General and Robust Error Estimation and Reconstruction for Monte Carlo Rendering

Pablo Bauszat\textsuperscript{1}, Martin Eisemann\textsuperscript{1,2}, Elmar Eisemann\textsuperscript{2}, Marcus Magnor\textsuperscript{1}

\textsuperscript{1} Computer Graphics Lab, TU Braunschweig, Germany
\textsuperscript{2} Delft University of Technology, Netherlands
Monte Carlo Rendering

- Today’s industry standard
- General and unbiased
- Covers variety of natural phenomena

- Requires extensive sampling
  - Pixel (2D integral)
  - Camera lens (2D integral)
  - Time (1D integral)
  - Global illumination (2D integral per bounce)
  - ... and more ...
Filtering

Noisy

Reference

Uniform filter (small)

Uniform filter (large)

Adaptive filtering
Adaptive Reconstruction

• **Filter bank**
  • Set of filters with different properties
  • Select best filter on a per-pixel level
Problem statement

How to choose the best filter from the set for a pixel?
Previous work

Overbeck et al. 2009

Li et al. 2012

Rousselle et al. 2011/2012/2013

Kalantari et al. 2013

Moon et al. 2014
Limitations of previous work

• Filter selection based on noisy image
• Often tailored for specific filters
• Switching filters may cause seams
Our method
Insights

Our method is based on three key insights:

1. Filter selection is often more crucial than sampling rate
2. Filter error is locally smooth for most image regions
3. Often multiple filters are close-to-optimal choices
1. Filter selection is often more crucial than sampling rate

<table>
<thead>
<tr>
<th>Filter Bank</th>
<th>SPP</th>
<th>MSE $^{-3}$</th>
<th>Best Choice</th>
<th>MSE $^{-3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Gaussian and 4 Joint Bilateral</td>
<td>32</td>
<td>12.3</td>
<td>32 spp</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.6 MSE$^{-3}$</td>
<td>(x 7.7)</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>2.3</td>
<td>16 spp</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.3 MSE$^{-3}$</td>
<td>(x 5.3)</td>
</tr>
</tbody>
</table>

Recently employed by [Li2012] and [Rousselle2013]
1. Filter selection is often more crucial than sampling rate

<table>
<thead>
<tr>
<th>Scene</th>
<th>SURE 32 spp</th>
<th>Best choice 32 spp</th>
<th>Best choice 16 spp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference</td>
<td>12.327</td>
<td>1.605 (x 7.7)</td>
<td>2.344 (x 5.3)</td>
</tr>
<tr>
<td>Sibenik</td>
<td>0.758</td>
<td>0.157 (x 4.8)</td>
<td>0.258 (x 2.9)</td>
</tr>
<tr>
<td>Toasters</td>
<td>0.187</td>
<td>0.096 (x 1.9)</td>
<td>0.156 (x 1.2)</td>
</tr>
<tr>
<td>San Miguel</td>
<td>16.880</td>
<td>6.419 (x 2.6)</td>
<td>9.831 (x 1.7)</td>
</tr>
</tbody>
</table>

Mean squared error (MSE) * 10^{-3} – Same filter bank
Insights

Our method is based on three key insights:

1. Filter selection is often more crucial than sampling rate
2. Filter error is locally smooth for most image regions
3. Often multiple filters are close-to-optimal choices
2. Error smoothness – Gaussian filters

Gaussian $\sigma=7$  
Gaussian $\sigma=11$  
Gaussian $\sigma=13$
2. Error smoothness – Guided Image Filtering [He2010]

Guided radius=4
Guided radius=8
Guided radius=16
Insights

Our method is based on three key insights:

1. Filter selection is often more crucial than sampling rate

2. Filter error is locally smooth for most image regions

3. Often multiple filters are close-to-optimal choices
3. Often multiple filters are close-to-optimal choices
3. Often multiple filters are close-to-optimal choices

Regularized selection via ground truth
- MSE down to 8.0% from noisy image
- Variations in selection are penalized
What do we learn from the insights?

- Filter selection is crucial
- Filter error is piece-wise smooth
- Non-optimal filter selection does not imply large error
Our Method

1. Filter bank generation
2. Sparse reference pixels
3. Sparse error computation
4. Dense error interpolation
5. Filter compositing
1. Filter bank generation

Filter bank generation involves creating a set of filters that can be applied to a signal or image to extract relevant features. The sample budget is allocated to generate these filters, with options for different sample rates such as 16 spp and 30 spp. The filter bank consists of multiple filters, including Filter 1, Filter 2, ..., up to Filter n, each at different stages of the process.
2. Sparse reference pixels

Sample Budget

16 spp

Filter bank

Filter 1

Filter 2

... Filter n

128 spp per reference pixel

22
3. Sparse error computation

- Serves as reference
- Used to estimate filter error
- Low-variance estimator

Sample Budget

16 spp

128 spp per filter cache

Filter bank

Filter 1

Filter 2

\ldots

Filter n
4. Dense error interpolation

• Interpolation of sparse error estimate (per filter)
4. Dense error interpolation

