PCBEX: Point-Based Color Bleeding With Volumes Thesis Defense

Christopher James Gibson

California Polytechnic University
CSC Department

June 9, 2011

Schedule

Outline

Graphics Intro

Rendering is the process of producing 2D images from a 3D scene description

At its core, all of **computer graphics** is the visualization of how light interacts in these scenes

Modelling physically correct light interactions can get extremely computationally expensive

Volume Rendering

Volumes *(or participating media)* represent:

- Fog
- Smoke
- Water
- Clouds
- Other Mediums...

Better for visualizing three-dimensional datasets

Important to simulate and light properly for film, entertainment & medical uses

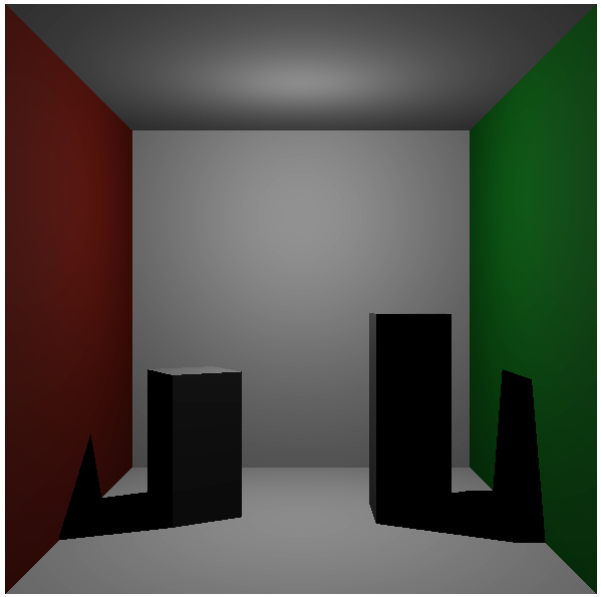
Very computationally expensive to light properly

Skull visualized via 3D dataset

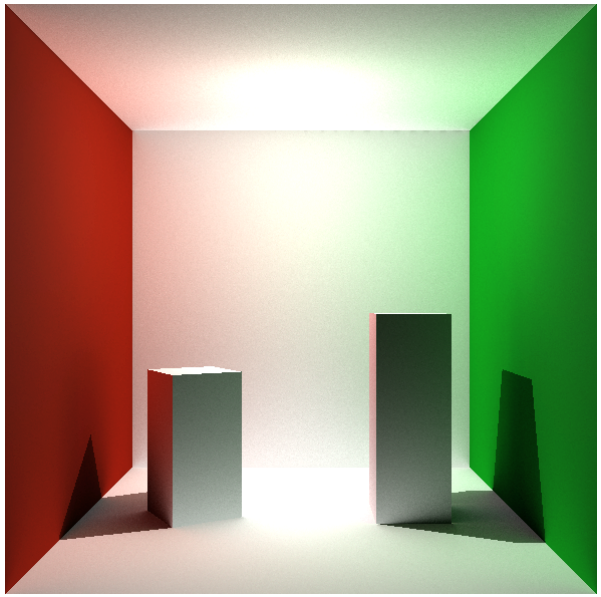
Global Illumination

Global Illumination represents the set of algorithms that estimate complex lighting systems

The results lead to richer, more realistic images



Direct Lighting



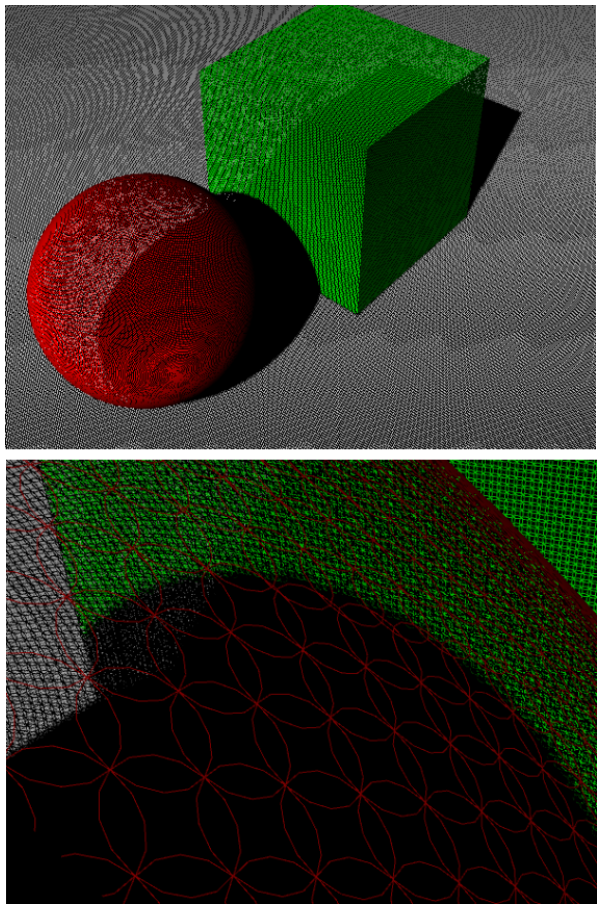
Direct Lighting & Global Illumination

Point-Based Color Bleeding (PCB)

Cheap, accurate global illumination effects using color bleeding

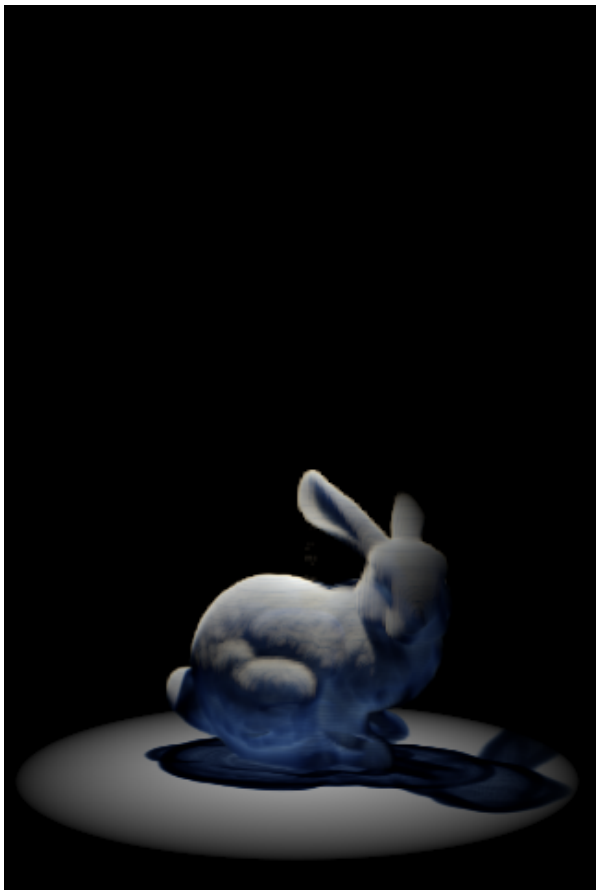
Utilizes direct light point cloud representation of scene's direct lighting

