Denoising Monte Carlo Sequences
This time with: Recurrent Auto-Encoders and Temporal Gradient Estimation
Speaker: Nick Heppert - 20186505
Microfacet model and Microfacet-based BRDF

- Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations
- Fast Global Illumination with Discrete Stochastic Microfacets Using a Filterable Model
Extracting Microfacet-based BRDF Parameters from Arbitrary Materials with Power Iterations

Contribution

Idea:
- Find the NDF
- Approximize the Fresnel term

Properties:
- Robustness
- Simplicity
- Speed
- Reproducibility
Fast Global Illumination with Discrete Stochastic Microfacets Using a Filterable Model

Idea:

If a footprint cover a large surface,
- individual glint are not noticeable;
- average contribution

In practice:

Filterable model preafer for
- material far from camera;
- Several bounce in global illumination
Motivation

- High samples per pixels (spp) → A lot of time
- Cut down time by creating low samples images → Noisy
- Desired: Same Performance
- + Sequences: Consistent through time
Papers

- Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder
- Gradient Estimation for Real-Time Adaptive Temporal Filtering
Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder

C. R. A. Chaitanya et al.
Environment

- Interactive Path Tracer
  - 1spp (!) for 1080p@30FPS
  - Next event estimation
  - Use rasterization to store shading attributes in G-Buffer
  - OptiX

- No depth of field or motion blur

- Rasterization to generate G-Buffer: 4 scalar values per pixel
  - View-space shading normals (2D)
  - Linearized depth (1D)
  - Material roughness (1D)

- + RGB (3D) → 7 scalar values per pixel
1 spp

Direct Shadow = 1 Ray

Indirect Shadow = 2 rays
Denoising Autoencoder (DAE)
Recurrent Denoising Autoencoder (RAE)
Training Setup

- **Data augmentation for sequence**
  - 1024 x 1024 → 128 x 128 randomly sampled
  - Random beginning of sequence
  - Forward and backward replay
  - Random stop
  - Random rotation

- **Loss L1:**
  - pixel-wise → Single images
    \[ \mathcal{L}_t = \frac{1}{N} \sum_{i=1}^{N} (|\nabla P_i - \nabla T_i|) \]
  - pixel-wise gradient → Edges
    \[ \mathcal{L}_t = \frac{1}{N} \sum_{i=1}^{N} (|P_i - T_i|) \]
  - pixel-wise time derivative → Coherent over time
    \[ \mathcal{L}_t = \frac{1}{N} \sum_{i=1}^{N} \left( \left| \frac{\partial P_i}{\partial t} - \frac{\partial T_i}{\partial t} \right| \right) \]
Gradient Estimation for Real-Time Adaptive Temporal Filtering

C. Schied et al.
Problem

- Static scenes
- Moving scenes: ghosting & temporal lagging
  - Constant temporal accumulation factor $\alpha$
  - Shading samples from previous frame re-used
  - Sudden change $\rightarrow$ not relevant anymore

→ Solution:

- Adaptively change temporal accumulation factor per frame per pixel
  - Fast response time for sudden changes
  - Aggressive re-use for static regions
Back-Propagation

\[ \hat{c}_i(x) = \alpha \cdot c_i(x) + (1 - \alpha) \cdot \hat{c}_{i-1}(\hat{x}) \]

where

\[ \hat{c}_i \] new temporally filtered frame

\[ i \] current timestep

\[ x \] pixel position in current frame

\[ c_i \] current frame

\[ \hat{c}_{i-1} \] previously temporally filtered frame

\[ \hat{x} \] pixel position in previous frame
Back-Propagation

\[
\hat{c}_i(x) = \alpha \cdot c_i(x) + (1 - \alpha) \cdot \hat{c}_{i-1}(\hat{x})
\]

Big \( \alpha \rightarrow \) Current Frame
Small \( \alpha \rightarrow \) Previous Frame
Temporal Gradient

Basic Version:

\[ \delta_{i,j} := f_i(\overrightarrow{G_{i-1,j}}) - f_{i-1}(\overrightarrow{G_{i-1,j}}) \]

Extended Version:

\[ \delta_{i,j} := f_i(\overrightarrow{G_{i-1,j}, \xi_{i,j}}) - f_{i-1}(\overrightarrow{G_{i-1,j}, \xi_{i-1,j}}) \]
Controlling the Temporal Accumulation Factor

Normalized History Weight:

$$\lambda(p) := \min \left(1, \frac{|\delta_i(p)|}{\Delta_{i,j}(p)} \right)$$

$\Delta_{i,j}$: Subset

Normalizer:

$$\Delta_{i,j} := \max \left(f_i(\vec{G}_{i-1}, j, \xi_{i-1,j}), f_{i-1}(\vec{G}_{i-1}, j, \xi_{i-1,j}) \right)$$

Adaptive Temporal Accumulation Factor:

$$\alpha_i(p) := (1 - \lambda(p)) \cdot \alpha + \lambda(p)$$

$$\hat{c}_i(x) = \alpha \cdot c_i(x) + (1 - \alpha) \cdot \hat{c}_{i-1}(\vec{x})$$
Evaluation: RAE vs. A-RAE

- **RAE**
  - no temporal accumulation

- **A-RAE**
  - Adaptive temporal accumulation

- **Trade-Off**
  - Image Quality
  - Temporal ghosting & lagging

- **Inference Time** RAE: 191ms
  - Adaptive temporal accumulation: 2ms
  - Negligible
A-SVGF (ours, slowed)
A-SVGF (ours, slowed)
Summary

● Baseline auto-encoder network
● + Skip connections
● + Recurrent blocks
● + Adaptive temporal accumulation
● → 1spp real-time
  ○ Quake 2
  ○ http://brechpunkt.de/q2vkpt/ (open source)
  ○ https://youtu.be/vY0W3MkZFs4 (closed source)
Quiz

1. Which buffer is not used to train the recurrent auto-encoder (RAE)?
   a. View-space shading normals (2D)
   b. Raw texture color (3D)
   c. Material roughness (1D)
   d. Linearized depth (1D)

2. When the temporal gradient \( \delta \) increases which is frame is getting preferred?
   a. previous frame
   b. current frame

3. When increasing the temporal accumulation factor \( \alpha \) which frame is getting preferred?
   a. previous frame
   b. current frame