- Best selection from interpolated error leads to seams
5. Filter compositing

Globally optimize filter selection (seek labeling $L$)

$$\arg\min_L E(L) = E_{Data}(L) + \lambda \cdot E_{regularizer}(L)$$

Data term
- Local error maps
- Minimize MSE

Regularization term
- Solution image gradients
- Avoid seams
5. Filter compositing

- Solve by **graph-cuts**
  
  "Fast approximate energy minimization via graph cuts", Boykov et al. 2001
5. Filter compositing

- Solve by **graph-cuts**
  
  „Fast approximate energy minimization via graph cuts“, Boykov et al. 2001
Our Method

1. Filter bank generation
2. Sparse reference pixels
3. Sparse error computation
4. Dense error interpolation
5. Filter compositing
Bells & Whistles

• Choice of regularization in filter compositing
• Integration of high-quality radiance values (not included the filter bank)
• Select „best“ pixels for sparse error estimate
Adaptive placement of sparse estimates

- Required for highly variant error regions
- Reduces residual variance in radiance estimate
Results
Results – San Miguel

Global illumination

MC 4096 spp
15,449 sec

MC 32 spp
146 sec

Our result 32 spp
146 + 13 sec
Results - Chess

Depth-of-field

MC 4096 spp
1,492 sec

MC 8 spp
9 sec

Our result 8 spp
9 + 29 sec
Results - Poolball

Motion blur

MC 4096 spp 10,989 sec
MC 8 spp 25 sec
Our result 8 spp 25 + 25 sec
Results - Teapot

Glossy materials

MC 4096 spp 3,619 sec
MC 16 spp 14 sec
Our result 16 spp 14 + 8 sec
Results - Dragon

Participating media

MC 4096 spp
12,464 sec

MC 32 spp
95 sec

Our result 32 spp
95 + 12 sec
Results - Timings

Intel Core i7-2600, 3.40 GHz, 16 GB RAM, NVIDIA GeForce 780 GTX, Windows 7 64-bit
Rendered with PBRT 2 path tracing.
Error analysis

Two error sources

Interpolation error

Residual variance in radiance
Results – GID
(„Removing the Noise in Monte Carlo Rendering with General Image Denoising Algorithms”, Kalantari et al. 2013)

GID (8 spp)  Ours (8 spp)  GID (32 spp)  Ours (32 spp)  Reference

Chess scene

MSE=2.6491  MSE=1.38179  MSE=2.4006  MSE=0.8962
SSIM=0.9516  SSIM=0.9874  SSIM=0.9558  SSIM=0.9948
Results – RD
(„Robust Denoising using Feature and Color Information”, Rousselle et al. 2013)

Dragon scene

<table>
<thead>
<tr>
<th></th>
<th>RD (16 spp)</th>
<th>Ours (16 spp)</th>
<th>RD (32 spp)</th>
<th>Ours (32 spp)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>13.6693</td>
<td>10.1914</td>
<td>9.3887</td>
<td>7.8838</td>
<td></td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9654</td>
<td>0.9599</td>
<td>0.9781</td>
<td>0.9768</td>
<td></td>
</tr>
</tbody>
</table>
Error sparsity

• Sparsity of error maps in transform domain (CDF 9/7 wavelets)
• Redundant information

<table>
<thead>
<tr>
<th></th>
<th>Gaussian $\sigma=7$</th>
<th>Gaussian $\sigma=11$</th>
<th>Gaussian $\sigma=13$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guided radius=4</td>
<td>86.46%</td>
<td>88.58%</td>
<td>89.86%</td>
</tr>
<tr>
<td>Guided radius=8</td>
<td>81.34%</td>
<td>87.07%</td>
<td>89.43%</td>
</tr>
<tr>
<td>Guided radius=16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NLM</td>
<td>60.06%</td>
<td>67.35%</td>
<td>73.62%</td>
</tr>
<tr>
<td>BM3D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLS-GSM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results – SURE [Stein1981]

<table>
<thead>
<tr>
<th>Scene</th>
<th>Noisy MSE</th>
<th>SURE MSE</th>
<th>Our approach MSE</th>
<th>Reference MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sibenik</td>
<td>6.0644</td>
<td>0.7681</td>
<td>0.3556</td>
<td>0.9829</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9066</td>
<td>0.9643</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Eurographics 2015
Conclusion

• Summary
  • Redistributing samples can improve filter selection
  • Global filter selection removes image seams

• Benefits
  • Works with arbitrary filters
  • No assumptions regarding scene and image content
  • Easy integration into existing rendering frameworks
Outlook

• Investigate other interpolation schemes

• Adaptive sampling feedback loop

• Temporal coherence
Thank you for your attention!