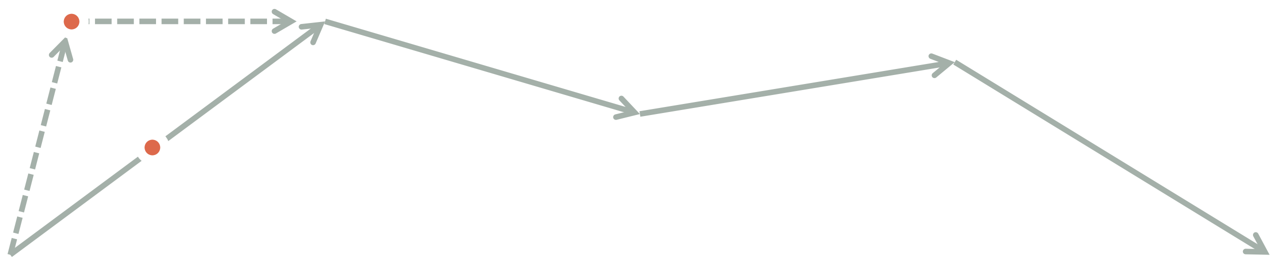
- Specified in terms of only dimensionless quantities, this system has a natural length scale independence.
- Also has a nice side-effect of correlated movements in the vertices.



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

Reversible jump MCMC

- However, 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.

Outline

- Motivation
- Posing the problem as a path integral
- Sampling from the path integrand
  - Trans-dimensional sampling
- Results and performance comparison

Intro to reversible jump

- Basic idea is similar to a standard MCMC.
- Given a sample in one vector space:
  - Propose a new sample in another vector space.
  - Calculate probability at each sample.
  - Accept/Reject based on the ratio of probabilities.

- But how do we compare the probabilities?
  - They are defined in different vector spaces,
  - Which can have different dimensions,
  - So a simple ratio doesn't make sense.

Intro to reversible jump

- Assume one vector space is smaller than the other
    - (If they are the same dimension, the generalised form still works).
  - Key behind reversible jump is to 'tack on' some extra probability distribution to the smaller vector space.
  - This pads out the dimensions so they are equal.

- We have 1 and 2 dimensional distributions:

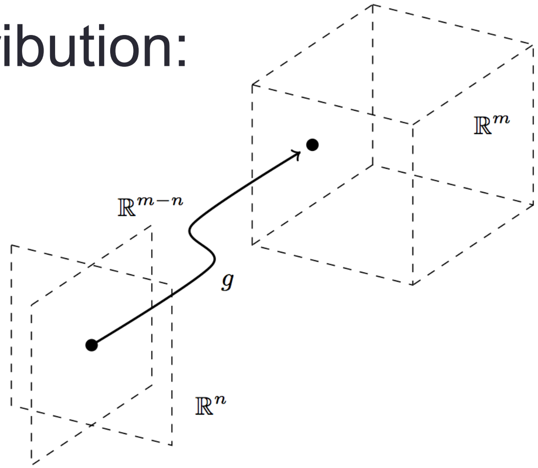
$$p_0 : \mathbb{R}^1 \to \mathbb{R} \qquad p_1 : \mathbb{R}^2 \to \mathbb{R}$$

- To match, we need a 2-1=1 dimensional distribution:

$$Q : \mathbb{R}^1 \to \mathbb{R}$$

- The proposal function is then

$$g : \mathbb{R}^1 \otimes \mathbb{R}^1 \to \mathbb{R}^2$$



Intro to reversible jump

- Then the accept/reject is based on the following ratio:

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

Proposal rates

Jacobian of proposal function.