Already used heavily in production due to its performance advantages



Surfels in PCB

Motivation

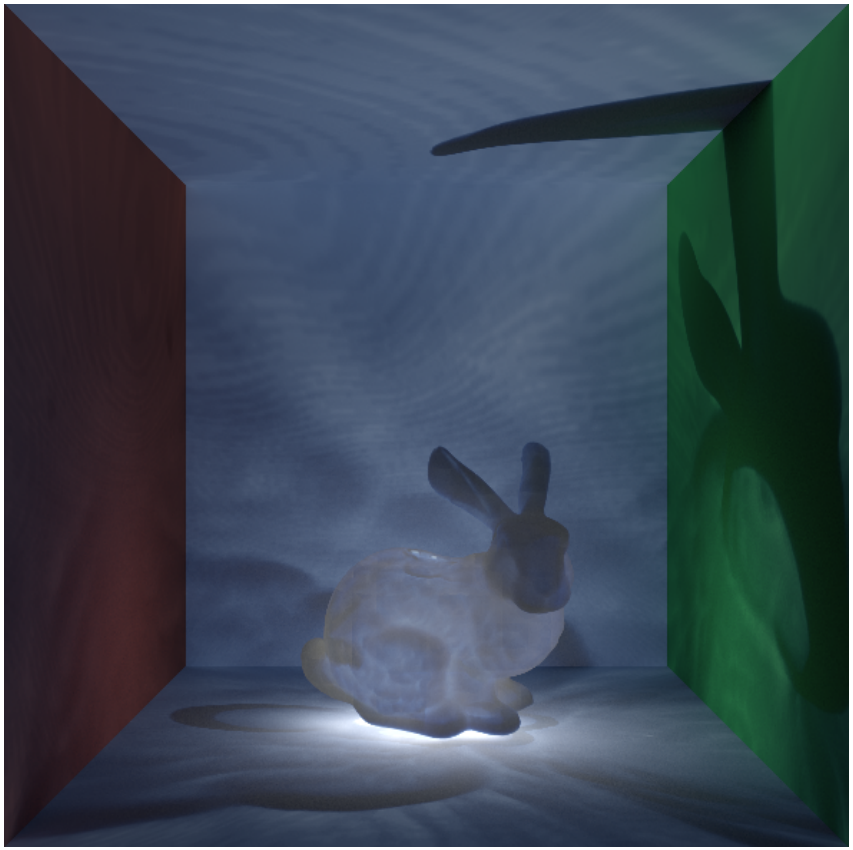


Complex volume lighting algorithms are computationally expensive and complex

Most GA algorithms do not include volume contributions

Goals

- Modify existing PCB in order to include volumetric lighting
- Accurately model scatter, absorption and scatter properties
- Modify volume integration algorithm to add scatter-in effect
- Return comparable results with shorter overall runtimes



