State of the Art in Photon Density Estimation

Progressive Expectation–Maximization for Hierarchical Volumetric Photon Mapping

Wojciech Jarosz

(slides courtesy of Wenzel Jakob)
Motivation

Volumetric photon mapping

1. Trace photons

2. Radiance estimate

Issues

- high-frequency illumination requires many photons
- time spent on photons that contribute very little
- prone to temporal flickering

Tuesday 22 April 14
Motivation

Beam radiance estimate: 917K photons

Per-pixel render time
Motivation

Per-pixel render time

Beam radiance estimate: 917K photons

Our method: 4K Gaussians

Render time: 281 s

Per-pixel render time

Render time: 125 s

Per-pixel render time

Our approach:

• represent radiance using a Gaussian mixture model (GMM)
• fit using progressive expectation maximization (EM)
• render with multiple levels of detail

Jakob et al. 2011. Proceedings of EGSR.
Motivation

Beam radiance estimate: 4M photons

Our method: 16K Gaussians

Our approach:

• represent radiance using a Gaussian mixture model (GMM)
• fit using progressive expectation maximization (EM)
• render with multiple levels of detail
Related work

- Hierarchical photon mapping [Spencer and Jones 09]

- Photon relaxation [Spencer and Jones 09]

- Progressive photon relaxation [Spencer and Jones 13]

- Photon parameterisation for robust relaxation constraints [Spencer and Jones 13]

Feature detection & preservation challenging
Density estimation
Density estimation

Given photons $x_1, x_2, \ldots$

approximately determine their density $f$

**Nonparametric:**
- Count the number of photons within a small region

**Parametric:**
- Find suitable parameters for a **known** distribution
Gaussian mixture models

- Photon density modeled as a weighted sum of Gaussians:

\[ f(\mathbf{x} | \Theta) = \sum_{i=1}^{k} w_i \ g(\mathbf{x} | \Theta_i) \]
Gaussian mixture models

- Photon density modeled as a weighted sum of Gaussians:

\[ f(x | \Theta) = \sum_{i=1}^{k} w_i \ g(x | \Theta_i) \]

Unknown parameters $\Theta$:

1. Weights
2. Means
3. Covariance matrices
Maximum likelihood estimation

Approach: find the “most likely” parameters, i.e.

\[ \Theta^* := \arg\max_{\Theta} \prod_{i=1}^{n} f(x_i | \Theta) \]

- Estimated parameters
- Mixture model
- Photon locations
- Expectation maximization
Expectation maximization

- Two components:
  
  **E-Step:** establish soft assignment between photons and Gaussians
  
  **M-Step:** maximize the expected likelihood

- Finds a locally optimal solution  
  \[ \rightarrow \text{good starting guess needed!} \]

- Slow and scales poorly — \( O(n^2) \)  
  (where \( n \): photon count)
Each photon exerts a “pull” on nearby Gaussian components.

Accelerated EM by [Verbeek et al. 06]
Accelerated EM

Stored cell statistics:
- photon count
- mean position
- average outer product
Stored cell statistics:
- photon count
- mean position
- average outer product

Our modifications:
- better cell refinement
Progressive EM

Stored cell statistics:
- photon count
- mean position
- average outer product

Our modifications:
- better cell refinement
- progressive photons shooting passes
Progressive EM

Stored cell statistics:
- photon count
- mean position
- average outer product

Our modifications:
- better cell refinement
- progressive photons shooting passes
- reduced complexity
  \[ \mathcal{O}(n^2) \rightarrow \mathcal{O}(n \log n) \]
Pipeline overview

Progressive EM

- Shoot photons
- Initial guess

E → M → Shoot more photons → Refine partition

- converged? yes → Build Hierarchy
- converged? no → Render
Rendering

\[
\text{pixel value} = \sum_{i=1}^{k} \text{contrib}(i)
\]

\[
\text{contrib}(i) = \int_{a}^{b} g(r(t)|\tilde{\Theta}_i) e^{-\sigma t} t \, dt = C_0 \left[ \text{erf} \left( \frac{C_3 + 2C_2b}{2\sqrt{C_2}} \right) - \text{erf} \left( \frac{C_3 + 2C_2a}{2\sqrt{C_2}} \right) \right]
\]
Level of detail hierarchy

Agglomerative construction:

- Repeatedly merge nearby Gaussians based on their Kullback-Leibler divergence
Rendering

Criterion 1: bounding box intersected?

Criterion 2: solid angle large enough?

Criterion 3: attenuation low enough?

Example hierarchy:
BRE: 1M Photons

23 + 192 = 215 s
Our method: 4K Gaussians
(fit to 1M photons)

\[35 + 24 = 59 \text{ s}\] (3.6×)
BRE: 18M Photons

507 + 609 = 1116 s

Tuesday 22 April 14
Our method: 64K Gaussians (fit to 18M photons)

868 + 66 = 934 s (1.2×)
**BRE:** 4M Photons

89 + 638 = 727 s
Our method: 16K Gaussians

330 + 127 = 457 s

(1.6x)
Temporal Coherence

- Feed the result of the current frame into the next one
  - Faster fitting, no temporal noise
Scene 1: **BumpySphere**

Volume caustics from a rotating light source
GPU-based rasterizer:

- Anisotropic Gaussian splat shader: 30 lines of GLSL
- Gaussian representation is very compact (4096-term GMM requires only ~240KB of storage)
Conclusion

• Rendering technique based on parametric density estimation
• Uses a progressive and optimized variant of accelerated EM
• Compact & hierarchical representation of volumetric radiance
• Extensions for temporal coherence and real-time visualization

Questions?