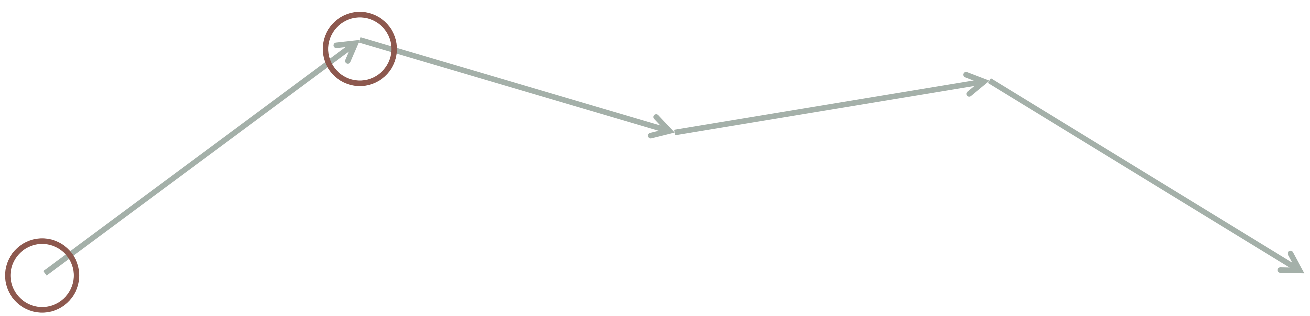
Padded probability distribution

- 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*.

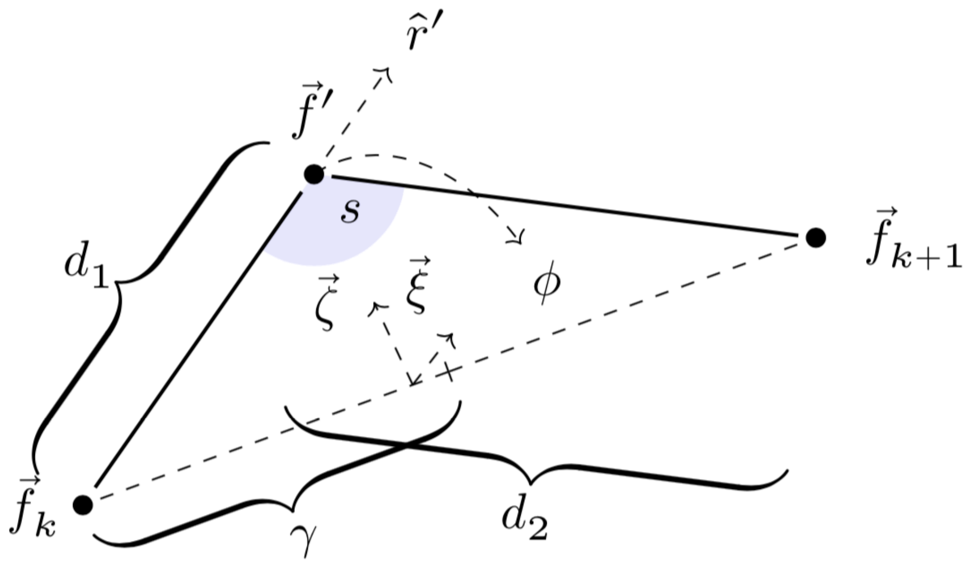
- g selects a pair of vertices.



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

Reversible jump for light propagation

- New vertex inserted using bi-spherical coordinates.



- σ, τ, θ are draw from a q distribution chosen to match the curvature of p(x) as closely as possible.

Outline

- Motivation
- Posing the problem as a path integral
- Sampling from the path integrand
  - Trans-dimensional sampling
- Results and performance comparison