Outline

Flux and Radiance

Flux is the measure of total light emitted *(Watts)*

Radiance & Radiance Invariance

$$L = \frac{d^2\,\Phi}{dw\;dA^{\perp}} = \frac{d^2\,\Phi}{dw\;dA\cos\theta}$$

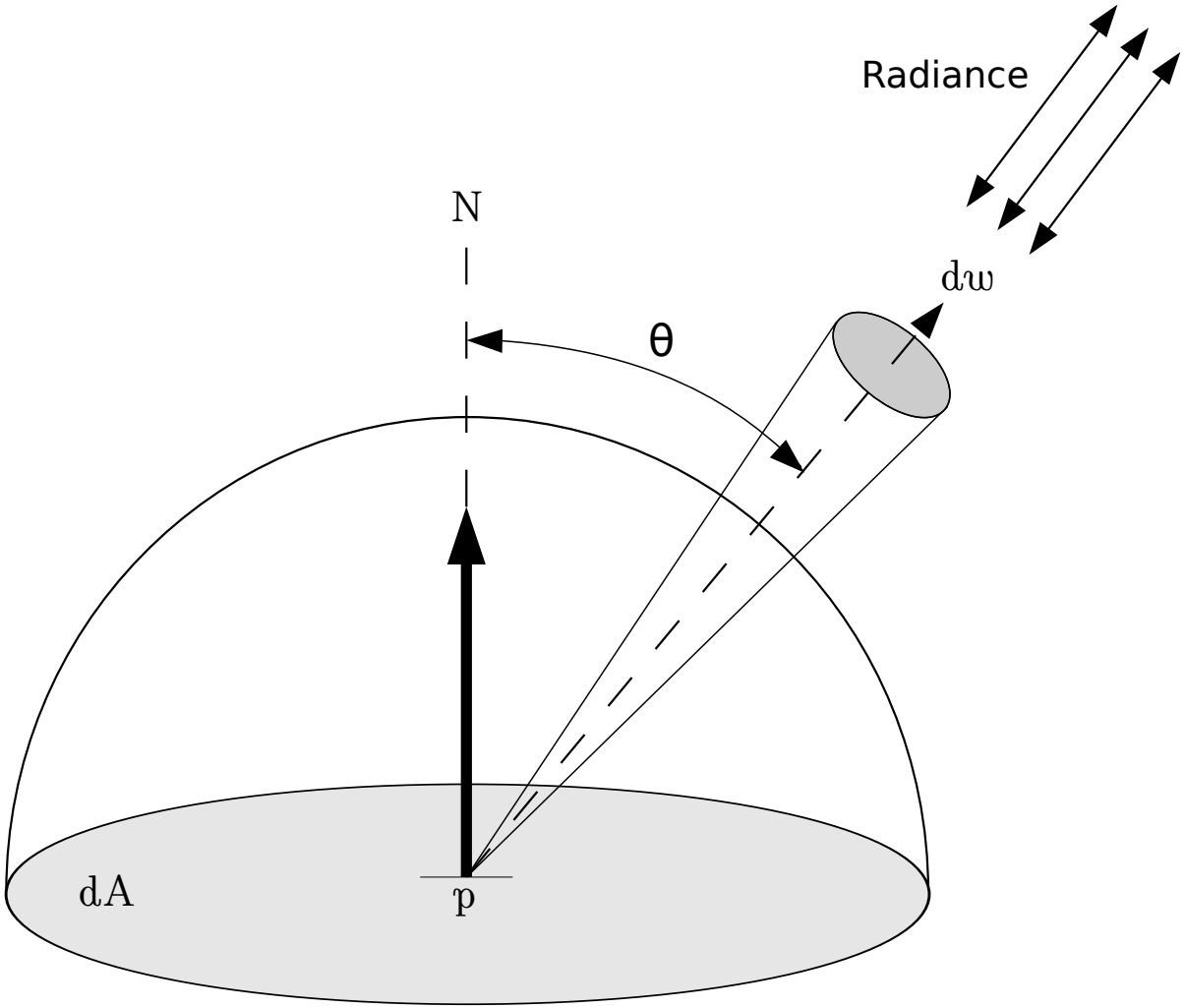
$$L(x \to y) = L(y \to x)$$

Irradiance

$$E = \frac{d\Phi}{dA}$$

$$E = \int L(\mathrm{p} \leftarrow w)\cos\theta dw$$

Irradiance



BRDF and BSRDF

BRDF

Bidirectional **R**eflectance **D**istribution **F**unction

Gives us a formalism for describing the reflection from a surface

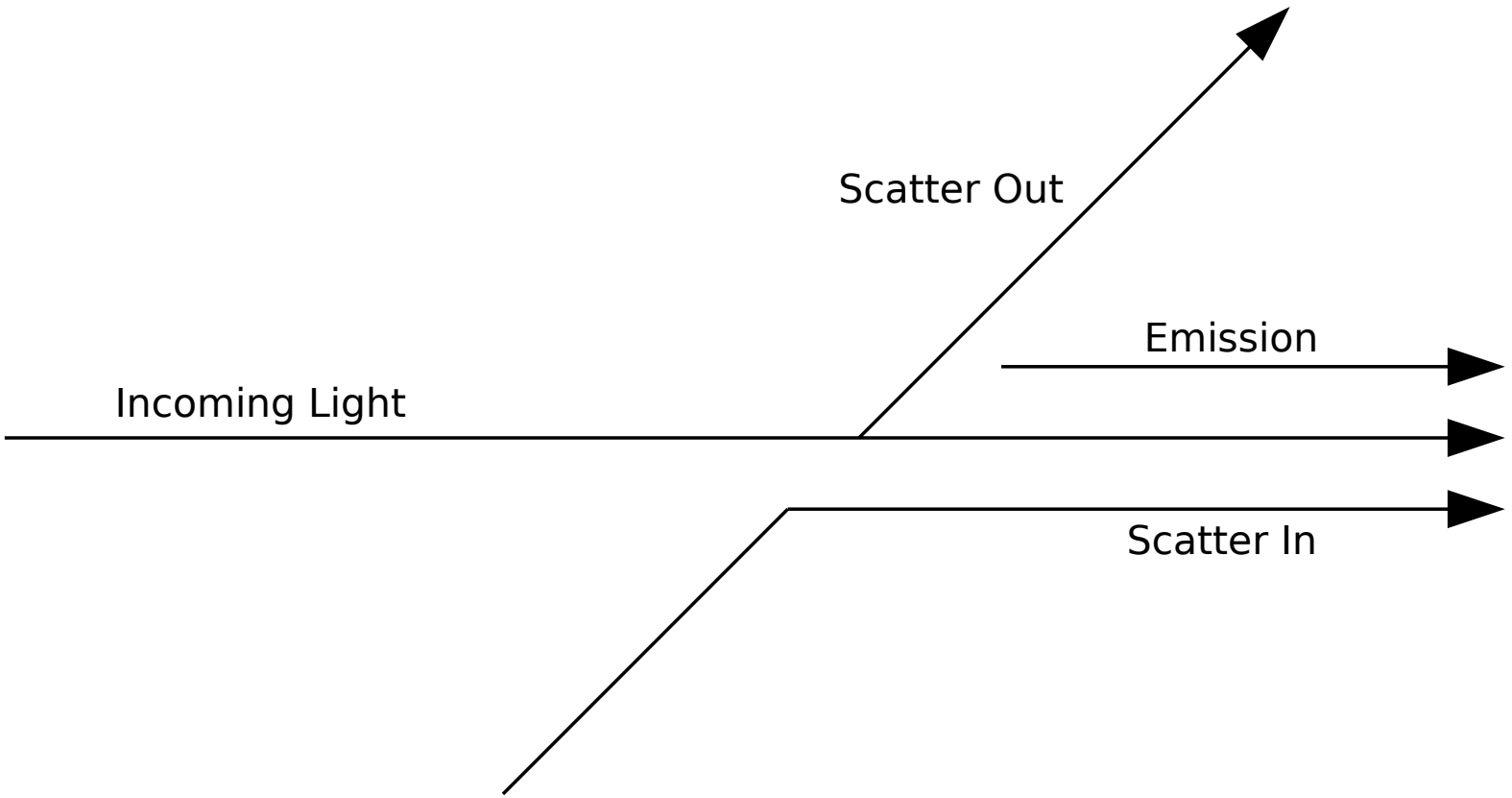
$$dE(\mathsf{p}, w_i) = \int L_i(\mathsf{p} \leftarrow w_i)\cos\theta_i dw_i.$$

BSRDF

Bidirectional **S**cattering-surface **R**eflectance **D**istribution **F**unction

Describes complex behavior of light within surface
(*subsurface scattering*.)

