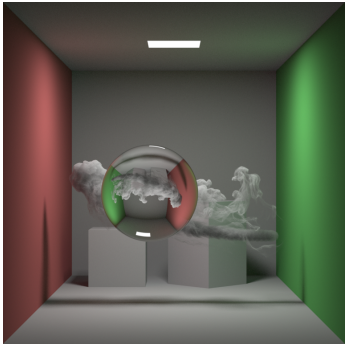
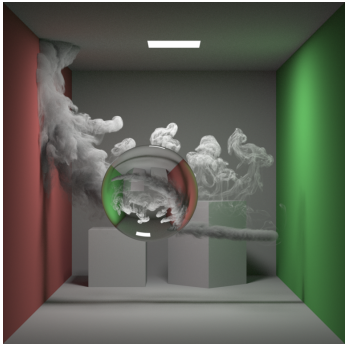
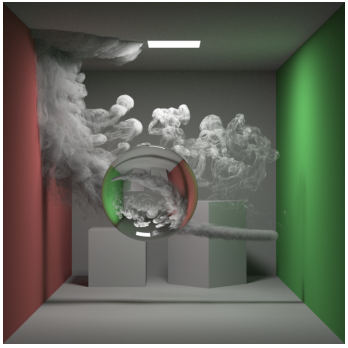
Decoupled Ray-marching of Heterogeneous Participating Media

Christopher Kulla

SONY PICTURES
imageworks







Goals

- Add volume rendering to our production raytracer (smoke, clouds, fire,...)
- Support arbitrary shaders (including time-varying)
- Avoid light caches, preprocessing, and large data structures
- Take advantage of previous work on lighting homogeneous media

Challenge

- Heterogeneous media add a *nested* integral to compute transmission
- This integral is *dependent* on the sample position along the line

Volume Rendering Equation [Kajiya and Von Herzen, 1984]

$$
\begin{aligned}
L(x,\vec{\omega}) &= \int_a^b \sigma_s\left(x+t\vec{\omega}\right) e^{-\tau(t)} \left(\int_{4\pi} \rho(\vec{\omega},\vec{\omega'}) L(x+t\vec{\omega},\vec{\omega'})\, d\vec{\omega'} \right) dt \\
\tau(t) &= \int_0^t \sigma_t(x+t'\vec{\omega})\, dt'
\end{aligned}
$$

Previous Work: Unbiased Methods

- "Unbiased Global Illumination with Participating Media" [Raab et al, 2006]
- "Image Synthesis using Adjoint Photons" [Morley et al, 2006]
- "Unbiased, Adaptive Stochastic Sampling for Rendering Inhomogeneous Participating Media" [Yue et al, 2010]

Unbiased Methods

- Use rejection sampling to implicitly generate samples according to $e^{-\tau(t)}$
- Requires an upper bound on σ_t for the ray (procedural shaders?)
- Acceleration structure [Yue et al] increases memory usage (motion blur?)
- Sample distribution is implicit, pdf is not directly evaluatable

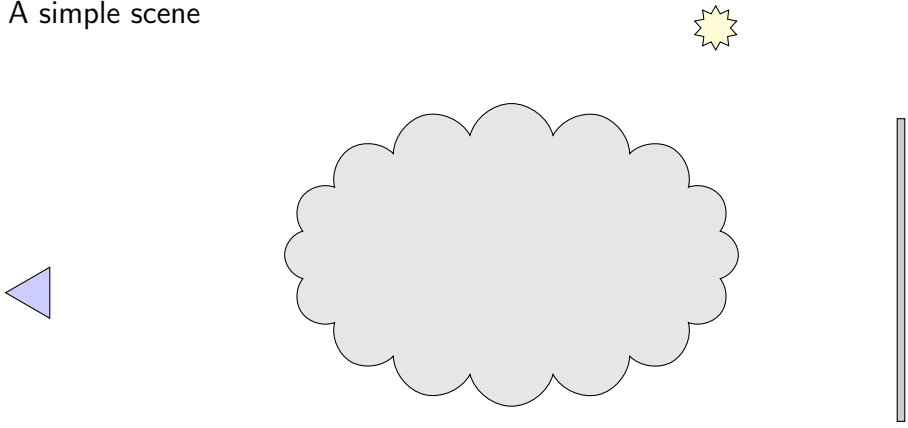
Previous Work: Biased Methods (1/2)

Computing lighting at every sample:

- "Hypertexture" [Perlin and Hoffert, 1989]
- "Production Volume Rendering" [Course notes, 2011]
- Ubiquitous method for VFX volume rendering

