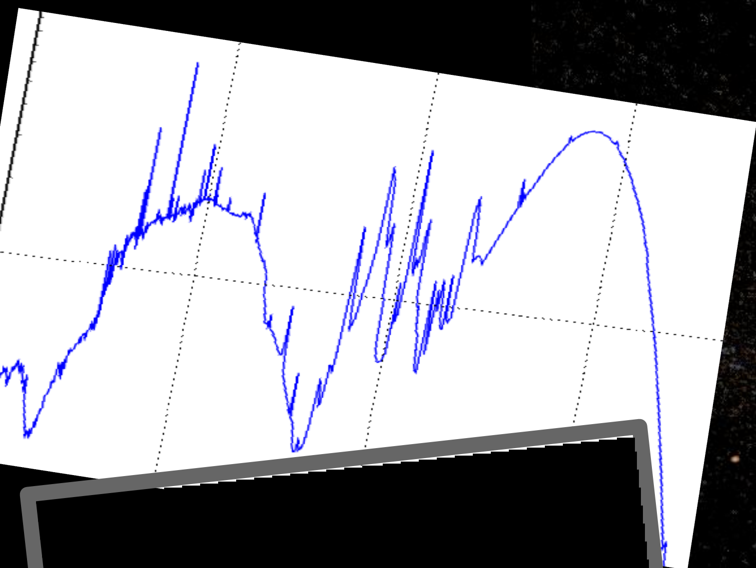
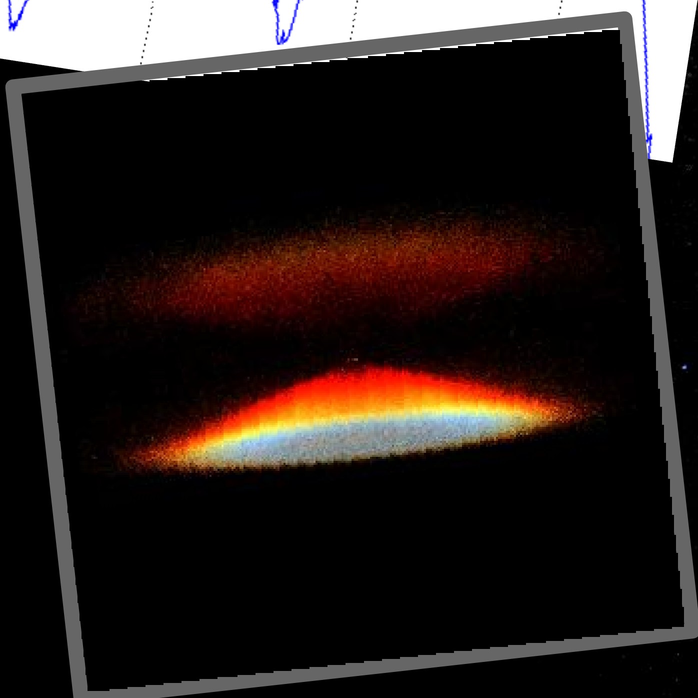
Polychromatic Monte Carlo with Sunrise

and some other ideas





Patrik Jonsson (UC Santa Cruz)
in collaboration with
Greg Novak (UCSC), Brent Groves (Leiden)
Joel Primack (UCSC),
Chris Henze (NASA ARC)