Volume Lighting



Volume Lighting

Absorption

$$e^{-\int_0^d \sigma_a(p+tw,w)dt}$$

Scatter Out

$$dL_o(\mathsf{p},w) = -\sigma_s(\mathsf{p},w)L_i(\mathsf{p},-w)dt$$

Extinction in a volume

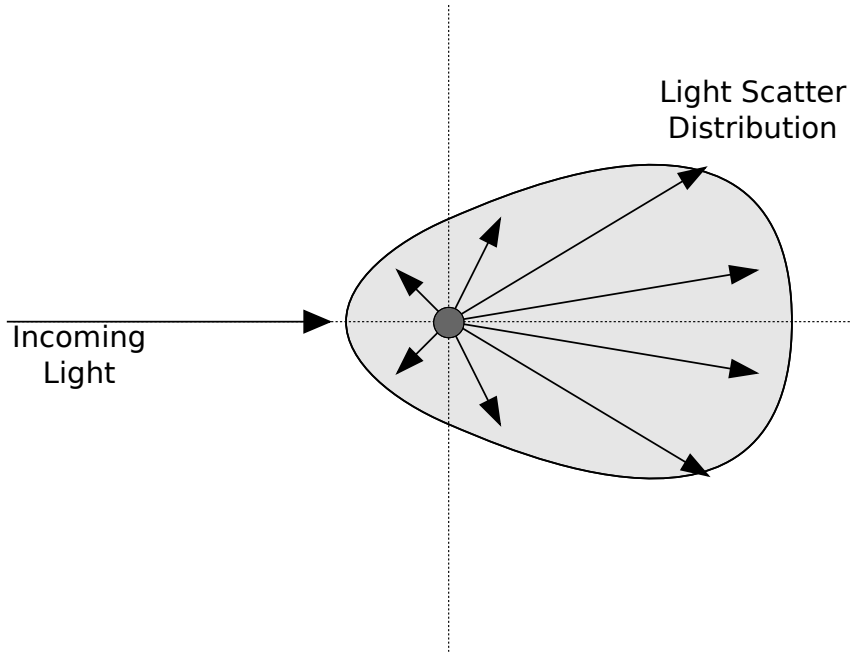
Volume Lighting

Phase Function

Described as $phase(w \rightarrow w')$

Source Normalization

$$\int_{\mathbb{S}^2} phase(w \rightarrow w')\mathrm{d}w' = 1.$$



Phase function distribution

Volume Lighting

Transmittance

$$T_r(\mathsf{p} \rightarrow \mathsf{p}') = e^{-\int_0^d \sigma(p+tw,w)dt}.$$

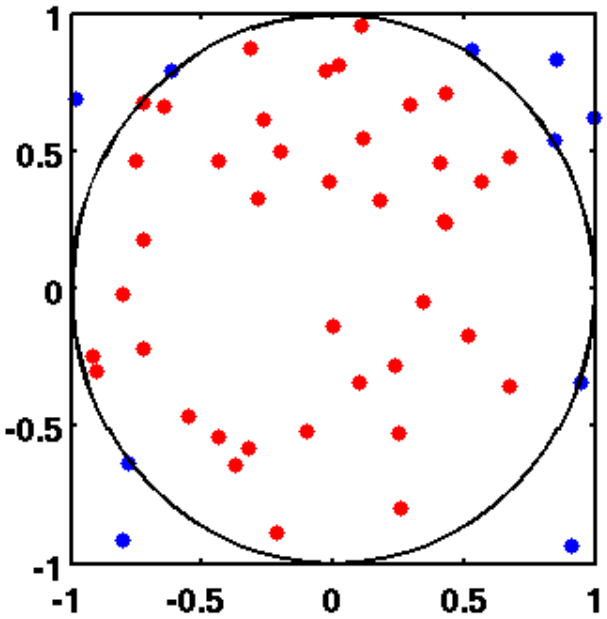
Scatter In

$$S(\mathsf{p}, w) = L_{\mathsf{ve}}(\mathsf{p}, w) + \sigma_{\mathsf{s}}(\mathsf{p}, w) \int_{\mathbb{S}^2} phase(\mathsf{p}, -w' \rightarrow w) L_i(\mathsf{p}, w') \mathsf{d}w'.$$

Monte Carlo Integration

Monte Carlo methods allow estimation of complex systems through use of probability functions and random numbers.

Most useful to us is Monte Carlo Integration.



Randomly sampling a circle

Outline

Related Works: Global Illumination

Heavy field of research in computer graphics.

Involves many priminent technologies and algorithms:

- Photon Mapping
- Radiometry/Radiosity
- Precomuted Light Transport

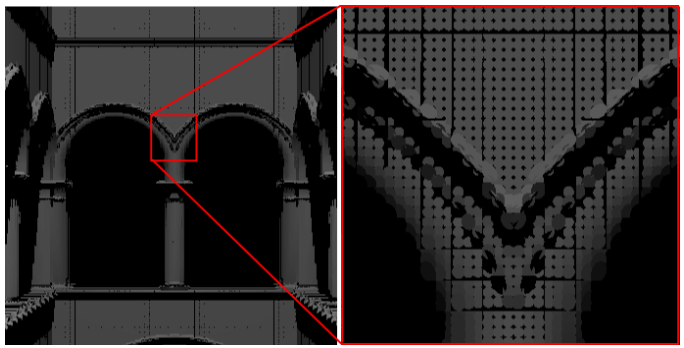
Complex scene with & without global illumination

Related Works: Point-Based Color Bleeding

Point-Based Approximate Color Bleeding by Per Christensen.

Subset of scene geometry is thoroughly sampled, creating point cloud.

Point cloud is sampled to determin incoming radiance.

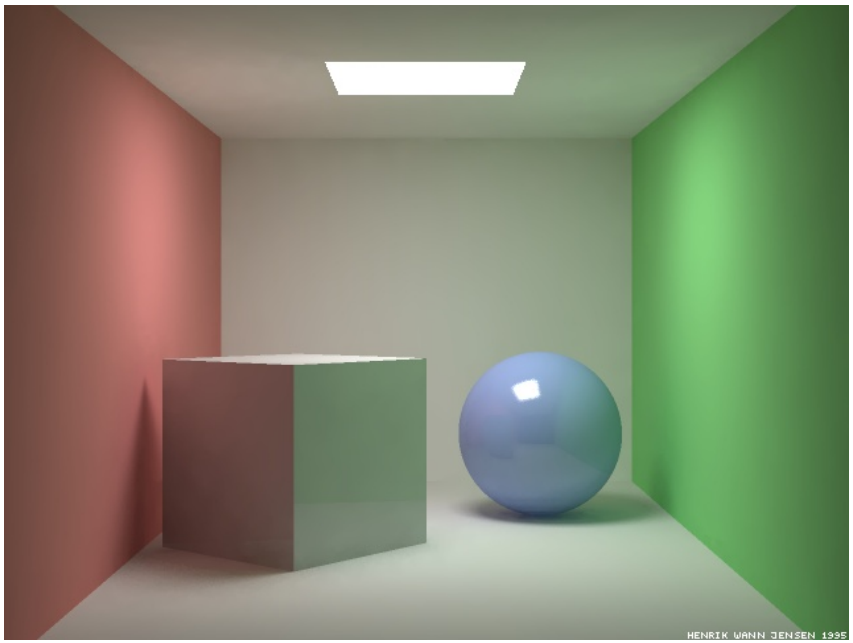


Randomly sampling a circle

Related Works: Photon Mapping

Photons cast from lights interact with the scene. At each hit, a photon can:

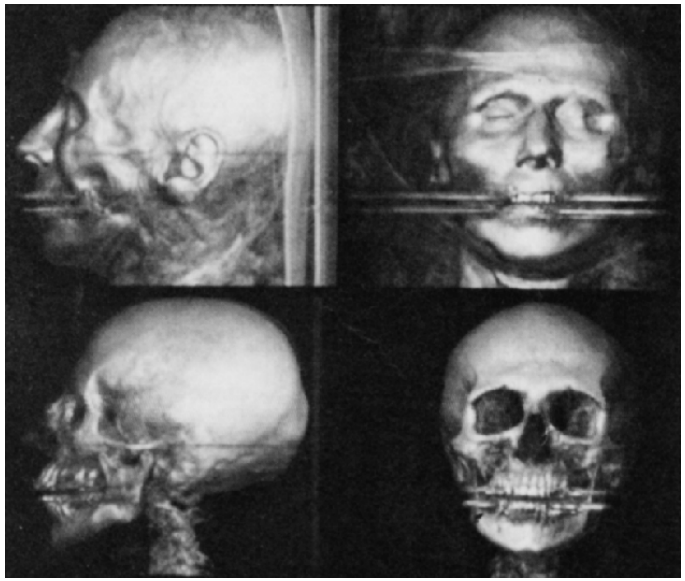
- Bounce
- Pass Through
- Be Absorbed



Related Works: Volume Rendering

Seminal research involved number of approaches...

- Polygonalization of straight voxels *(boxy)*
- Polygonal representation based on isosurfaces
- Opacity/Color arrays *(interpolation across voxels)*

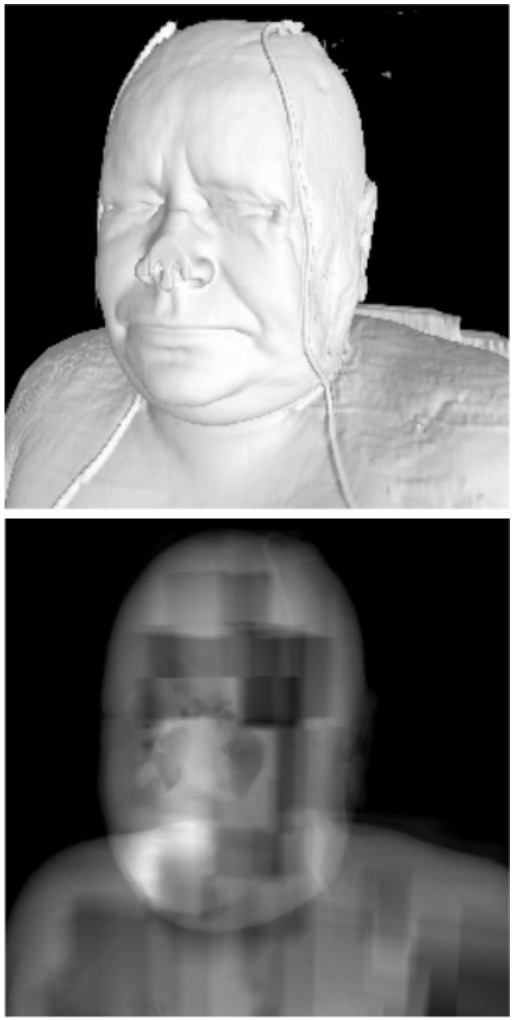


Renders of CT scan data

Related Works: Volume Rendering

Multi-resolution volumes help save on memory and allow for on-demand loading

Occlusion techniques involving volume acceleration structures help avoid extraneous computations