Path length distribution

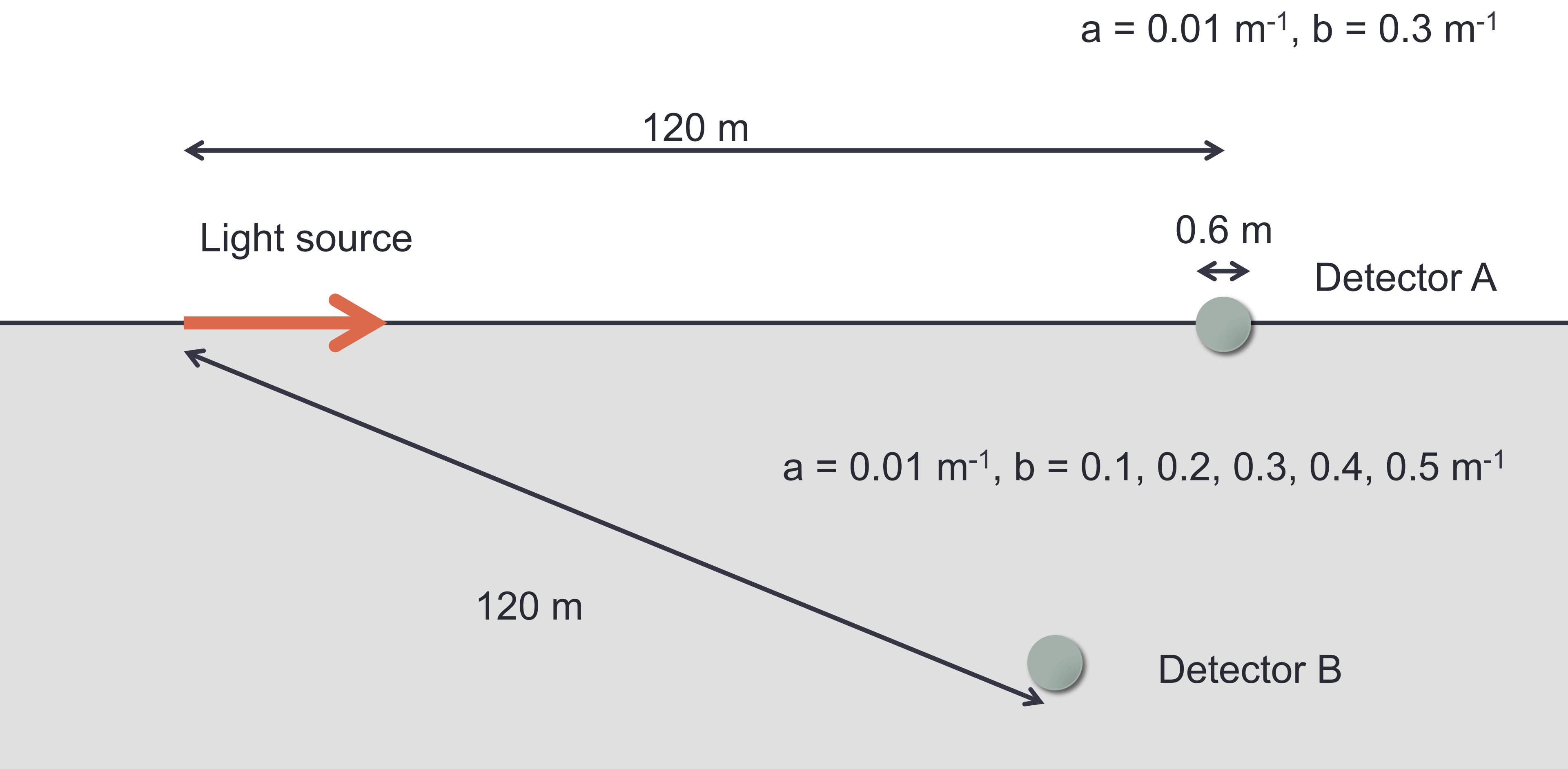
- From the samples created by the MCMC, the probability distribution for path length can be easily extracted.
  - IceCube measures photon arrival time, which is directly related to path length.
  - $P(L < X)$ = fraction of samples where the length of the path is less than X.

- 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.