Overview

Polychromatic Monte Carlo with Sunrise

Metropolis Light Transport

Fast dust temperatures using GPUs

“Sunrise” Monte Carlo code

Described in Jonsson 2006, MNRAS

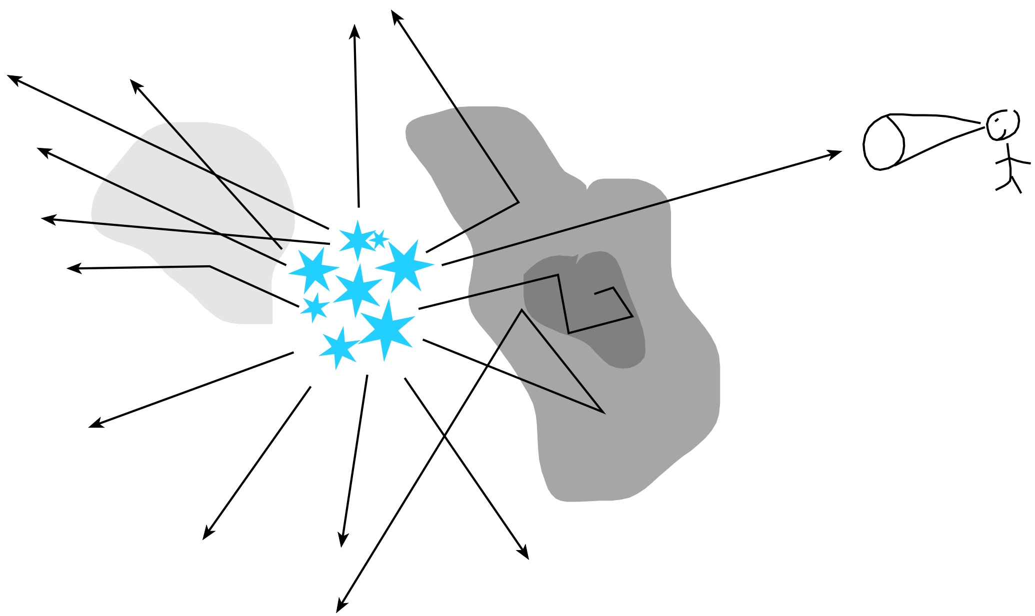
Free software (GPL), http://sunrise.familjenjonsson.org

Fully 3D adaptive-refinement grid used to describe geometry

Shared-memory parallel

Radiative transfer using “polychromatic” Monte Carlo method

Dust emission calculated through thermal equilibrium



“Polychromatic” Radiative Transfer

Most time spent traversing grid, ray tracing is expensive

With monochromatic rays, each wavelength must be traced separately, so wavelength resolution is expensive

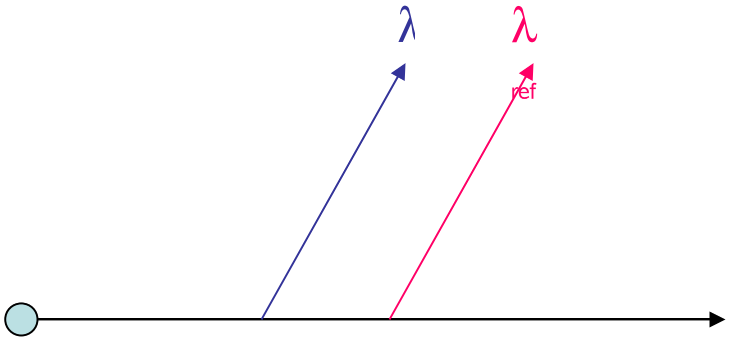
Can we combine all wavelengths into one ray?

(Done for direct radiation only in SKIRT, Baes et al 2005)

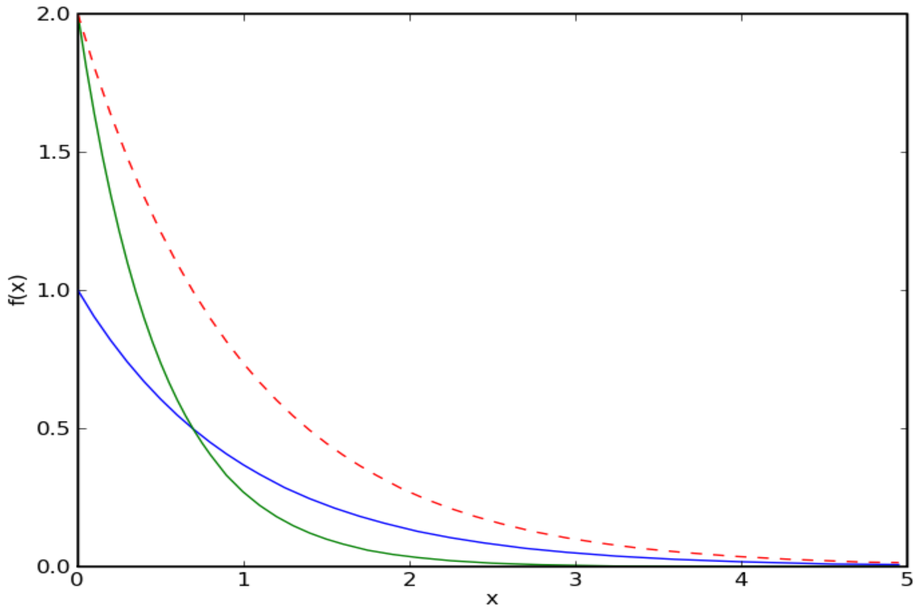
What’s the complication?