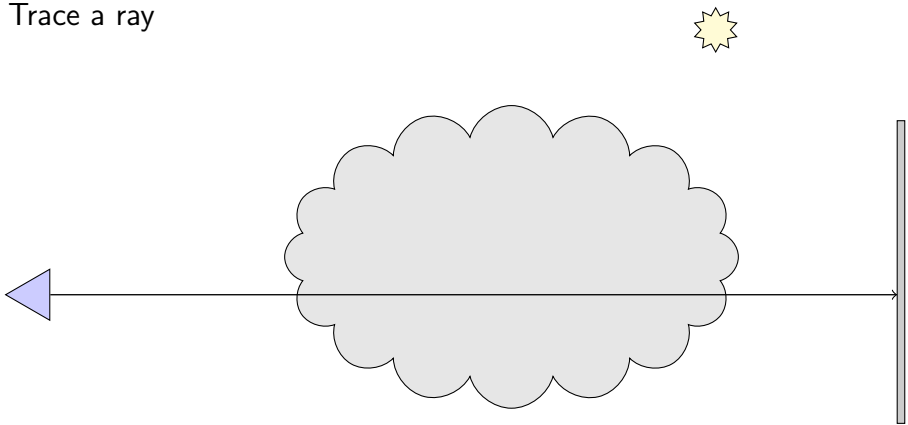
Traditional Ray-marching

A simple scene



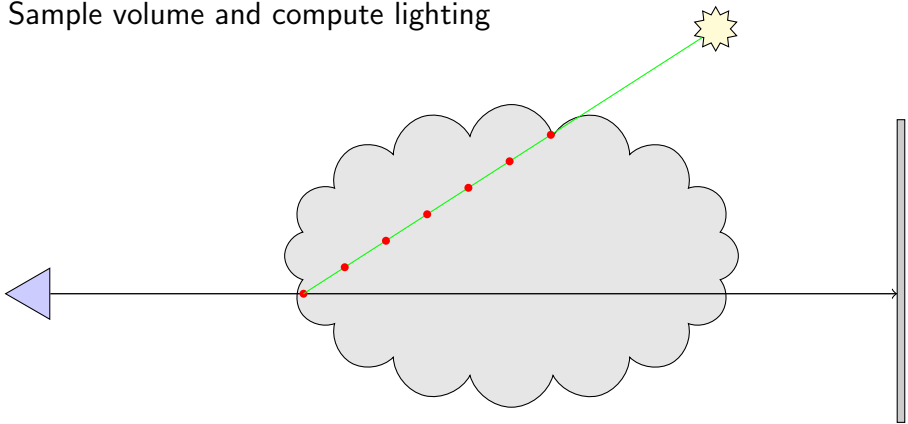
Traditional Ray-marching

Trace a ray

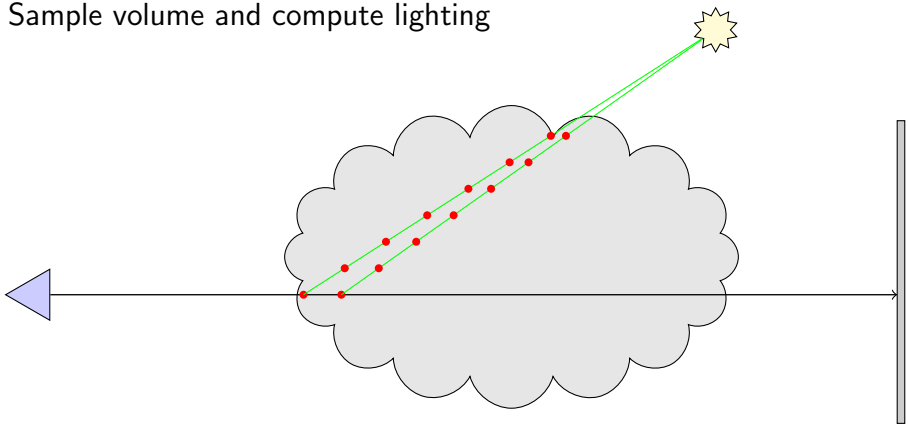


Traditional Ray-marching

Sample volume and compute lighting

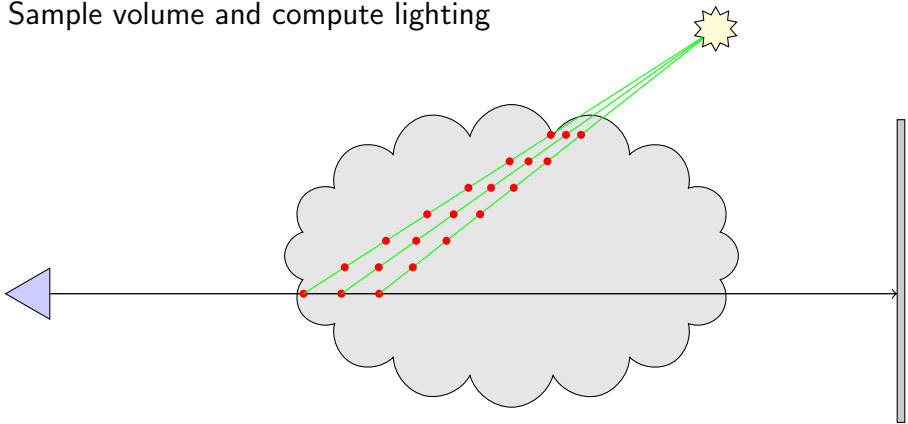


Traditional Ray-marching

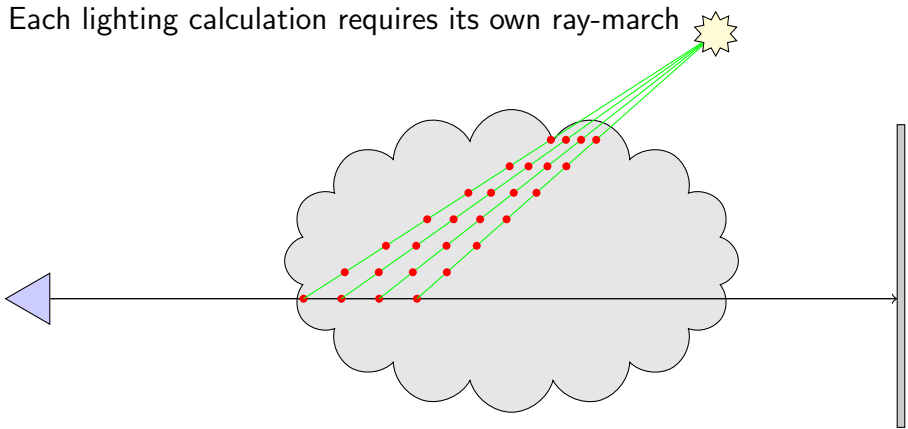


Traditional Ray-marching

Sample volume and compute lighting

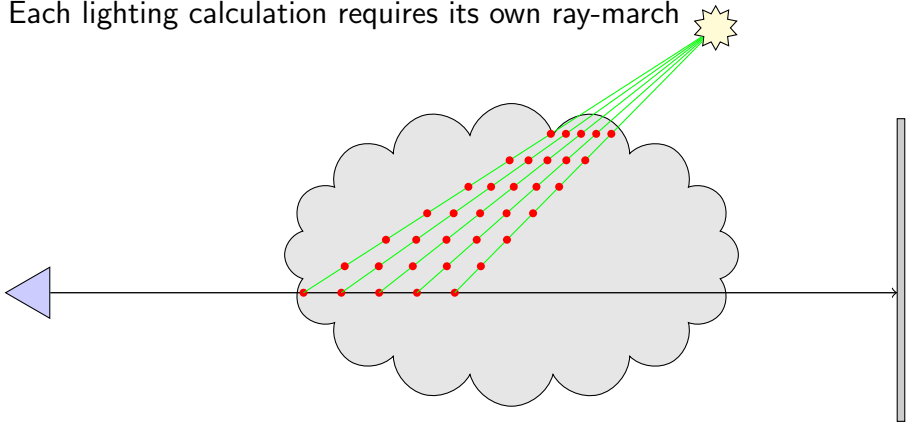


Traditional Ray-marching

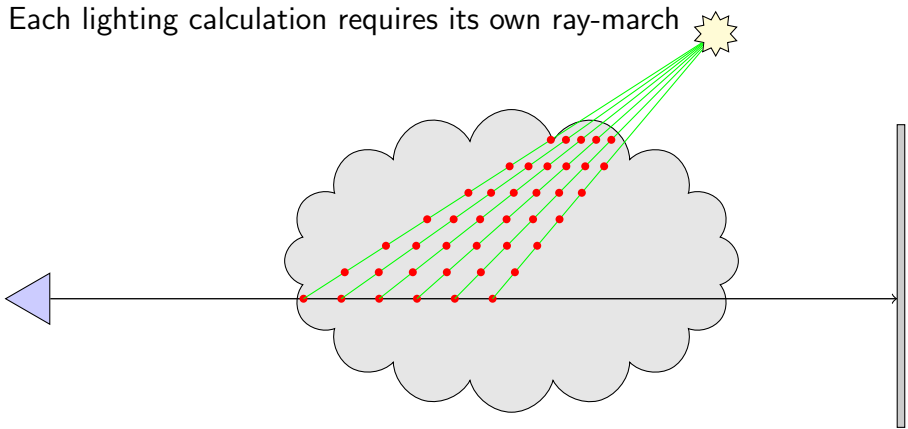


Traditional Ray-marching

Each lighting calculation requires its own ray-march

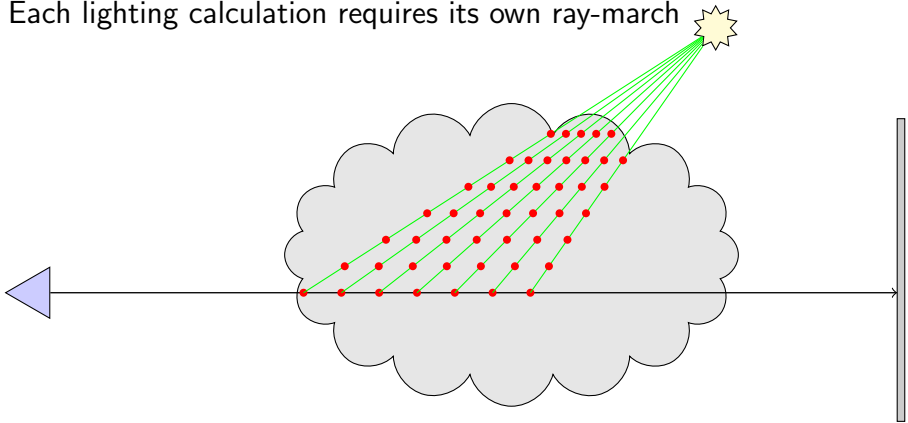


Traditional Ray-marching



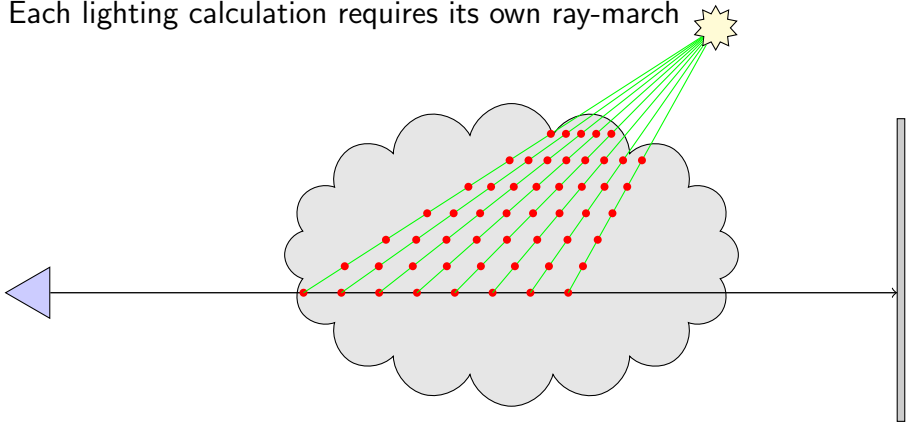
Traditional Ray-marching

Each lighting calculation requires its own ray-march



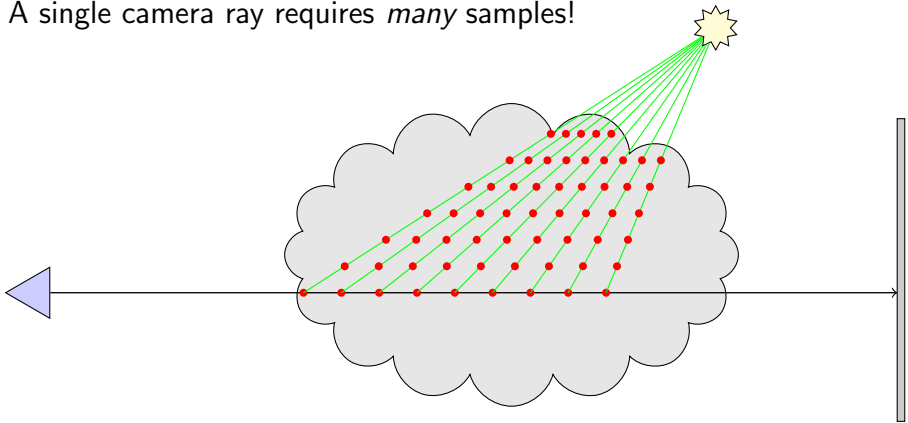
Traditional Ray-marching

Each lighting calculation requires its own ray-march



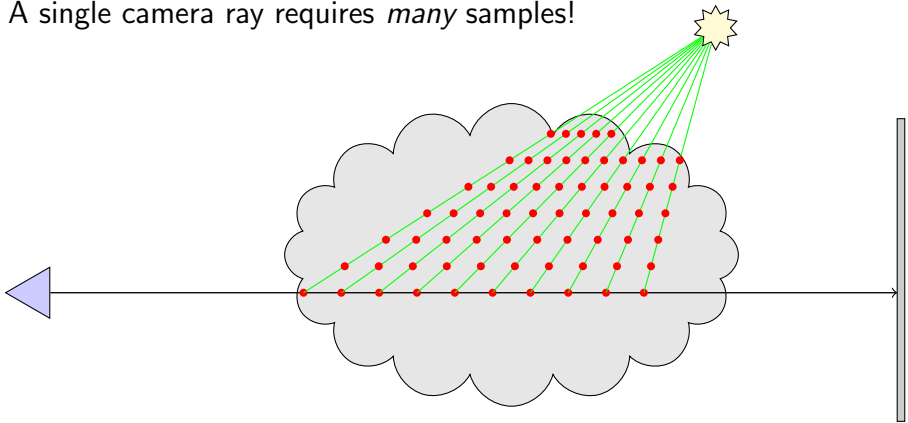
Traditional Ray-marching

A single camera ray requires *many* samples!



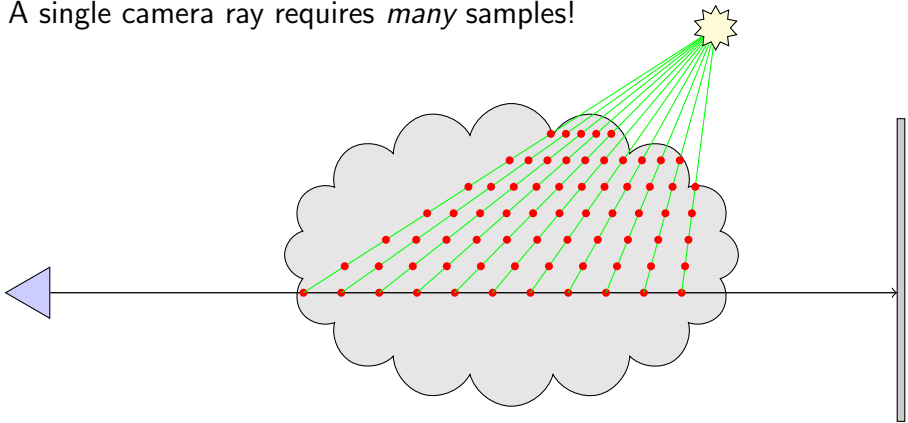
