Sequential Monte Carlo Adaptation in Low-Anisotropy Participating Media

Vincent Pegoraro        Ingo Wald        Steven G. Parker
Outline

• Introduction
• Related Work
• Monte Carlo Integration
• Radiative Energy Transfer
  – Control Variates
  – Importance Sampling
  – Adaptive Refinement
  – Estimate Evaluation
• Results
• Discussion and Future Work
• Conclusion
Introduction

• Motivation
  – Translucent materials
  – Gaseous volumes
  – Accurate simulation ⇒ scientific implications

• Applications
  – Movie and gaming industries
  – Safety-oriented research
  – Radiative energy transfer for gas dynamics
Related Work

- Irradiance & radiance caching
- Control variates
- Importance sampling
- Particle filtering / population Monte Carlo

⇒ Symbiotic sequential light transfer method
Monte Carlo Integration

- Estimating multi-dimensional integrals

- Stochastic nature $\Rightarrow$ noise

- Variance reduction techniques
Monte Carlo Integration

• Control Variates

\[ F = \int_D f(\vec{x})d\vec{x} = \int_D [f(\vec{x}) - g(\vec{x})] d\vec{x} + G \]

• Importance Sampling

\[ F = \int_D f(\vec{x})d\vec{x} = \int_D \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x})d\vec{x} = E \left[ \frac{f(\vec{X})}{p(\vec{X})} \right] \]
Monte Carlo Integration

- Combined Estimator

\[ F = E \left[ \frac{f(\tilde{X}) - g(\tilde{X})}{p(\tilde{X})} \right] + G \Rightarrow \hat{F} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(\bar{x}_i) - g(\bar{x}_i)}{p(\bar{x}_i)} + G \]

- Standard Deviation

\[ \sigma[\hat{F}] = \sqrt{\frac{1}{N} V \left[ \frac{f(\tilde{X}) - g(\tilde{X})}{p(\tilde{X})} \right]} = \frac{1}{N^{\frac{1}{2}}} \sigma \left[ \frac{f(\tilde{X}) - g(\tilde{X})}{p(\tilde{X})} \right] \]
Radiative Energy Transfer

- Radiative Transport Equation

\[
(\vec{\omega} \cdot \nabla)L(\lambda, \vec{x}, \vec{\omega}) = \sigma_a(\lambda, \vec{x})(L_e(\lambda, \vec{x}, \vec{\omega}) - L(\lambda, \vec{x}, \vec{\omega})) + \sigma_s(\lambda, \vec{x})(L_i(\lambda, \vec{x}, \vec{\omega}) - L(\lambda, \vec{x}, \vec{\omega}))
\]

where

\[
L_i(\lambda, \vec{x}, \vec{\omega}) = \int_{4\pi} L(\lambda, \vec{x}, \vec{\omega}_i) \Phi(\lambda, \vec{\omega}, \vec{\omega}_i) d\vec{\omega}_i
\]
Radiative Energy Transfer

- Solution to RTE

\[ L(\lambda, \vec{x}, \vec{\omega}) = e^{-\tau(\lambda, \vec{x}, \vec{x}_0)} L_b(\lambda, \vec{x}_0, \vec{\omega}) + \int_{\vec{x}}^{\vec{x}_0} e^{-\tau(\lambda, \vec{x}, \vec{x}')} (\sigma_a(\lambda, \vec{x}') L_e(\lambda, \vec{x}', \vec{\omega}) + \sigma_s(\lambda, \vec{x}') L_i(\lambda, \vec{x}', \vec{\omega})) d\vec{x}' \]

where

\[ \tau(\lambda, \vec{x}_a, \vec{x}_b) = \int_{\vec{x}_a}^{\vec{x}_b} \sigma_t(\lambda, \vec{x}) d\vec{x} \]

\[ \sigma_t = \sigma_a + \sigma_s \]
Radiative Energy Transfer

- Unbiased Ray-Integration

\[ L(\lambda, \vec{x}, \vec{\omega}) = \int_{\vec{x}}^{\vec{x}_\text{inf}} e^{-\tau(\lambda,\vec{x},\vec{x}')} \sigma_t(\lambda, \vec{x}') \]

\[ \left( \|\vec{x}' - \vec{x}\| < \|\vec{x}_0 - \vec{x}\| \Rightarrow L_t(\lambda, \vec{x}', \vec{\omega}) : L_b(\lambda, \vec{x}_0, \vec{\omega}) \right) d\vec{x}' \]

where

\[ L_t = (\sigma_a L_e + \sigma_s L_i) / \sigma_t \]
Radiative Energy Transfer

- Ray-Marching

\[ L(\lambda, \bar{x} + \Delta \bar{x}, \bar{\omega}) = e^{-\sigma_t(\lambda, \bar{x})\|\Delta \bar{x}\|} L(\lambda, \bar{x}, \bar{\omega}) + (1 - e^{-\sigma_t(\lambda, \bar{x})\|\Delta \bar{x}\|}) L_t(\lambda, \bar{x}, \bar{\omega}) \]

reformulated as

\[ I(\lambda, \bar{x}, \bar{\omega}) = \kappa_t(\lambda, \bar{x}) L(\lambda, \bar{x}, \bar{\omega}) + \kappa_s(\lambda, \bar{x}) L_i(\lambda, \bar{x}, \bar{\omega}) \]

\[ = \int_{\Omega_t \cup \Omega_s} \delta(\bar{\omega}_i, \Omega_t) \kappa_t(\lambda, \bar{x}) L(\lambda, \bar{x}, \bar{\omega}) + \delta(\bar{\omega}_i, \Omega_s) \kappa_s(\lambda, \bar{x}) L(\lambda, \bar{x}, \bar{\omega}_i) \Phi(\lambda, \bar{\omega}, \bar{\omega}_i) \, d\bar{\omega}_i \]
SMC in Low-Anisotropy Media