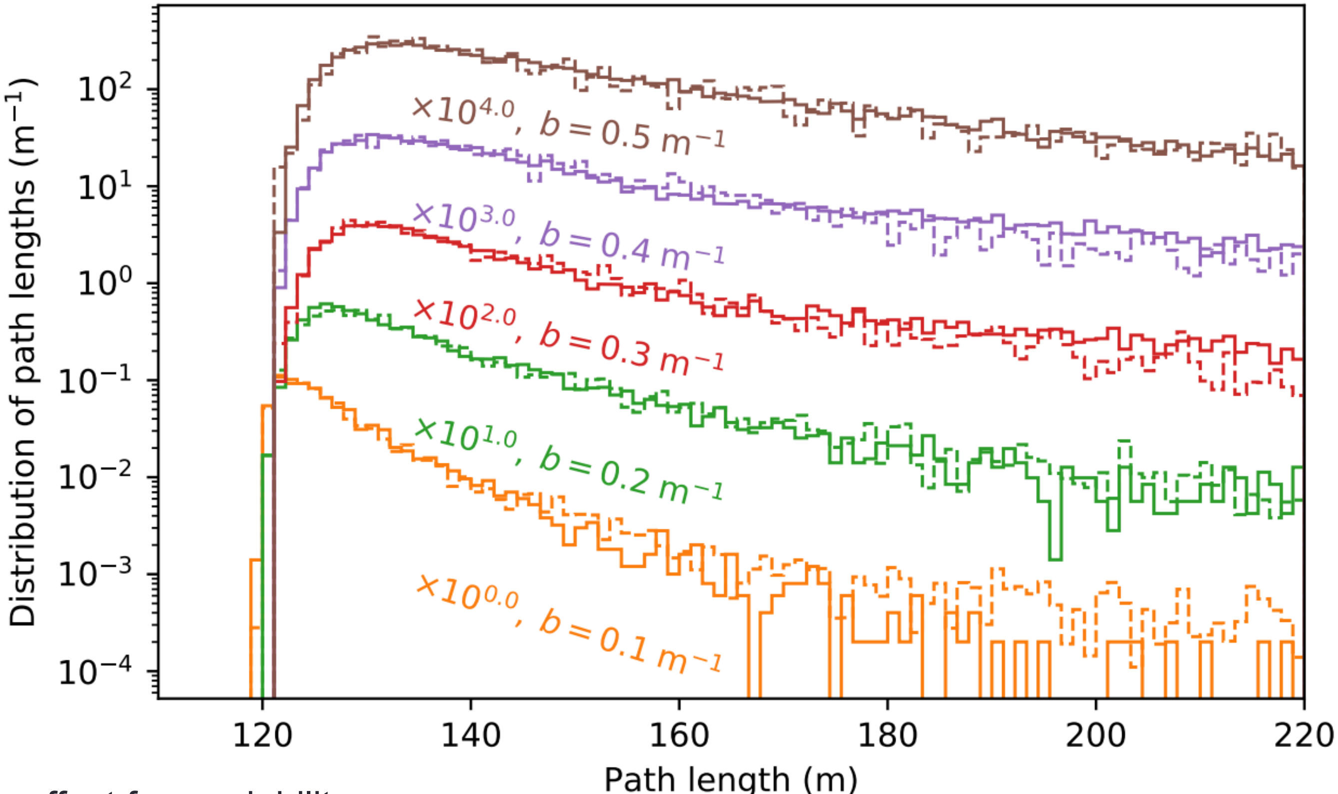
Synthetic test case

- 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.