Renders of CT scan data

Outline

Point-Based Color Bleeding

1. Sample the scene and generate a point cloud
2. Perform ray tracing on regular geometry
3. Replace ambient estimates with a gather stage using surrounding point cloud

Extension Overview

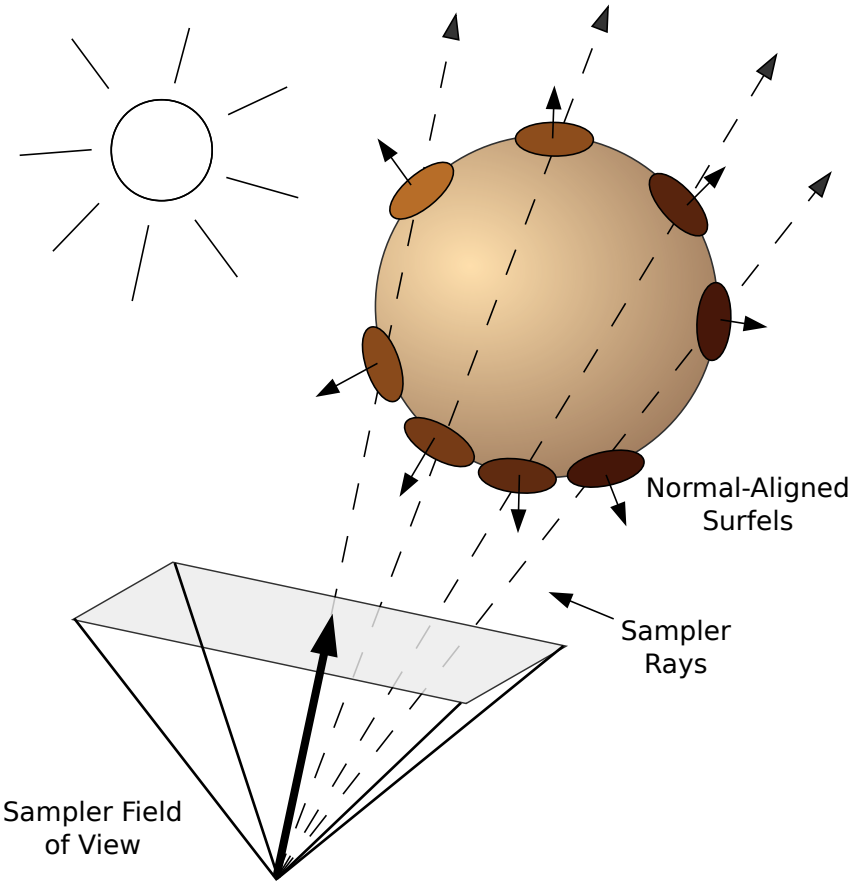
1. Sample the scene and generate a point cloud
2. Sample the participating media and evaluate scatter, absorbtion and direct lighting
3. Cast rays as normal
4. Orient hemispherical samples along the normals of the surfaces intersected
5. Model scatter-out and scatter-in properties during lighting gather stage

Sampling the Scene

Rays are cast and intersections with objects are recorded

Sampling "camera" is behind normal camera with wider field of view

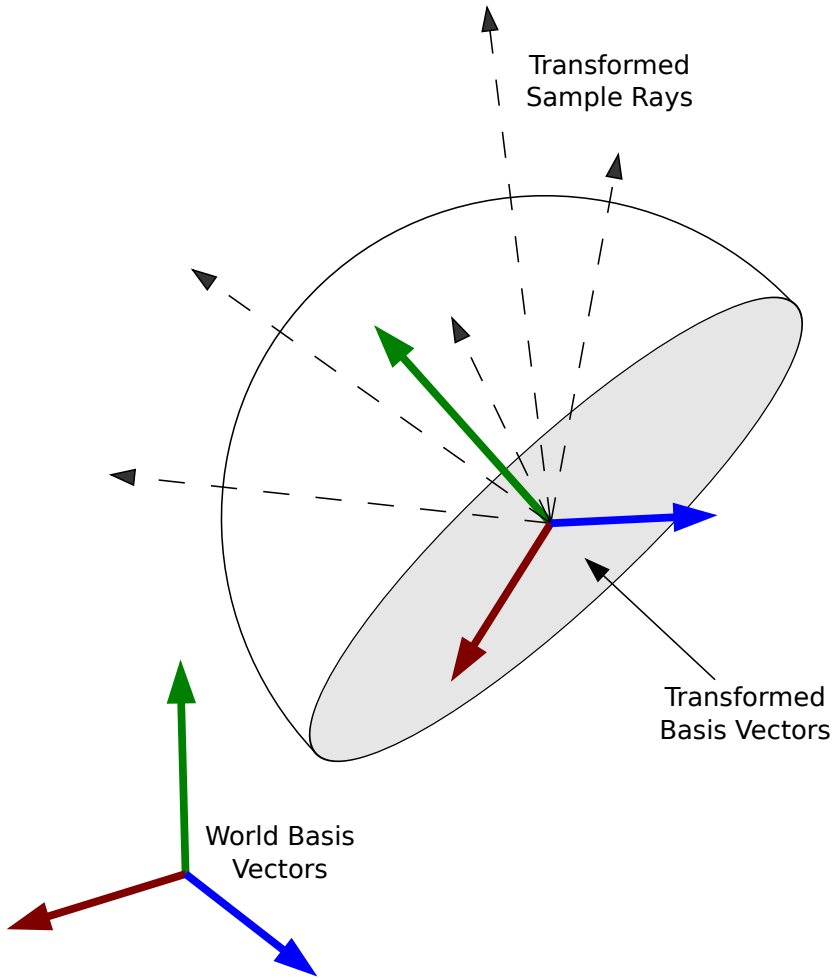
Surfels are oriented at the intersections aligned along the surface normal