Traditional Ray-marching

A single camera ray requires *many* samples!



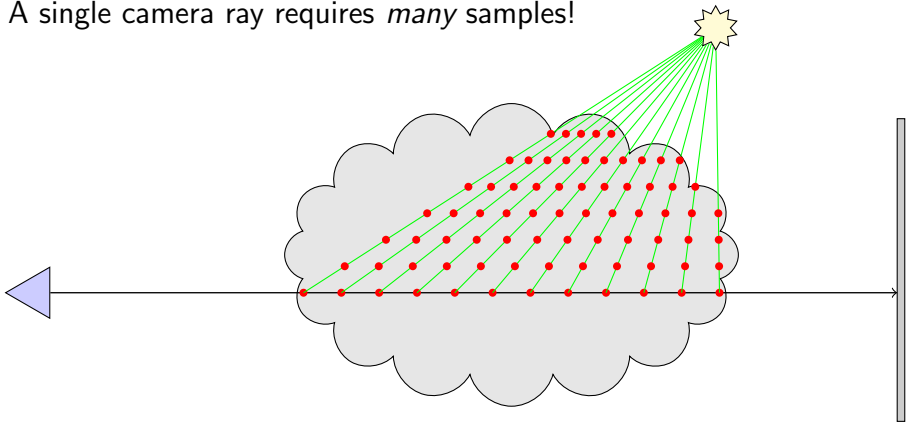
Traditional Ray-marching

A single camera ray requires *many* samples!



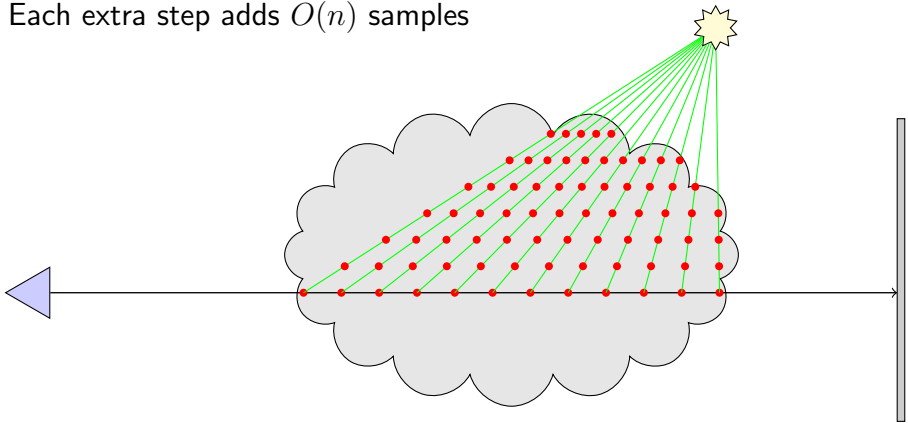
Traditional Ray-marching

A single camera ray requires *many* samples!



Traditional Ray-marching

Each extra step adds $O(n)$ samples



Previous Work: Biased Methods (1/2)

Computing lighting at every sample:

- Brute force algorithm is $O(n^2)$

Previous Work: Biased Methods (1/2)

Computing lighting at every sample:

- Brute force algorithm is $O(n^2)$
- Requires use of cached lighting

Previous Work: Biased Methods (1/2)

Computing lighting at every sample:

- Brute force algorithm is $O(n^2)$
- Requires use of cached lighting
- "Deep Shadow Maps" [Lokovic and Veach, 2000]

Previous Work: Biased Methods (1/2)

Computing lighting at every sample:

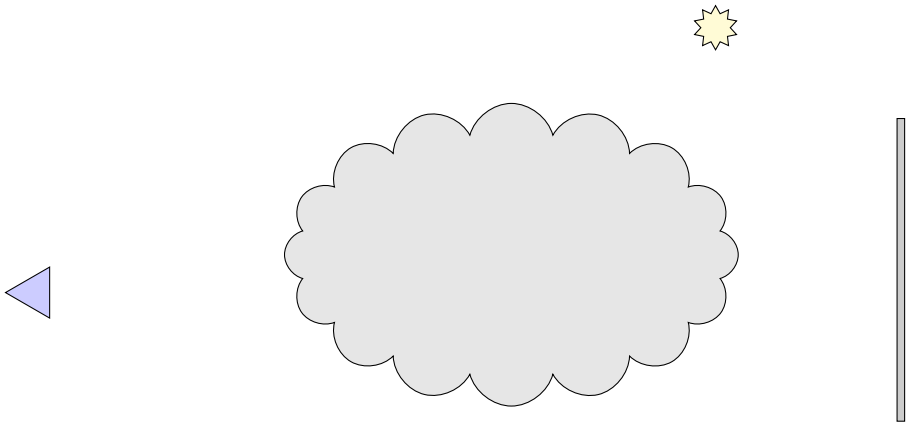
- Brute force algorithm is $O(n^2)$
- Requires use of cached lighting
- "Deep Shadow Maps" [Lokovic and Veach, 2000]
- Multiple scattering requires further approximations