Mean free path varies with wavelength

Angular distribution of scattered light varies with wavelength



Biasing



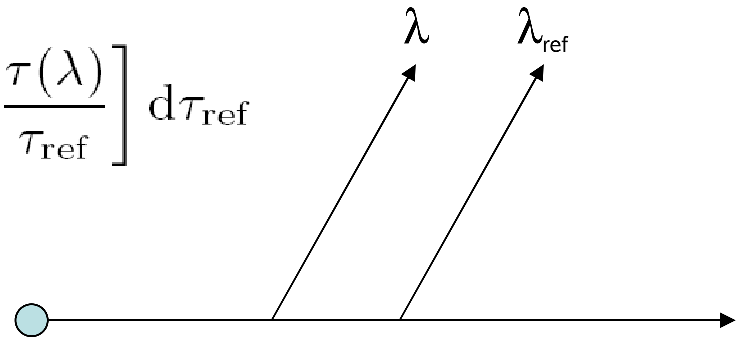
To sample the green distribution while drawing from the blue one, each sample drawn is weighted by the ratio of the distributions (red dashed line).

Not new idea, used by e.g. Yusef-Zadeh et al. (84) and Juvela (05).

Polychromatic biasing

When drawing a scattering distance, the probability can be expressed as

$$\mathrm{d}P\left[\tau(\lambda)\right] = e^{-\tau(\lambda)}\mathrm{d}\tau(\lambda) = e^{-\frac{\tau(\lambda)}{\tau_{\mathrm{ref}}}\tau_{\mathrm{ref}}}\left[\frac{\tau(\lambda)}{\tau_{\mathrm{ref}}}\right]\mathrm{d}\tau_{\mathrm{ref}}$$



Intensities are multiplied by weighting factor

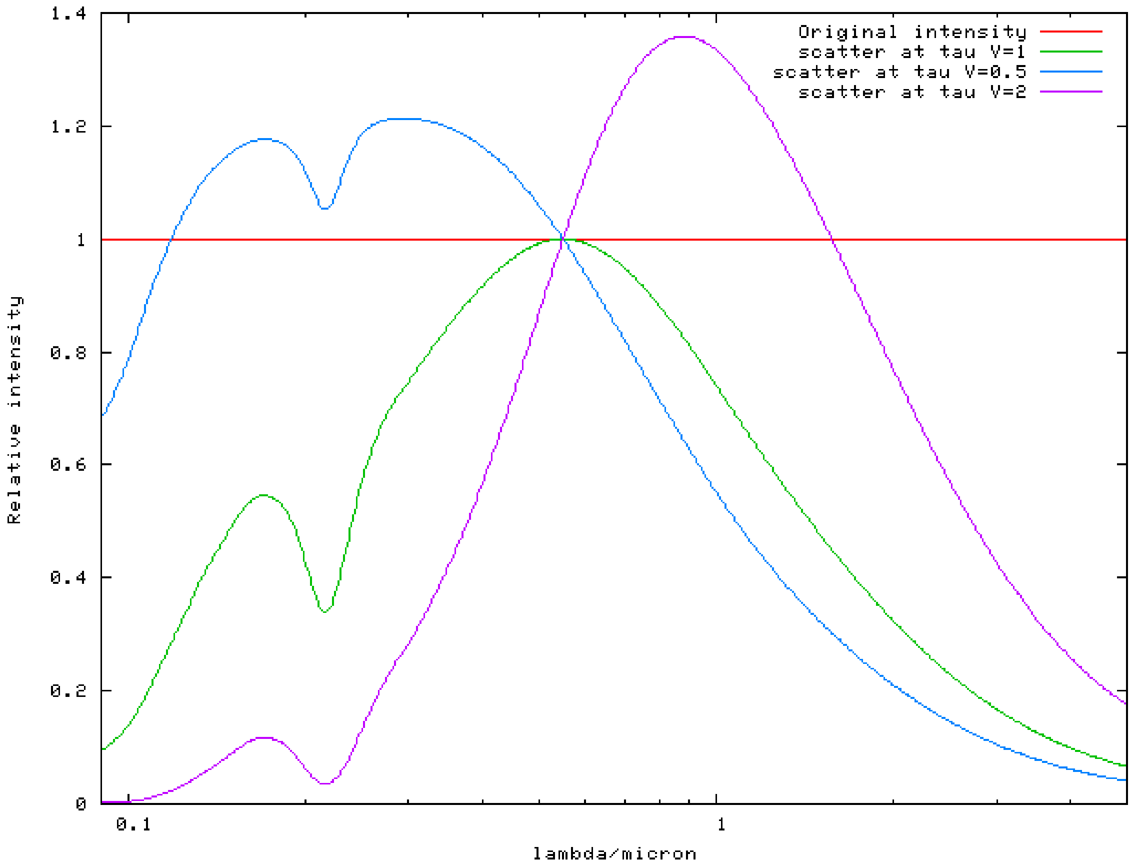
$$w_{\lambda} = \frac{P\left[\tau(\lambda)\right]}{P\left[\tau_{\mathrm{ref}}\right]} = e^{\tau_{\mathrm{ref}}-\tau(\lambda)}\left[\frac{\tau(\lambda)}{\tau_{\mathrm{ref}}}\right]$$