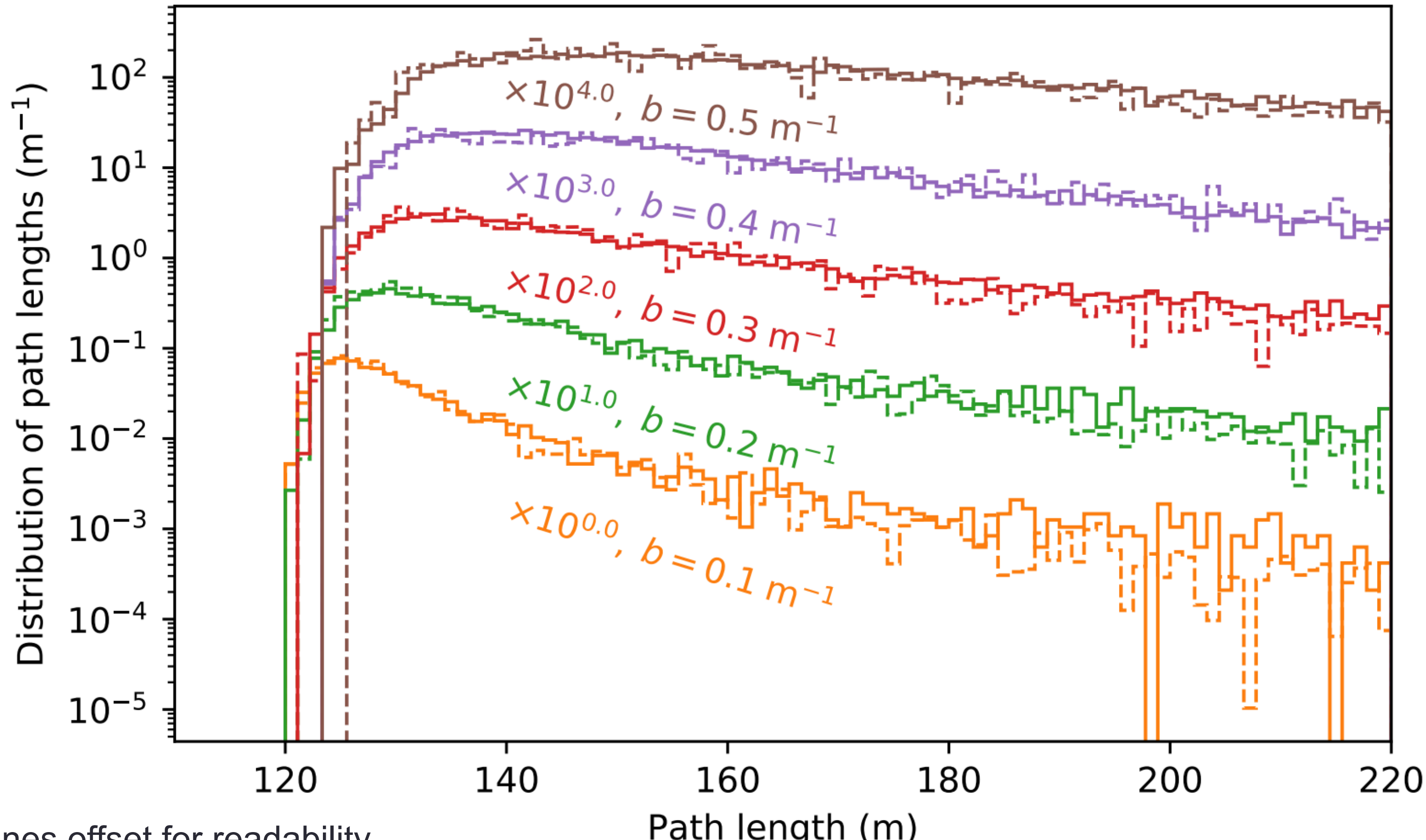


Detector A

Lines offset for readability

Path length distribution

- Acceptance rate ~ 20%



Detector B

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

Relative light yield

- In principle, the relative light yield between the two detectors can also be calculated.
  - Absolute light yield is much more difficult.

- Relative light yield is given by the ratio of normalisations for each detector.
  - This is equivalent to finding a Bayes factor in Bayesian inference.