Previous Work: Biased Methods (2/2)

Using ray marching to propose a sample point:

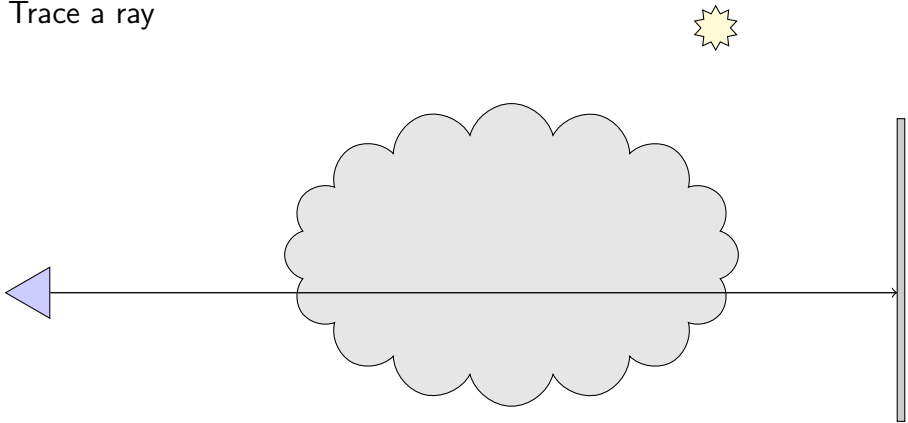
- "Rendering Participating Media with Bidirectional Path Tracing" [Lafortune et al, 1996]
- "Metropolis Light Transport for Participating Media" [Pauly et al, 2000]