to compensate for the “incorrect” sampling of interaction lengths

Similar expressions used for scattering angles

Wavelengths that are more likely to interact at the sampled point have their intensities boosted.

Wavelengths that are less likely to interact at the sampled point have their intensities decreased.

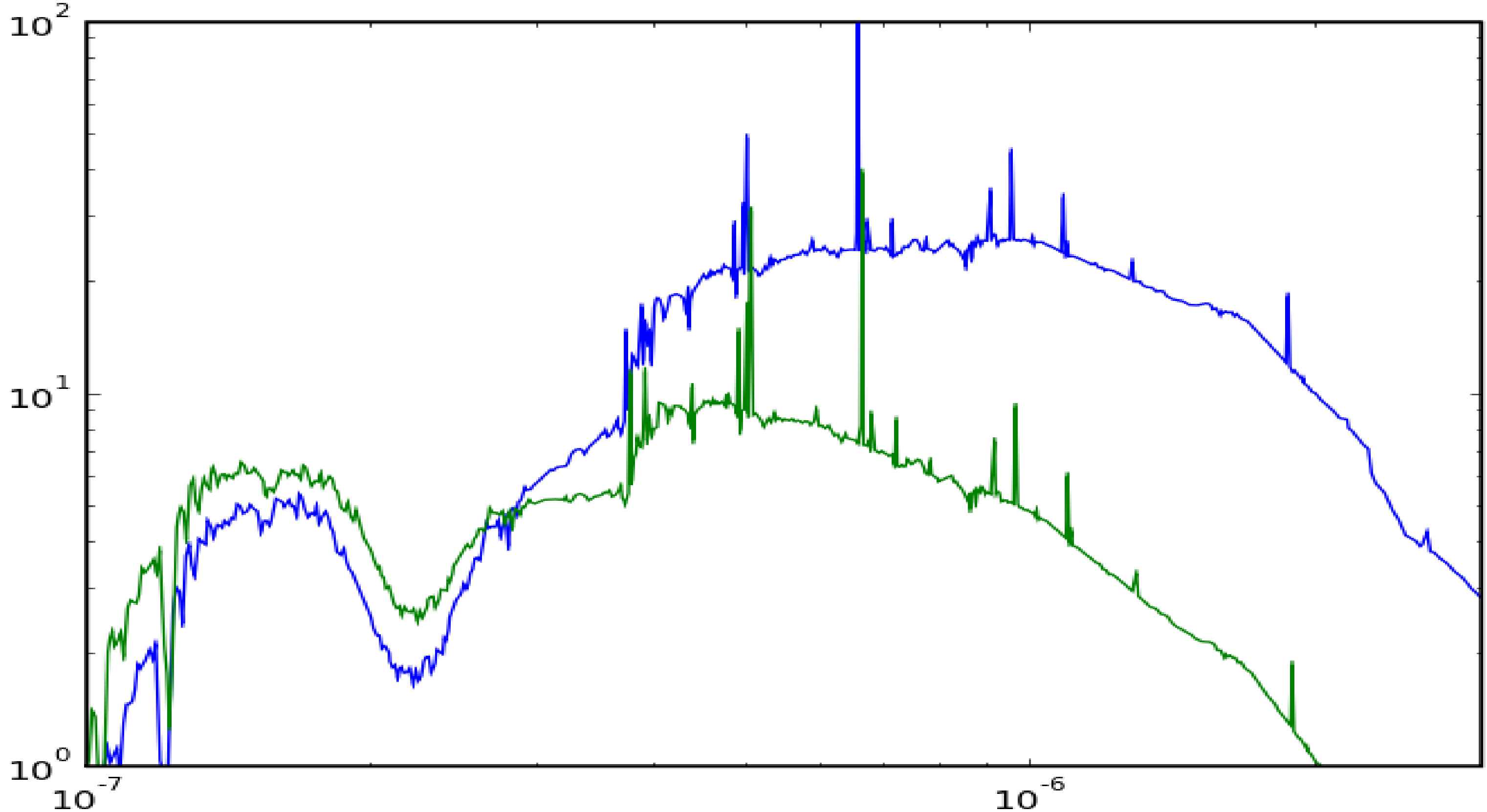


Advantages

High wavelength resolution possible

- ~50x more efficient for 500 wavelengths

Minimizes differential “color noise” between wavelengths.



Drawbacks

- For large optical depths, bias factor can get large as more flux is concentrated into fewer rays
  - In severe cases can lead to incorrect estimate (black swan effect)
  - Moderated by ray splitting
  - Scatter rays at several locations

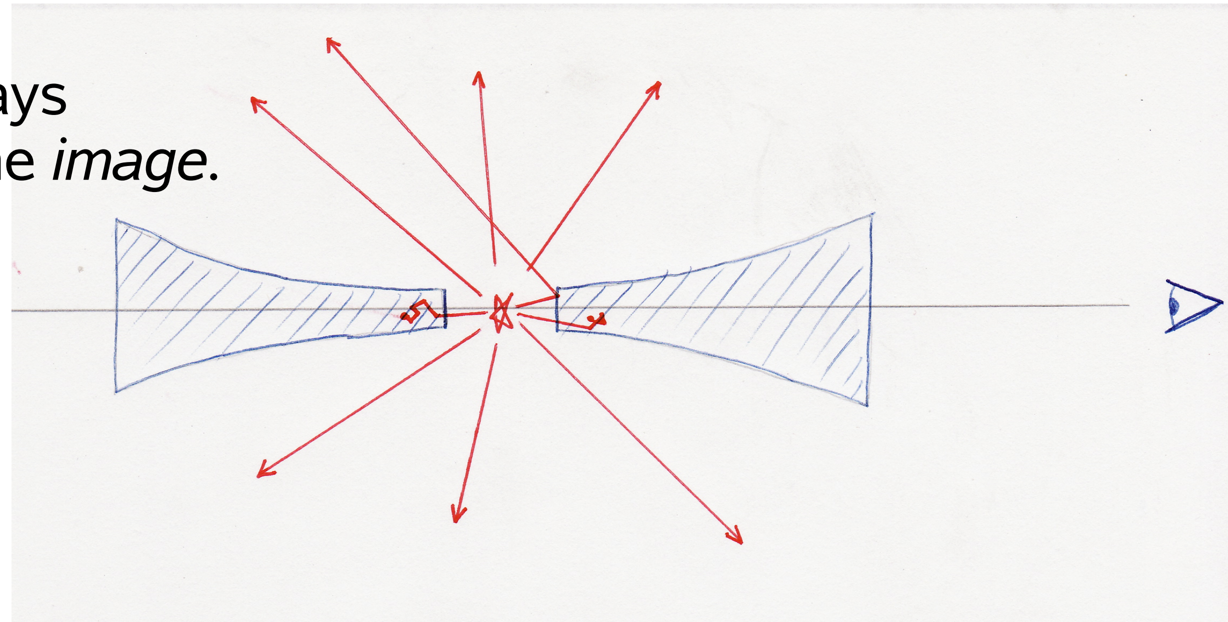
Overview

Polychromatic Monte Carlo with Sunrise

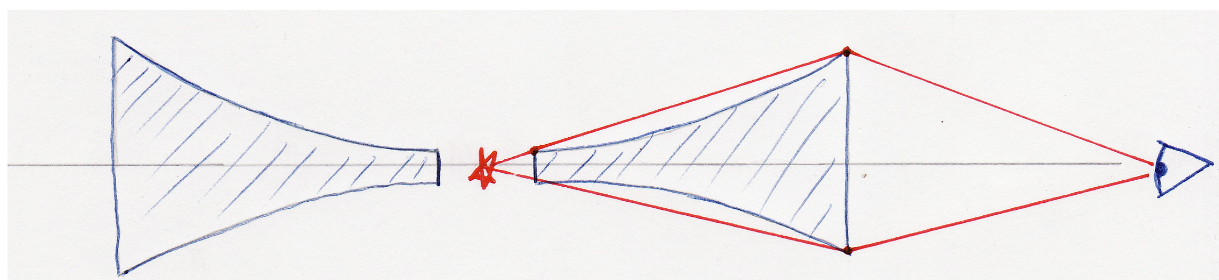
Metropolis Light Transport

Fast dust temperatures using GPUs

Problem: In a (straight)forward Monte Carlo calculation, we follow the *energy*.
But (most) calculated rays may not contribute to the *image*.



What we want is spend effort on rays that DO contribute:



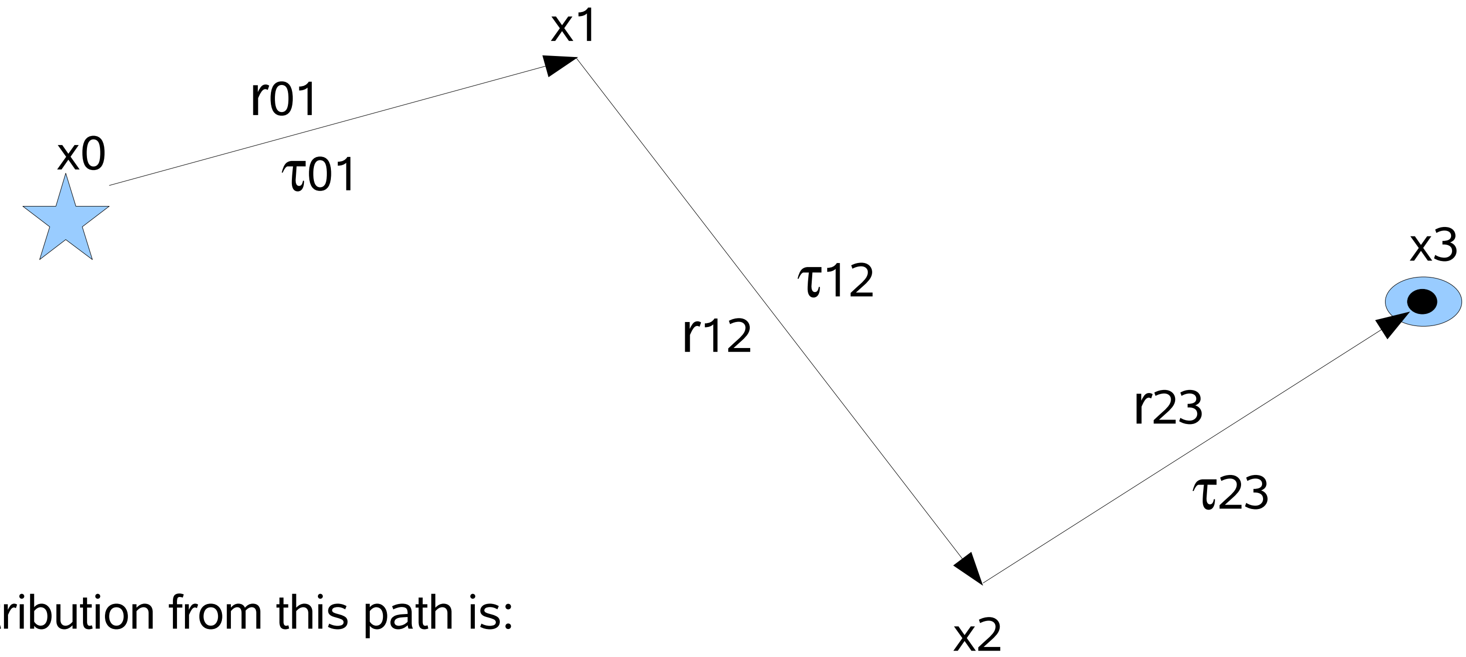
The Metropolis algorithm allows us to do exactly this

Metropolis Light Transport

Uses the Metropolis-Hastings algorithm to sample the distribution of ray paths that contribute to the image.

Pioneered by Veach and Guibas in computer graphics (SIGGRAPH97).

Need only be able to evaluate the contribution from a sample (ray path), and a method for mutating the sample.