Gathering Light

At render time, rays are cast from normal camera

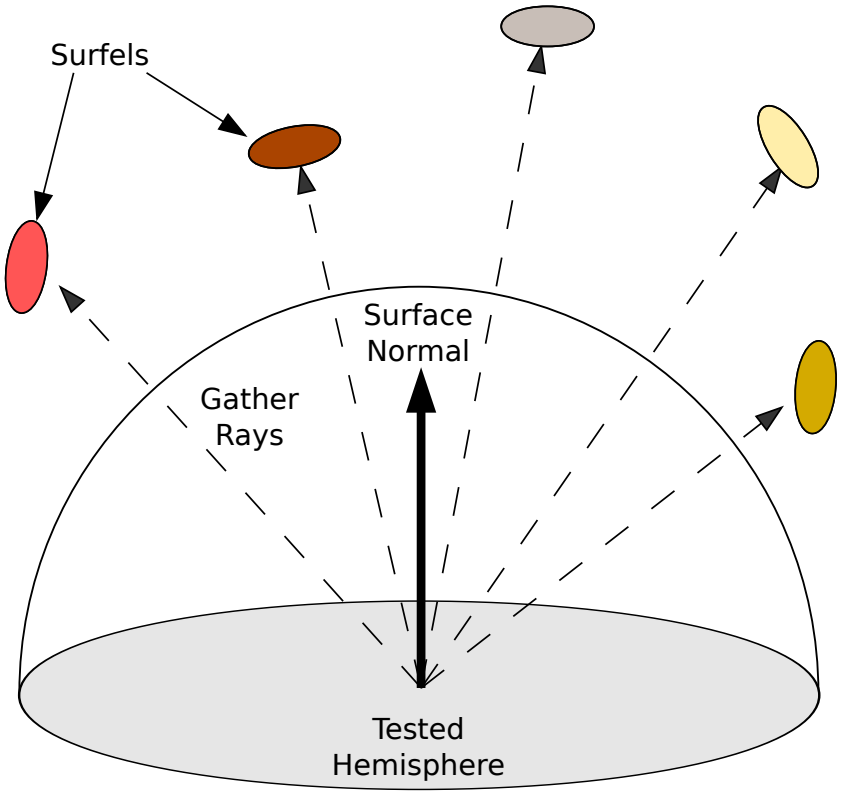
Samples are cast from intersecting surface oriented along its normal



Gathering Light

Samples cast into point cloud "gather" light via intersection

Samples are returned and used to evaluate the integral over the aligned hemisphere

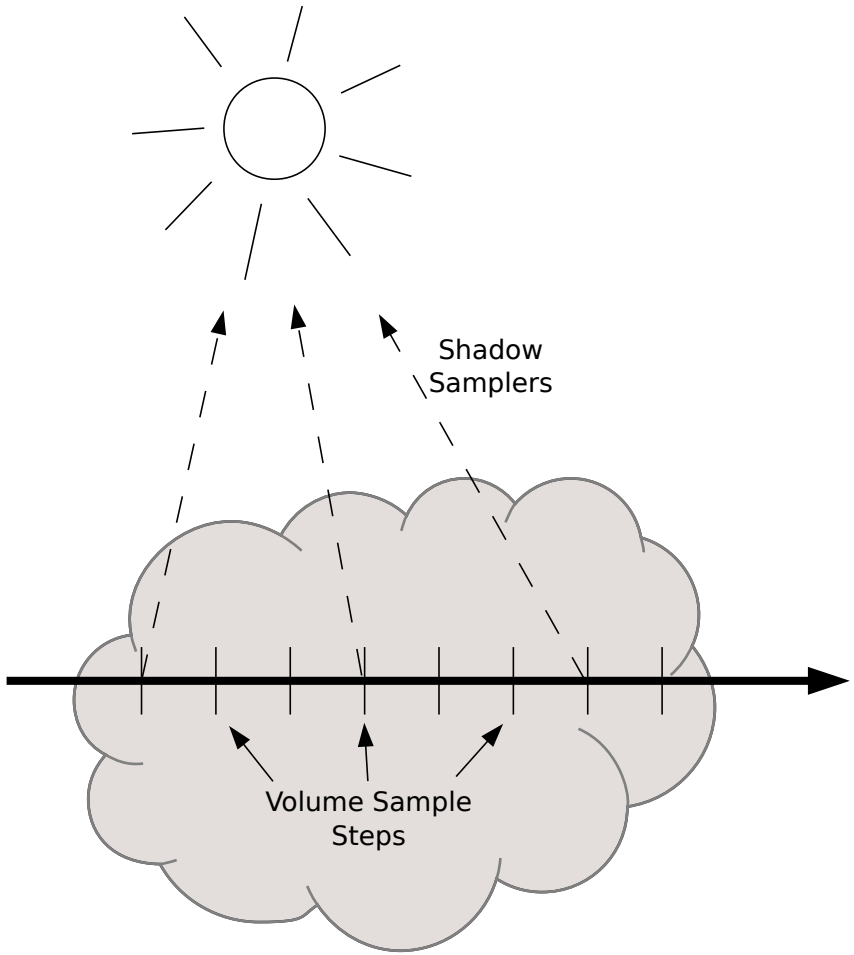


Integrating Volume Data

Samples are taken at discrete steps across the volume domain

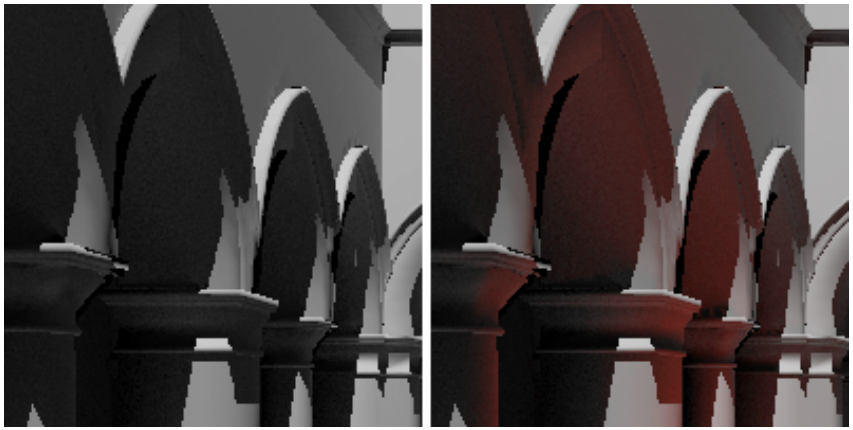
Scatter/Absorbtion/Lighting factors are averaged within each step