Stochastic Ray-marching



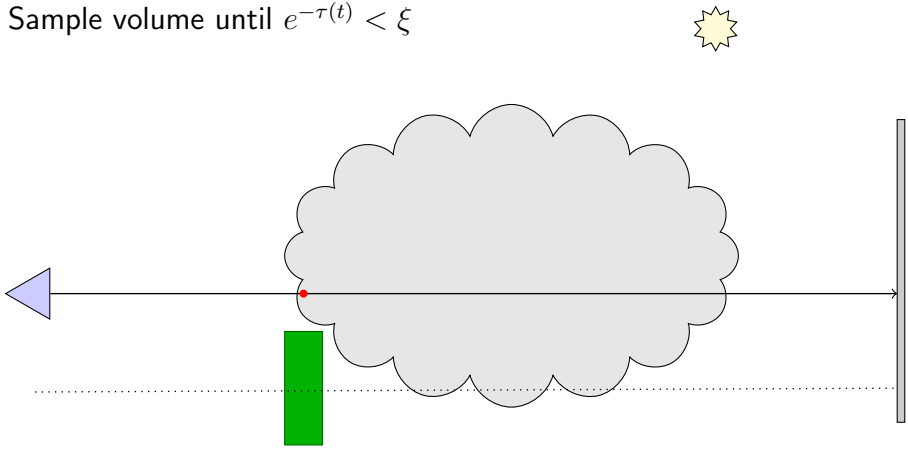
Stochastic Ray-marching

Trace a ray



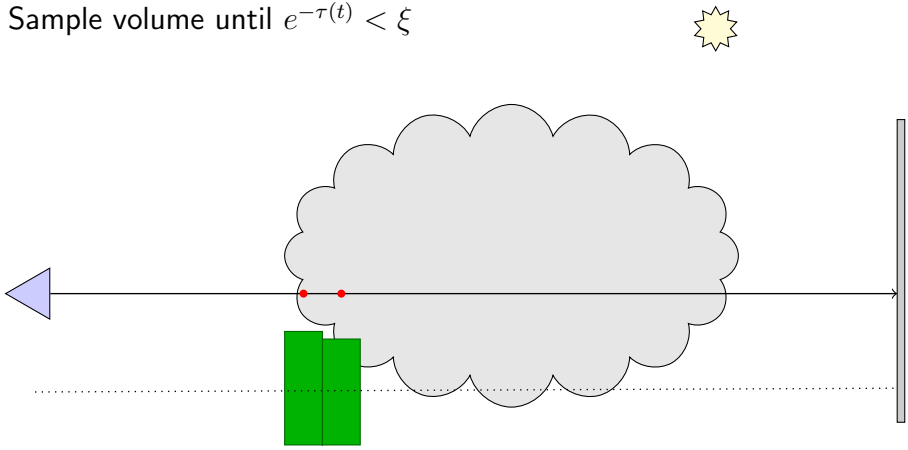
Stochastic Ray-marching

Sample volume until $e^{-\tau(t)} < \xi$



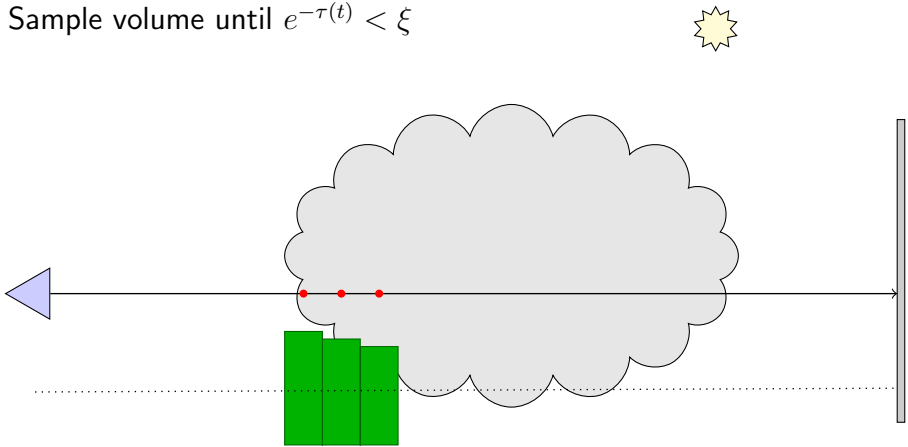
Stochastic Ray-marching

Sample volume until $e^{-\tau(t)} < \xi$



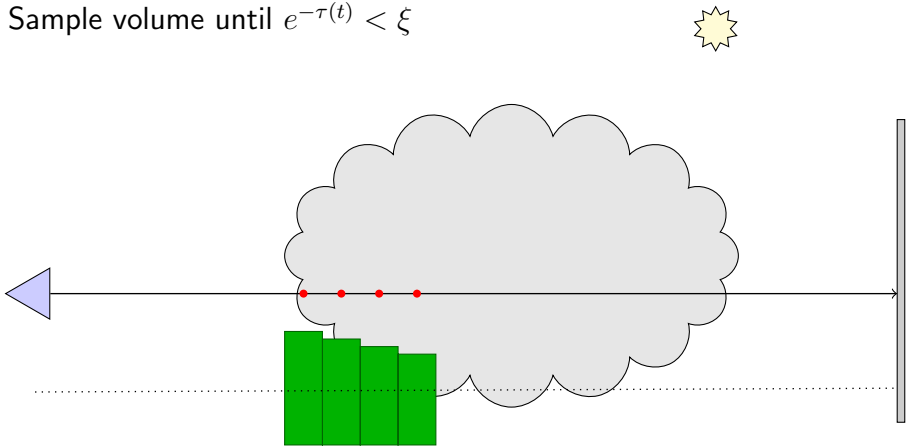
Stochastic Ray-marching

Sample volume until $e^{-\tau(t)} < \xi$



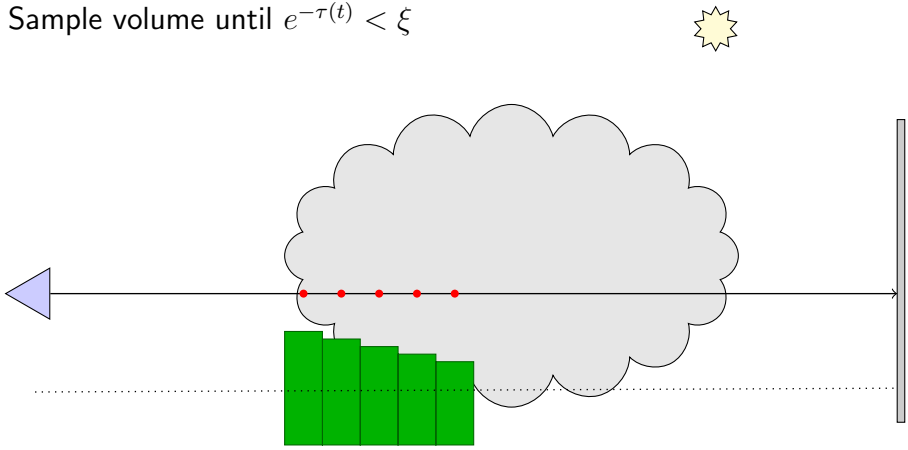
Stochastic Ray-marching

Sample volume until $e^{-\tau(t)} < \xi$



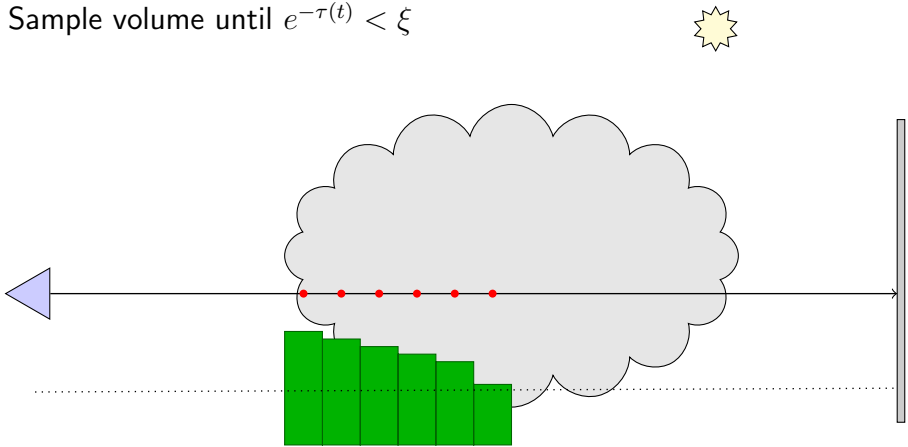
Stochastic Ray-marching

Sample volume until $e^{-\tau(t)} < \xi$



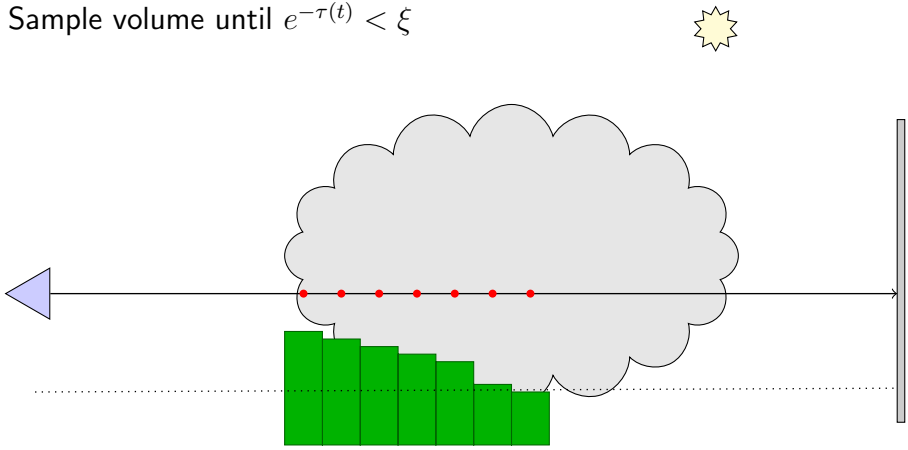
Stochastic Ray-marching

Sample volume until $e^{-\tau(t)} < \xi$

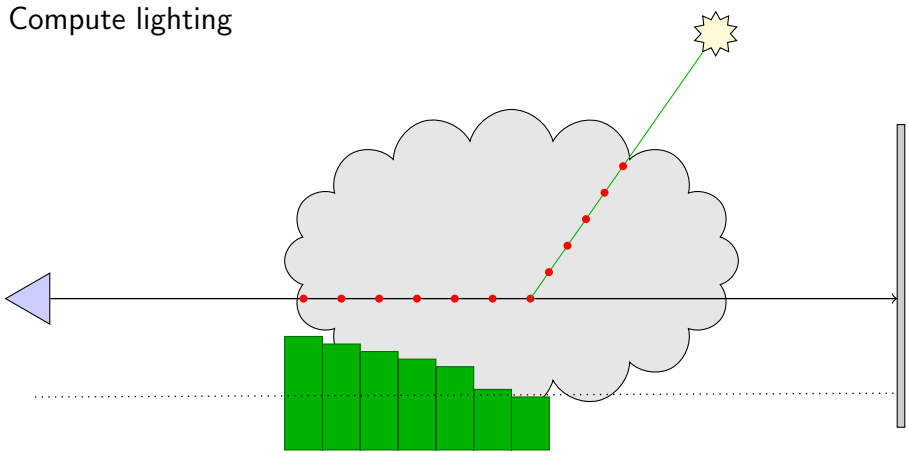


Stochastic Ray-marching

Sample volume until $e^{-\tau(t)} < \xi$



Stochastic Ray-marching



Previous Work: Biased Methods (2/2)

Using ray marching to propose a sample point:

- Similar to unbiased algorithm, but biased due to fixed size steps

Previous Work: Biased Methods (2/2)

Using ray marching to propose a sample point:

- Similar to unbiased algorithm, but biased due to fixed size steps
- Sample distribution is only implicit, pdf is not directly evaluatable

Previous Work: Biased Methods (2/2)

Using ray marching to propose a sample point:

- Similar to unbiased algorithm, but biased due to fixed size steps
- Sample distribution is only implicit, pdf is not directly evaluatable
- Taking n samples for a single ray requires restarting the ray march ...

Previous Work: Biased Methods (2/2)

Using ray marching to propose a sample point:

- Similar to unbiased algorithm, but biased due to fixed size steps
- Sample distribution is only implicit, pdf is not directly evaluatable
- Taking n samples for a single ray requires restarting the ray march ...
- ... or sorting the sampling pattern ($\xi_1 > \xi_2 > \ldots > \xi_n$)

Previous Work: Biased Methods (2/2)

Using ray marching to propose a sample point:

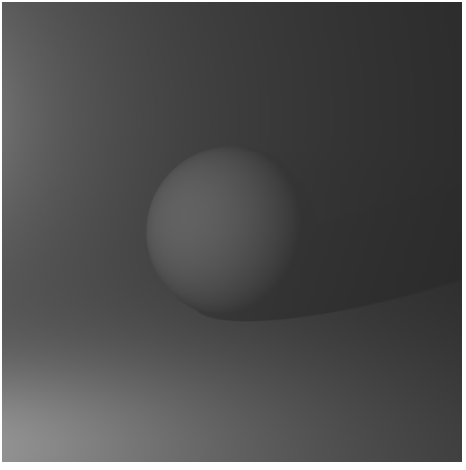
- Similar to unbiased algorithm, but biased due to fixed size steps
- Sample distribution is only implicit, pdf is not directly evaluatable
- Taking n samples for a single ray requires restarting the ray march ...
- ... or sorting the sampling pattern ($\xi_1 > \xi_2 > \ldots > \xi_n$)
- Lighting and transmission computed with the same number of samples

Our Algorithm

- Decouples ray marching from lighting
- PDF along the ray can be evaluated and inverted efficiently
- Allows combining this work with methods from previous talk
- No restrictions on shaders
- No large data structures

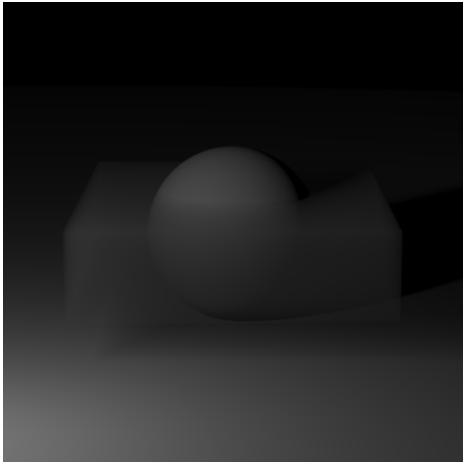
Observation

- Homogeneous case is efficient



Observation

- Homogeneous case is efficient
- Even when restricted to a finite region

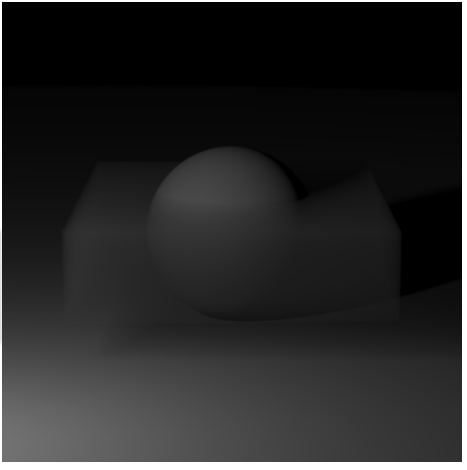


Observation

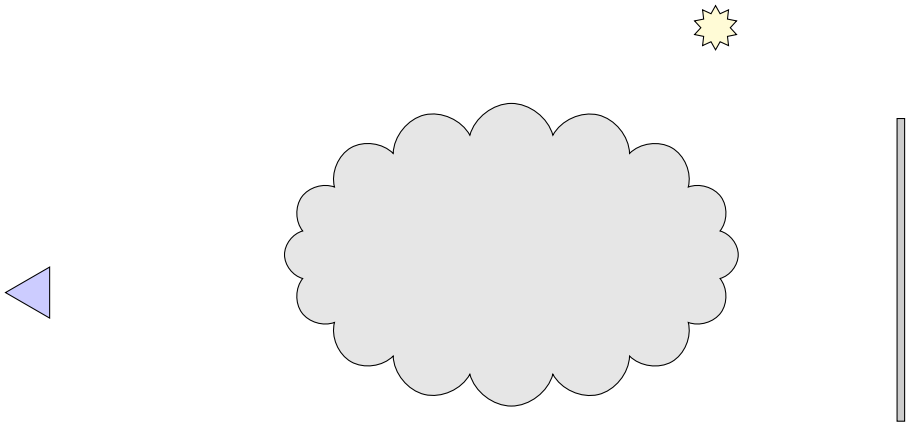
- Homogeneous case is efficient
- Even when restricted to a finite region

Idea

Extend our methods to deal with n homogeneous segments along the ray

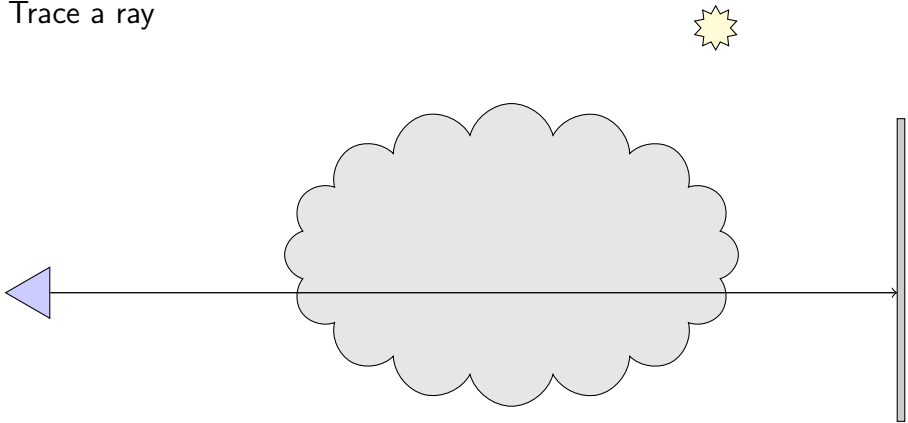


Decoupled Ray-marching



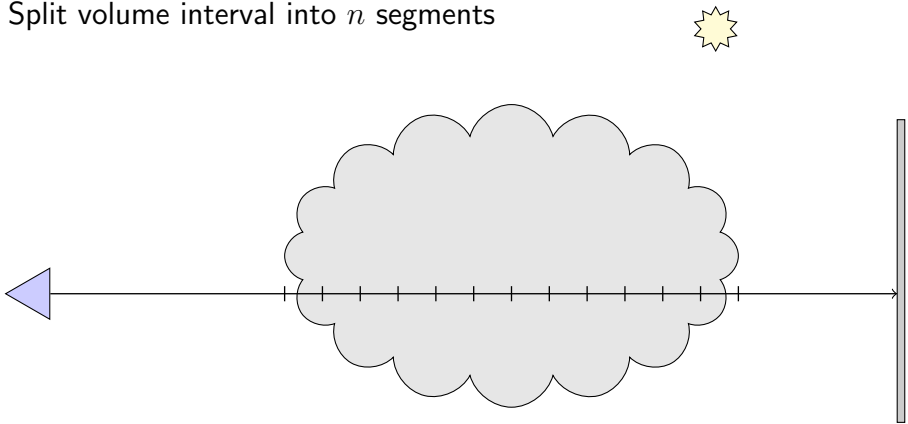
Decoupled Ray-marching

Trace a ray



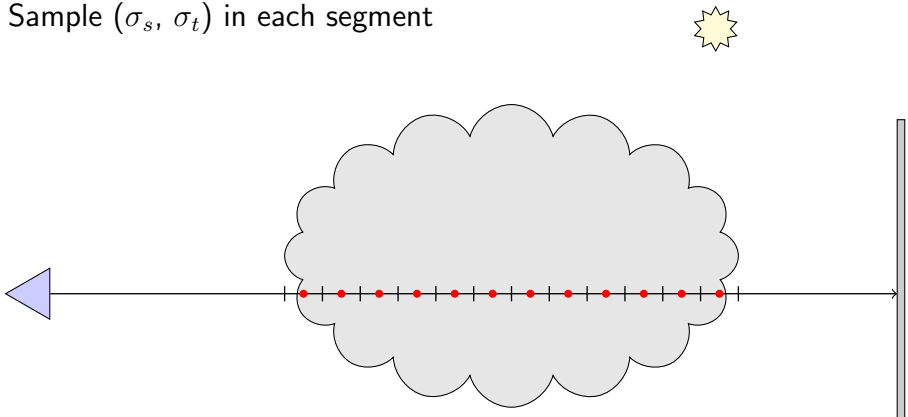
Decoupled Ray-marching

Split volume interval into n segments



Decoupled Ray-marching

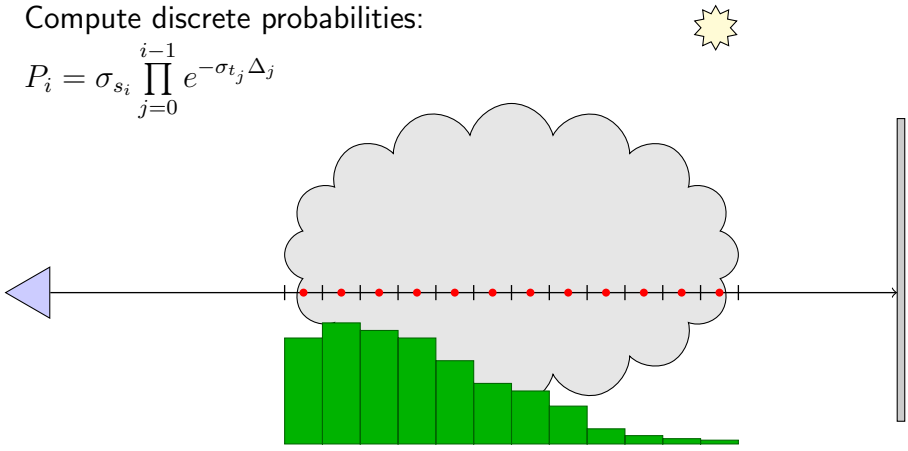
Sample (σ_s, σ_t) in each segment



Decoupled Ray-marching

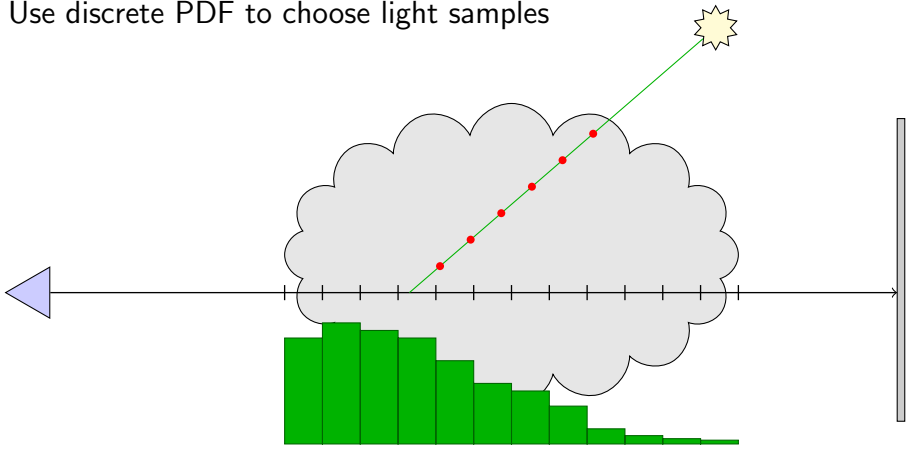
Compute discrete probabilities:

$$P_i = \sigma_{s_i} \prod_{j=0}^{i-1} e^{-\sigma_{t_j} \Delta_j}$$



Decoupled Ray-marching

Use discrete PDF to choose light samples



Decoupled Ray-marching: Advantages

- Unbiased when segments are really homogeneous

Decoupled Ray-marching: Advantages

- Unbiased when segments are really homogeneous
- Converges as step size approaches Nyquist limit

Decoupled Ray-marching: Advantages

- Unbiased when segments are really homogeneous
- Converges as step size approaches Nyquist limit
- $\tau(t)$ estimate is *decoupled* from number of lighting samples

Decoupled Ray-marching: Advantages

- Unbiased when segments are really homogeneous
- Converges as step size approaches Nyquist limit
- $\tau(t)$ estimate is *decoupled* from number of lighting samples
- Lighting samples focus on areas where $\sigma_s > 0$

Time Complexity

Assuming we take $O(n)$ steps for ray-marching:

- Obtain CDF from PDF in $O(n)$

Time Complexity

Assuming we take $O(n)$ steps for ray-marching:

- Obtain CDF from PDF in $O(n)$
- Can draw samples in $O(\log(n))$ (binary search on CDF)

Time Complexity

Assuming we take $O(n)$ steps for ray-marching:

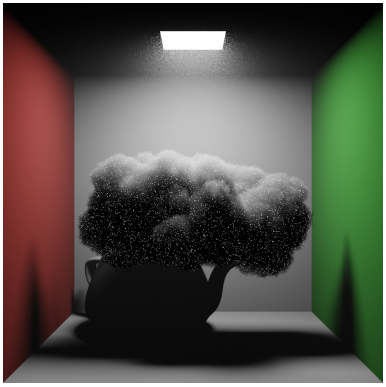
- Obtain CDF from PDF in $O(n)$
- Can draw samples in $O(\log(n))$ (binary search on CDF)
- Can evalute PDF and τ for any t in $O(log(n))$ (binary search on segments)

Time Complexity

Assuming we take $O(n)$ steps for ray-marching:

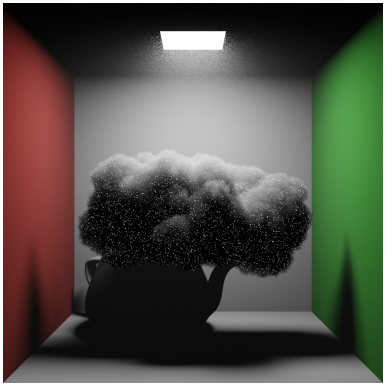
- Obtain CDF from PDF in $O(n)$
- Can draw samples in $O(\log(n))$ (binary search on CDF)
- Can evalute PDF and τ for any t in $O(log(n))$ (binary search on segments)
- Allows combinations of line-sampling strategies!

Multiple Importance Sampling

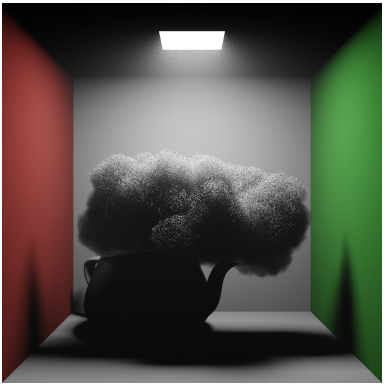


Density Sampling

Multiple Importance Sampling

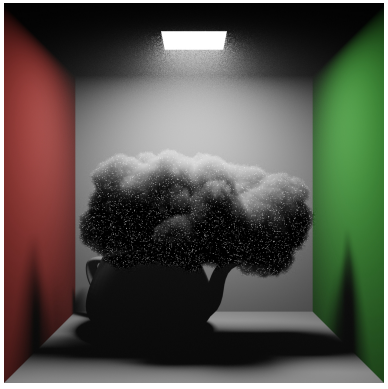


Density Sampling

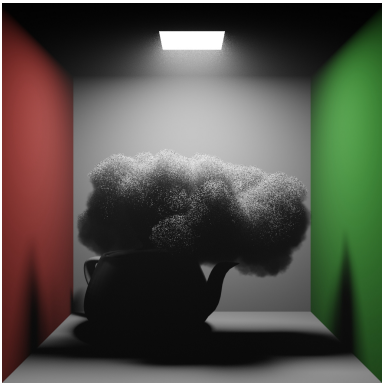


Equi-angular Sampling

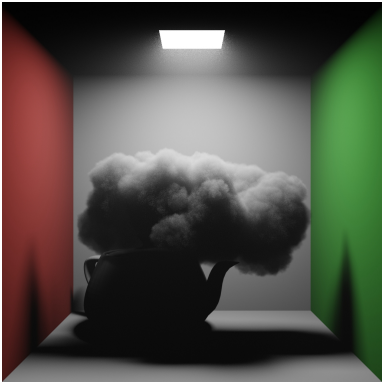
Multiple Importance Sampling



Density Sampling

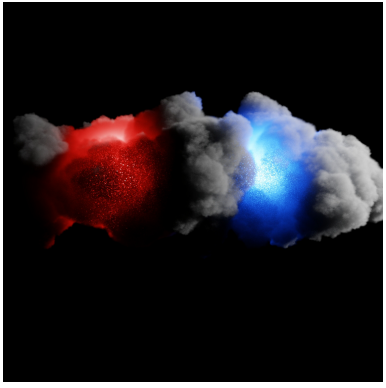


Equi-angular Sampling



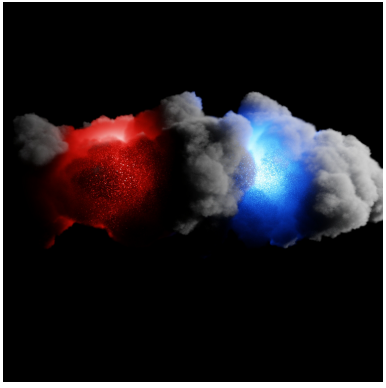
MIS

Multiple Importance Sampling

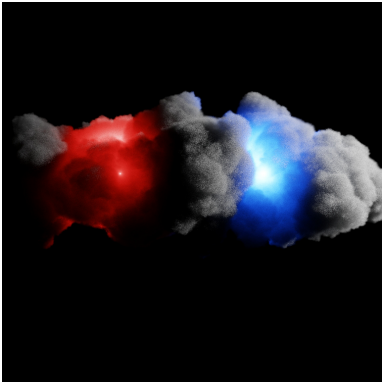


Density Sampling

Multiple Importance Sampling

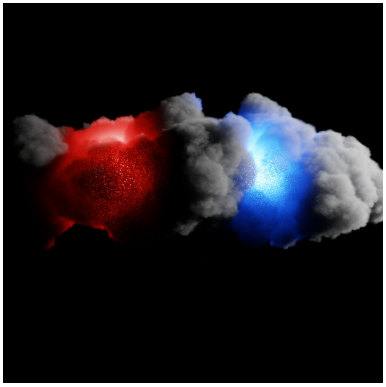


Density Sampling

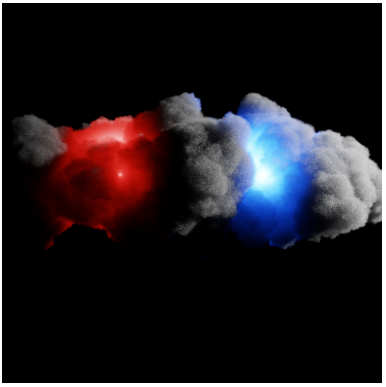


Equi-angular Sampling

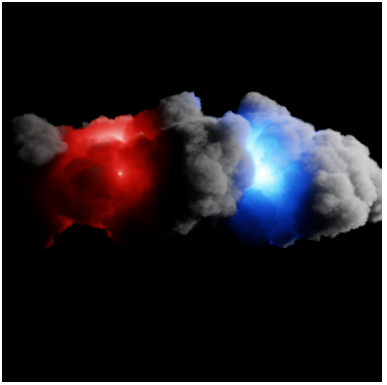
Multiple Importance Sampling



Density Sampling



Equi-angular Sampling

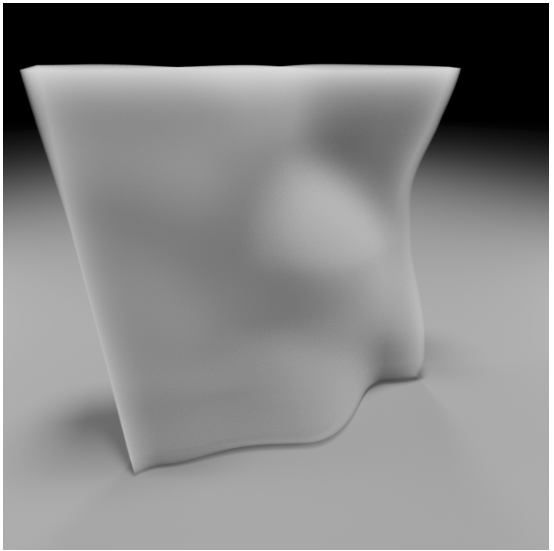


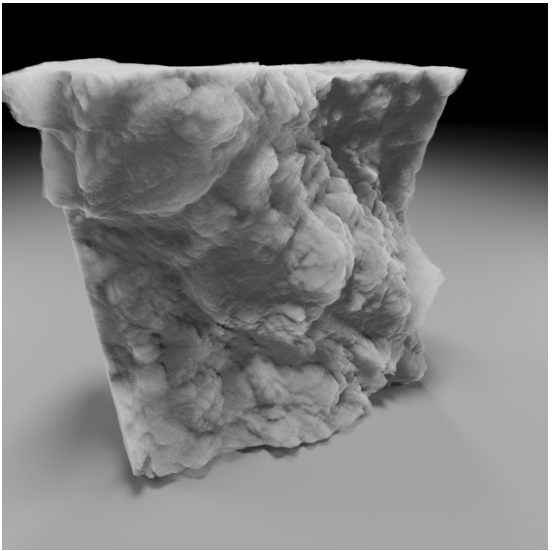
MIS

Implementation Details: Memory Management

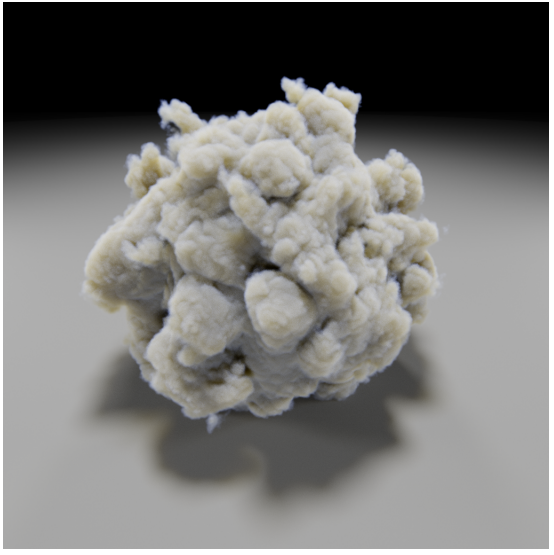
- Algorithm requires temporary arrays
- Use a per-thread memory pool with a stack interface
- Push pool state before a ray is processed
- Pop state when the ray volume shading is done
- Allows memory to grow only with the recursion depth (only a few Kb in practice)

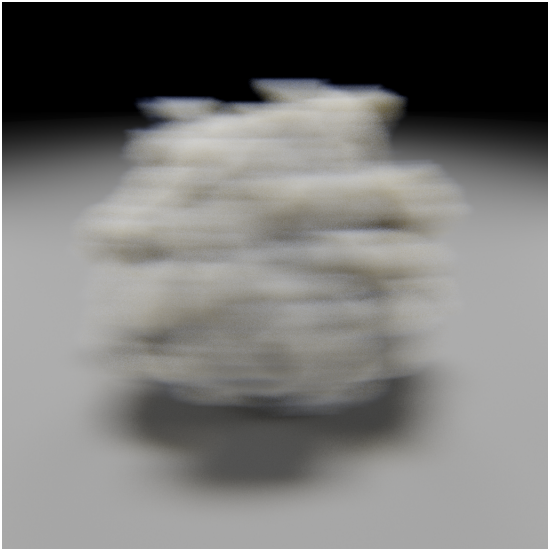
Examples



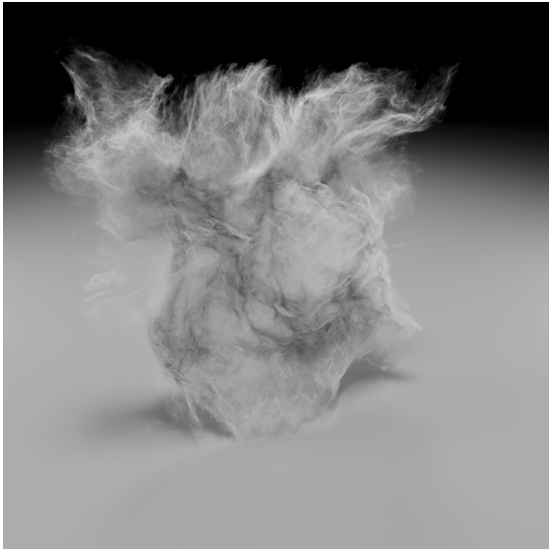


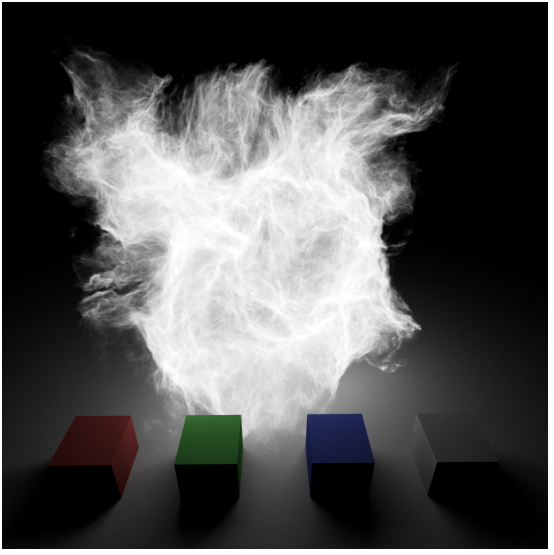
Examples





Examples





Examples

0 bounces
3m28s

1 bounces
10m14s

2 bounces
20m21s

Examples

Videos

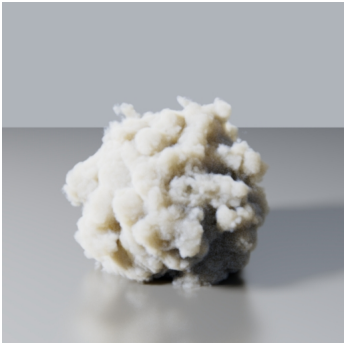
Summary

- Practical method for ray traced lighting of heterogeneous volumes
- Lighting calculations are decoupled from volume sampling
- Supports procedural shaders
- Supports all light source types
- Allows for multiple scattering
- Does not require any (permanent) auxiliary data structures

Future Work

- Investigate better weights for MIS
- Investigate sampling distributions for multiple scattering
- Automate the step size using ray differentials
- Efficient illumination from volumes (fire as light source)
- Out-of-core datasets

Thanks for listening!



Acknowledgment: Magnus Wrenninge (models, fluid sims)