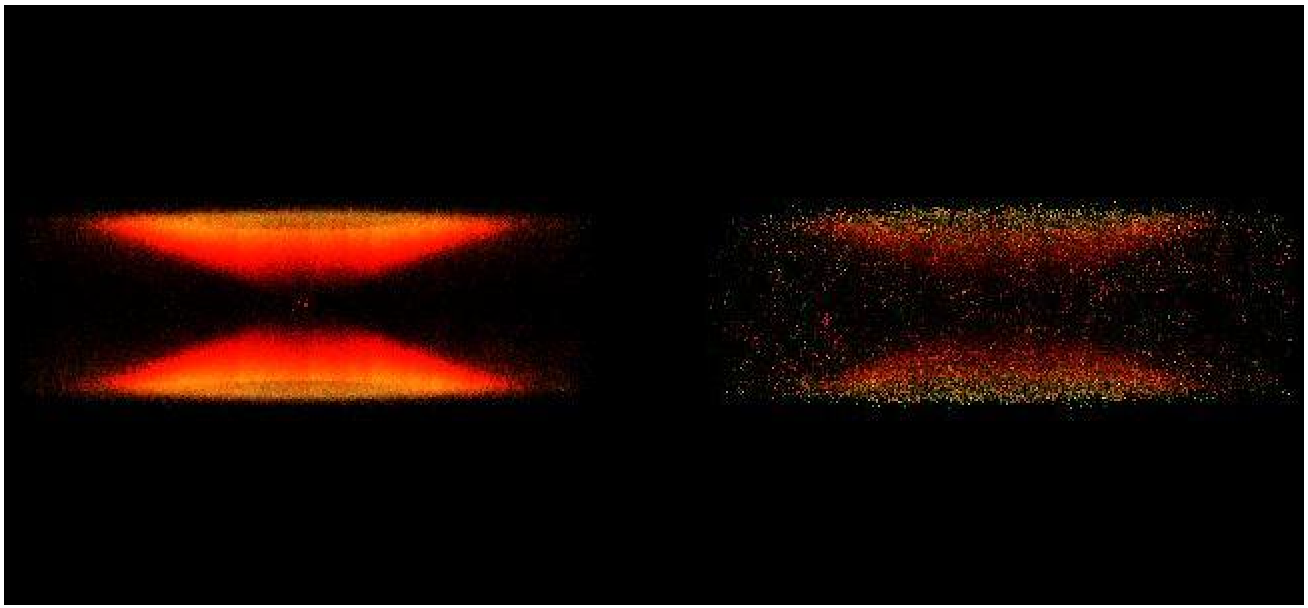
The contribution from this path is:

$$I = 1/(4\pi\, r01^2) * \exp(-\tau 01) * \kappa\, \rho(x1) * a * Ps(x0\text{->}x1\text{->}x2)/r12^2 *$$
$$\exp(-\tau 12) * \kappa\, \rho(x2) * a * Ps(x1\text{->}x2\text{->}x3)/r23^2 *$$
$$\exp(-\tau 23)$$

Then path is mutated by randomly:
- Adding a scattering point
- Removing a scattering point
- Perturbing a scattering point to a new location

and the new path is either accepted or rejected according to Metropolis-Hastings criterion (accept paths that make bigger contributions, basically)

20 min calculation of the edge-on τ=100 circumstellar disk model from Pascucci et al 04:



Metropolis Light Transport
3e6 rays

Normal Monte Carlo
0.75e6 rays

MLT can use more rays because it doesn't need to recalculate the entire path, only the perturbed part. This makes it faster.

Movie!

Overview

Polychromatic Monte Carlo with Sunrise

Metropolis Light Transport

Fast dust temperatures using GPUs

GPU computing

Graphics processors are extremely powerful parallel computers

Latest Nvidia GPUs approach 1000Gflops

Quad-core 3GHz Xeon: ~80Gflops

C-like API: CUDA

Have been used for accelerating N-body and fluid codes

Lectures are available at astrogpu.org.

GPU Temperature calculation

Dust grain equilibrium temperature and emission SED

50,000 grid cells, 30 grain sizes, 50 wavelengths

Xeon E5420, 2.5GHz (Sunrise, C++):	90s
Nvidia 8600GTS (CUDA):	1.4s

60x Speedup!! (and this is a $100 card)

Great potential for expensive calculations of thermally fluctuating grains.

Summary

Polychromatic Monte Carlo can be very effective, especially if high wavelength resolution is desired

Metropolis Light Transport can be a very efficient way of generating images

GPUs are fast and can give order(s) of magnitude speedups (for the right problems)

We can learn a lot from computer graphics research – algorithms are similar and it's a big field