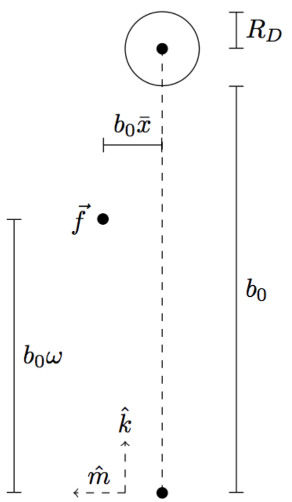
- We can use the geometric estimator:

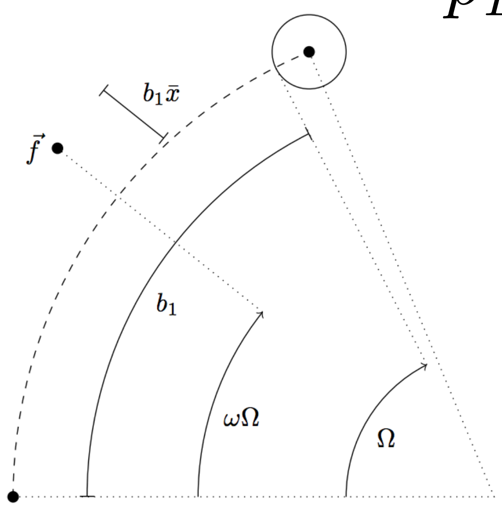
$$\mathcal{B} = \frac{\mathbb{E}_A[\sqrt{p_B(x)/p_A(x)}]}{\mathbb{E}_B[\sqrt{p_A(x)/p_B(x)}]}$$

Relative light yield

- Computing $\mathbb{E}_A[\sqrt{p_B(x)/p_A(x)}]$ requires both distributions to be in the same coordinate system.
  - In fact, we actually have identical distributions in different coordinate systems.
- If we transform paths from one detector, to corresponding paths in the other detector, we can have two distributions in the same coordinate system.

$$p_B(x) = p_A(T(x))dT/dx$$





Relative light yield

- To estimate the variance on the light yield
  - Ray tracer and path sampler were run four times.
  - Standard deviation in parentheses.

b	Ray tracer	Path sampler
0.1 m^{-1}	0.67(3)	0.64(7)
0.2 m^{-1}	0.79(2)	0.77(1)
0.3 m^{-1}	0.73(2)	0.76(7)
0.4 m^{-1}	0.68(2)	0.64(4)
0.5 m^{-1}	0.59(2)	0.50(6)

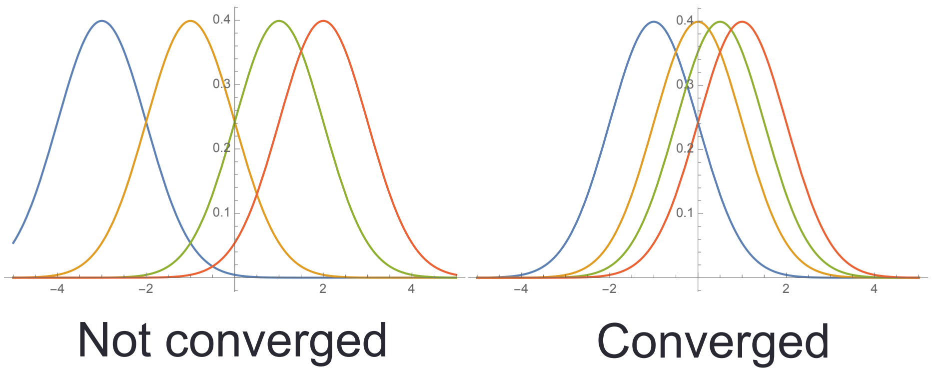
  - Path sampler agrees to within the standard deviation.
  - However, it is generally varies more than the ray tracer.

Convergence diagnostics

- Convergence can be estimated by running multiple MCMC chains and comparing their outputs.
  - If the chains have converged, the outputs should look similar.

- Can use a metric called the “potential scale reduction factor”.
  - Compares the variance of a variable within a chain to the variance between chains.

- It is difficult to compute this for all coordinates, as the dimensionality of the chain is always changing.



Convergence diagnostics

- Instead, the potential scale reduction factor can be computed for an observable of the path.
  - Eg: the total path length.

b	R	Acceptance
0.1 m^{-1}	1.11	30%
0.2 m^{-1}	1.08	23%
0.3 m^{-1}	1.19	20%
0.4 m^{-1}	1.12	20%
0.5 m^{-1}	1.02	19%

- For robust, automated usage, more diagnostics will be required.