- Radiance estimates cached in 5D structure

- Low-anisotropy $\Rightarrow$ efficient integration

- Dynamic predicate functions without bias
SMC in Low-Anisotropy Media

- Control Variates
  - Low-cost read/write access
  - Efficient integration
  $\Rightarrow$ B-splines

- Representation
  - Cheap & continuous interpolants $\Rightarrow$ order 1
  - Adaptive grid / azimuth period / polar average
  - Update cell’s coefficient + integral + averages
SMC in Low-Anisotropy Media

• Control Variates
SMC in Low-Anisotropy Media

• Importance Sampling
  – Efficiency $\Rightarrow$ same resolution
  – CDF inversion $\Rightarrow$ low-orders
  – Continuity not crucial $\Rightarrow$ order 0

• Representation
  – Cell also contains scalar estimate of $|f-g|
  – Compute scalar PDF sample $\leftrightarrow f-g$ channels
  – Tree of partial PDF sums $\Rightarrow$ efficient sampling
SMC in Low-Anisotropy Media

• Importance Sampling
SMC in Low-Anisotropy Media

• Adaptive Refinement
  – Representation adapts to records population
  – Positional interpolation: continuity in efficiency
    ⇒ octree of spatial partitions
  – Initialization: single node & uniform sampling
  – Update: radiance estimate cached based on sample position and direction
  – Refinement criterion ⇒ subdivide, set counters
  – Inheritance ⇒ non-zero PDFs
SMC in Low-Anisotropy Media

• Adaptive Refinement
SMC in Low-Anisotropy Media

• Refinement Criterion
  – Threshold on average of records counters
  – Promote refinement in highly sampled regions
  – Controls inertia
  – Versatile structure quickly morphing to target
  – Unreliable predicates $\Rightarrow$ increased variance
  – Optimal value determined empirically
SMC in Low-Anisotropy Media

• Estimate Evaluation

1. EstimateRayIntegral()
2. (position, weight) = GetSamplePositionFromRayPDF();
3. if (position < mediumBoundary)
4.   cache = octree.GetCache(position);
5.   (direction, p) = cache.GetSampleDirection();
6.   G = cache.GetIntegralForIsotropicPhaseFunction();
7.   g = cache.GetRadiance(direction);
8.   g *= isotropicPhaseFunction.GetWeight(direction);
9.   radiance = TraceRay(position, direction);
10.  f = radiance * phaseFunction.GetWeight(direction);
11.  estimate = G + (f - g) / p;
12.  node = octree.GetNode(position);
13.  node.AddRecordToCache(direction, radiance);
14.  if (node.CriterionIsMet()) node.Refine();
15.  else
16.    estimate = TraceBackgroundRay();
17.  return estimate * weight;
• Root Mean Squared Error

Results
• Efficiency: \( \frac{1}{\text{variance} \times \text{cost}} \)
Results

10816 spp  8836 spp  3025 spp  4096 spp
Results

174 spp

250 spp

166 spp

256 spp
Results

1064 spp

1024 spp
## Results

- Characteristics

<table>
<thead>
<tr>
<th>Figure</th>
<th>Lucy</th>
<th>Cloud</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td>512</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Nodes</td>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>Leaves</td>
<td>12577</td>
<td>2489</td>
<td>6913</td>
</tr>
<tr>
<td>Cells</td>
<td>11005</td>
<td>2178</td>
<td>6049</td>
</tr>
<tr>
<td>Memory</td>
<td>23810048</td>
<td>3887712</td>
<td>6525440</td>
</tr>
<tr>
<td></td>
<td>0.978 GB</td>
<td>163.5 MB</td>
<td>274.8 MB</td>
</tr>
</tbody>
</table>
Discussion and Future Work

• Memory requirements
• Parallel implementation
• Extend to high-anisotropy media
• Optimal refinement criterion
• Combination with bidirectional approaches
Conclusion

• Symbiotic control variates / imp. sampling
• Dynamic predicates & marginal overhead ⇒ convergence and efficiency increase
• Inheritance strategy ⇒ well-behaved PDFs
• Online estimation and caching without bias ⇒ no pre-pass + visual imp. & scene driven
• Scene independent but exploit coherency
• Simple to implement and to tune
• Sequential adaptation ⇒ learning estimator
Acknowledgments

• U.S. Department of Energy

• Solomon Boulos, Dave Edwards, Thiago Ize and Peter Shirley

• Thank you!