Lvoxels (spheres) instead of *surfels*



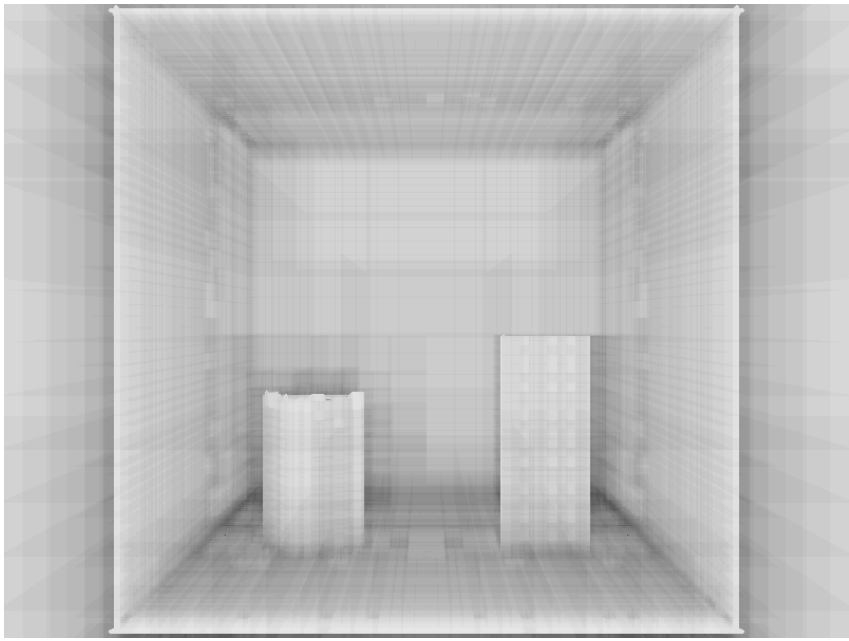
Integrating Volume Data - Scatter Out

Gather stage remains same, no changes necessary



Octree must be traversed from back-to-front in order to integrate properly

Must keep track of transmittance during traversal

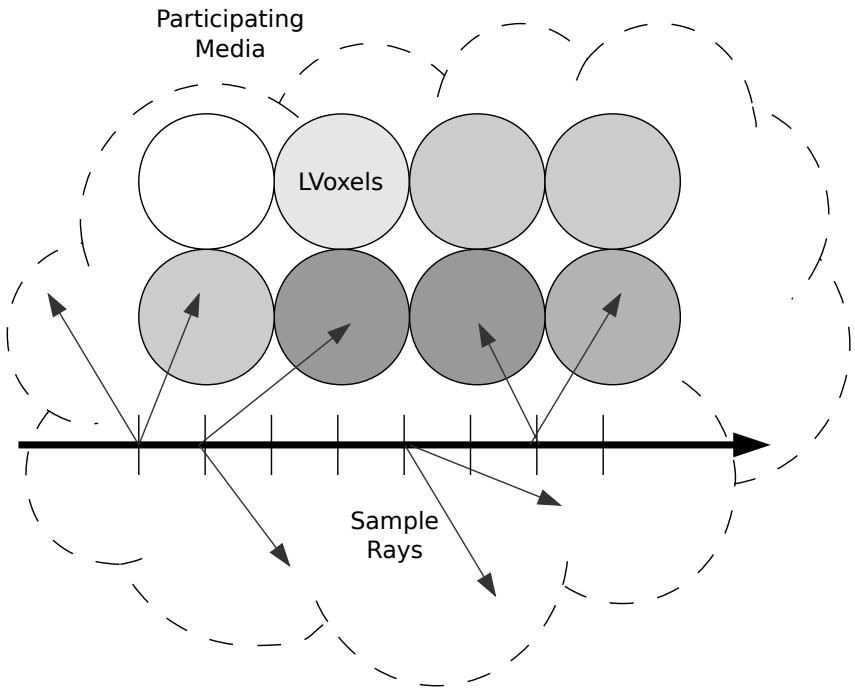


Integrating Volume Data - Scatter In

Modify integration code (normal volume rendering)

Add spherical sampling at each volume step

For increased performance, guarantee full distribution across multiple steps, not at each step

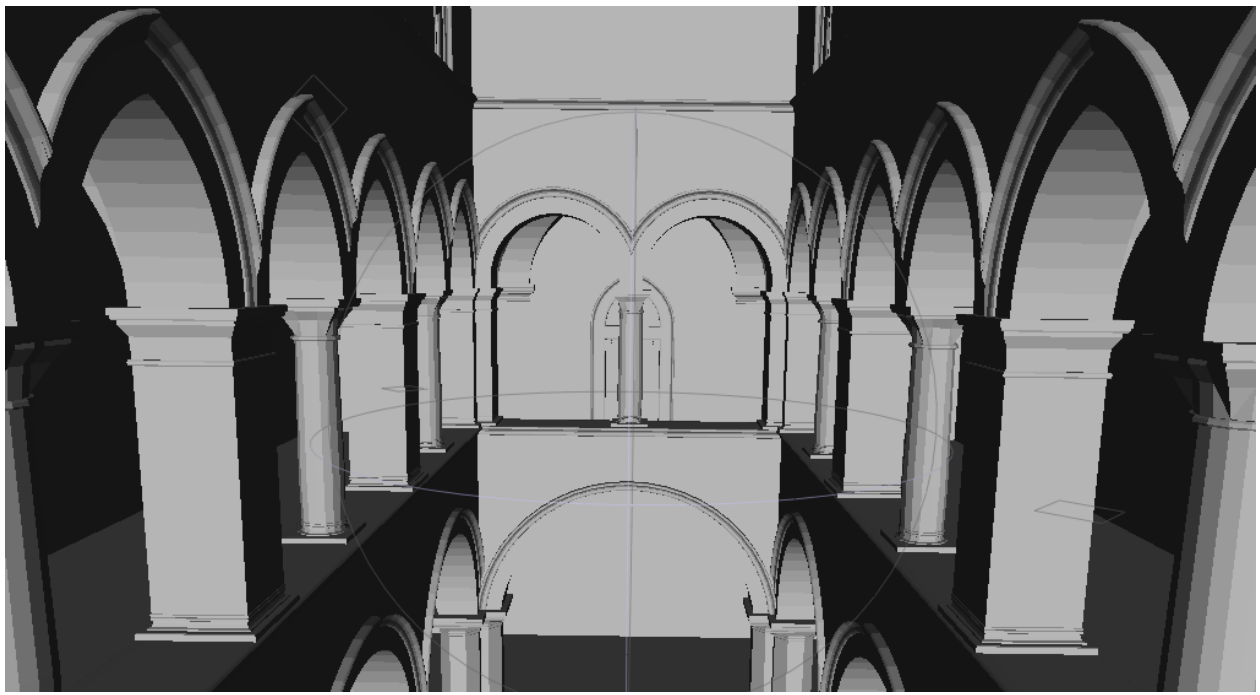


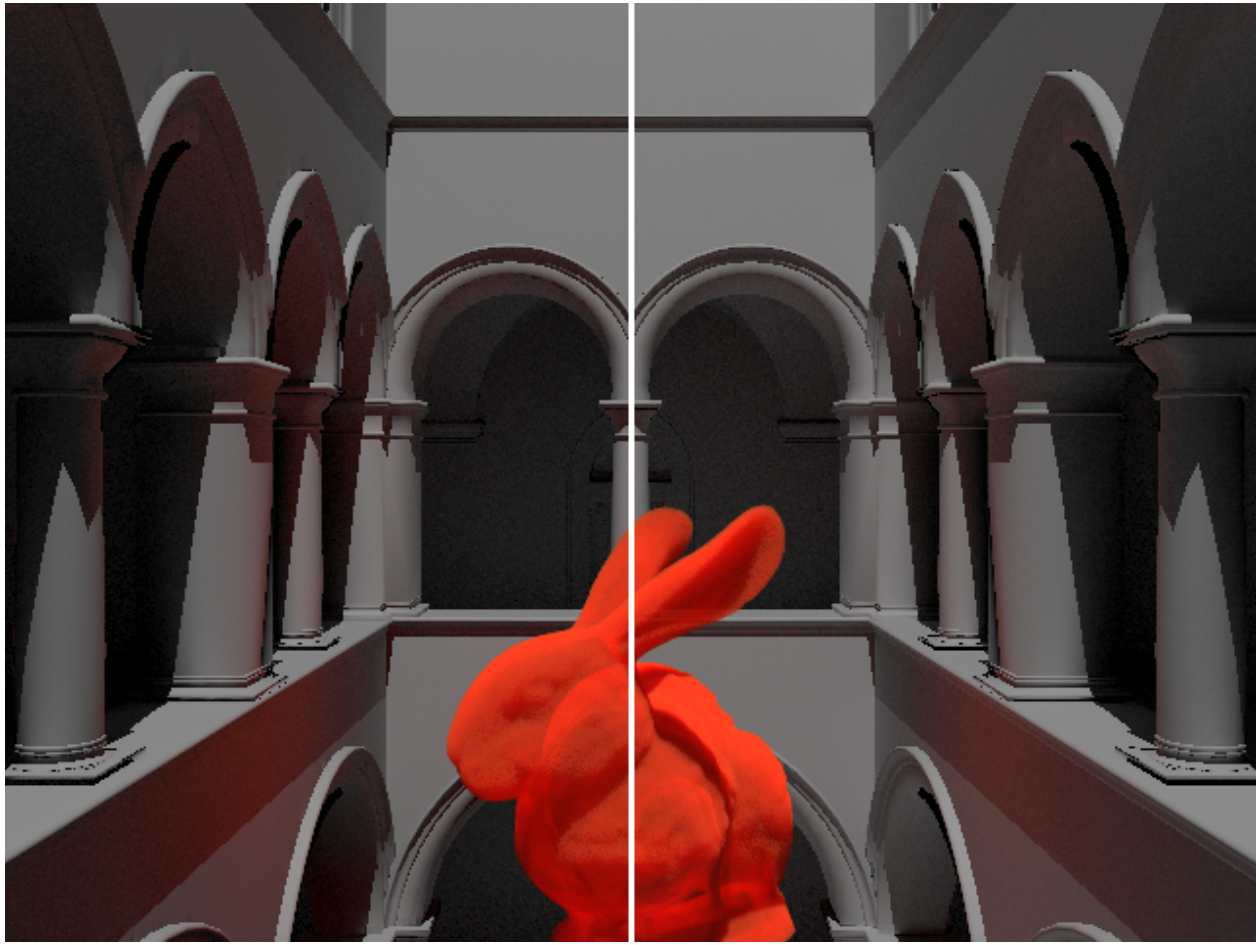
Outline

Results - Environment

Test scene involved:

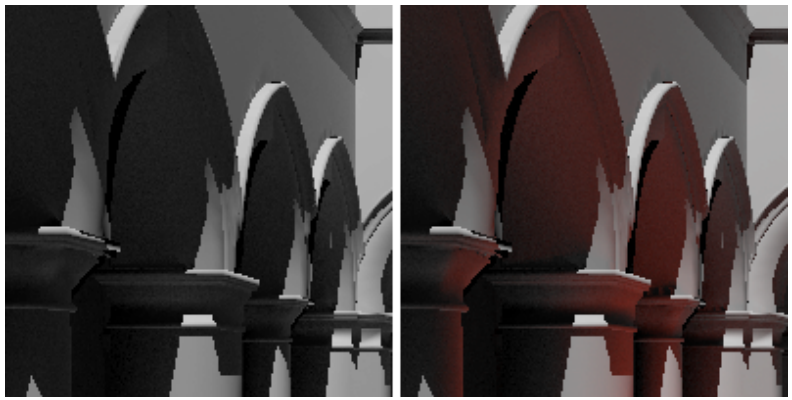
- 60,000 triangle Sponza Atrium Model
- Stanford Bunny *(Volume Data)*



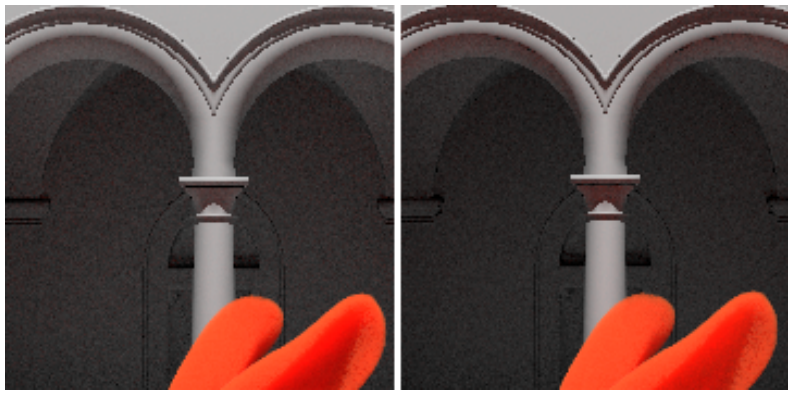


Results

Scene	Render Time (s)	Image Delta	Memory Overhead
Monte Carlo w/o PCB	3351 sec	NONE	NONE
Traditional PCB	348 sec	11.0%	390 Mb (4.0%)
Extended PCB	397 sec	4.8%	395 Mb (4.1%)



PBC vs PBCEX



Monte Carlo vs PBCEX

Outline

Future Work

1. Multiple bounce
2. Phase functions for volumes
3. Parallelism
4. Optimal Sampling
5. GPU Acceleration

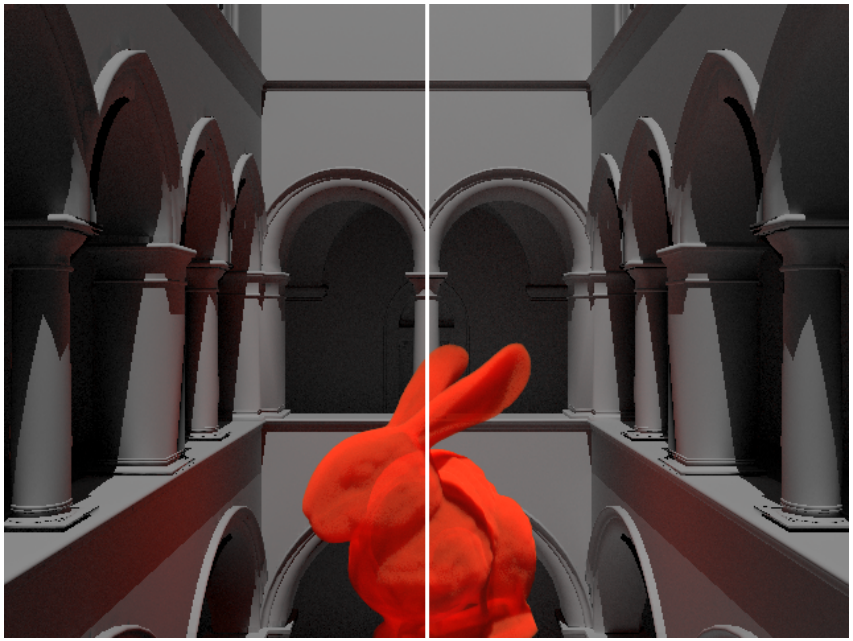
Outline

Conclusion

Modifications can be made to PCB to incorporate volume light contributions

Even simple implementations show great improvements in performance, nearly ten times faster than traditional Monte carlo

Image quality is comparable to Monte Carlo results



PBCEX *(left)* vs Monte Carlo *(right)*