Application to IceCube

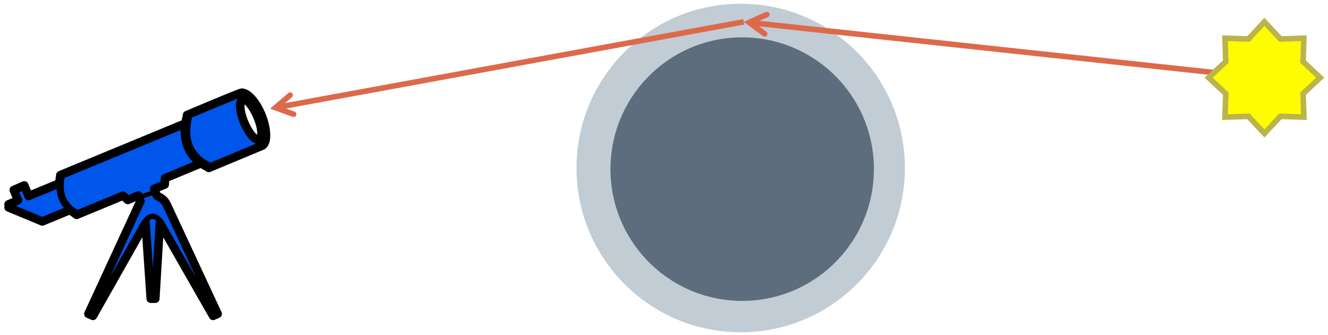
- IceCube DOM is half the radius of the synthetic test case used here.
    - An additional factor 4 performance improvement relative to a ray tracer.
- The IceCube ray tracer is implemented on a GPU.
    - Gives 100x performance increase compared to CPU.
    - Path sampling will also need a GPU implementation.
        - However, MCMC based methods are less amenable to parallelisation.

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.

- Relative light yield between detectors is easier if they share geometry.
  - Eg: Yield between pixels on a CCD is considerably easier.
    - They are contiguous and share a plane.
    - Does not require a coordinate transformation.

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 1000 times 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.

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

BACKUP

Angular scattering distribution

- Distribution

$$\sigma(\cos\psi) = f_{\mathrm{SL}} p_{\mathrm{SL}}(\cos\psi) + (1 - f_{\mathrm{SL}}) p_{\mathrm{HG}}(\cos\psi),$$

- Simplified Liu:

$$p_{\mathrm{SL}}(\cos\theta) = \frac{1}{2}\frac{1+g}{1-g}\left[\frac{1+\cos\theta}{2}\right]^{\frac{2g}{1-g}}$$

- Henyey-Greenstein:

$$p_{\mathrm{HG}}(\cos\theta) = \frac{1}{2}\frac{1-g^2}{(1+g^2-2g\cos\theta)^{3/2}}$$

Detection probability

- Conditional detection probability:

$$\rho(\vec{f}_{n-1}) = \exp\left(3\cos\omega - \ln\cosh(2\cos\omega + 0.7) - 1\right)$$

$$\cos\omega = \hat{\rho}\cdot\frac{\vec{f}_{n-1} - \vec{\eta}}{|\vec{f}_{n-1} - \vec{\eta}|}$$

- Chosen to follow IceCube DOM angular response.

Jump distributions

$$q(s) = \frac{\beta e^{-\beta \cos s}}{2 \sinh \beta} \sin s,$$

$$q(t) = (2 + 2 \cosh t)^{-1},$$

$$q(\phi) = \frac{1}{2\pi},$$

Jump rates

$$p(n \to n+1, k) = \frac{\tau_b(k)}{\sum_{l=1}^{n-1} \tau_b(l)},$$

$$p(n \to n-1, k) = \frac{1}{n-2}.$$

Incident angle distribution

- For a smoothly varying b: