Adaptive sampling and denoising

UCSD CSE 168
Rendering
Tzu-Mao Li
Today: more variance reduction ideas
Idea: can we just blur the image?

quiz: why doesn’t a Gaussian blur work? can we do better?
Need adaptive **denoising**:
large filter at smooth & noisy region,
small filter near edges
Need adaptive **sampling**: allocate more samples to regions that are still noisy.
Adaptive sampling and denoising

- noisy image
- denoised
- estimate error

more samples in high error regions
Adaptive sampling and denoising

renderer

noisy image

denoised

estimate error

more samples in high error regions
Denoising Monte Carlo rendering

**quiz:** what are the differences between denoising a rendering and denoising a real image?
Two main differences between Monte Carlo denoising & image denoising

- noise in rendering is usually *spatially-varying*
- some pixels are more noisy than others
Two main differences between Monte Carlo denoising & image denoising

- noise in rendering is usually \textit{spatially-varying}
- some pixels are more noisy than others
- we can have extra information from the renderer

\begin{itemize}
  \item color
  \item normal
  \item albedo
  \item depth
\end{itemize}
Estimating the amount of noise using Monte Carlo samples

how do we know how noisy each pixel is?
Estimating the amount of noise using Monte Carlo samples

- estimate the variance of the pixel color

quiz: how do we estimate the variance?
Denoising Monte Carlo rendering

![Denoising Monte Carlo rendering](image)

- color & variance
- "G-buffer"
- normal
- albedo
- depth
- denoised image
Typically, rendering denoising is done using an edge-aware filter.

\[ G\left( \frac{\|x - x'\|^2}{\sigma_s^2} \right) \ast G\left( \frac{\|C - C'\|^2}{\sigma_c^2} \right) \]

- **input**
- **kernel**
- **blurred output**

where:
- \(G\) is a Gaussian function.
- \(\sigma_s\) and \(\sigma_c\) are the standard deviations for position and color, respectively.

Examples of filters include:
- **bilateral**
- **Gaussian**
- **noisy**
Typically, rendering denoising is done using an edge-aware filter.

**quiz:** how do we modify bilateral filtering to incorporate extra images? e.g., bilateral filter

$$G\left(\frac{\| x - x' \|^2}{\sigma_s^2}\right) \ast G\left(\frac{\| C - C' \|^2}{\sigma_c^2}\right)$$

- **position**
- **color**

- input
- kernel
- blurred output

- noisy
- Gaussian
- bilateral
Cross-bilateral filters: use extra images for edge detection

\[ G(\frac{\|x - x'\|^2}{\sigma_s^2}) \ast G(\frac{\|C - C'\|^2}{\sigma_c^2}) \ast G(\frac{\|N - N'\|^2}{\sigma_n^2}) \ast G(\frac{\|a - a'\|^2}{\sigma_a^2}) \ast G(\frac{\|d - d'\|^2}{\sigma_d^2}) \ast \ldots \]

- pixel
- position
- color
- normal
- albedo
- depth

Flash Photography Enhancement via Intrinsic Relighting

2004

Elmar Eisemann*
Frédo Durand
What do we do with spatially-varying noise?

\[
G\left(\frac{\|x - x'\|^2}{\sigma_s^2}\right) \ast G\left(\frac{\|C - C'\|^2}{\sigma_c^2}\right) \ast G\left(\frac{\|N - N'\|^2}{\sigma_n^2}\right) \ast G\left(\frac{\|a - a'\|^2}{\sigma_a^2}\right) \ast G\left(\frac{\|d - d'\|^2}{\sigma_d^2}\right) \ast \ldots
\]

- pixel position
- color
- normal
- albedo
- depth
What do we do with spatially-varying noise?

\[
G\left(\frac{\|x - x'\|^2}{\sigma_x^2}\right) \ast G\left(\frac{\|C - C'\|^2}{\sigma_C^2}\right) \ast G\left(\frac{\|N - N'\|^2}{\sigma_N^2}\right) \ast G\left(\frac{\|a - a'\|^2}{\sigma_a^2}\right) \ast G\left(\frac{\|d - d'\|^2}{\sigma_d^2}\right) \ast \ldots
\]

adjust the filter support to accommodate noise!
Two ways to choose the filter size

2012

SURE-based Optimization for Adaptive Sampling and Reconstruction
Tzu-Mao Li
Yu-Ting Wu
Yung-Yu Chuang
National Taiwan University

used in movie production* e.g., Big Hero 6

*Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination
Christoph Schied
Nvidia
Karlsruhe Institute of Technology
Anton Kaplanyan
Chris Wyman
Anjul Patney
Nvidia
Chakravarty R. Alla Chaitanya
Nvidia
University of Montreal
McGill University

John Burgess
Shiqiu Liu
Nvidia
Carsten Dachsbacher
Karlsruhe Institute of Technology
Aaron Lefohn
Marco Salvi
Nvidia

used in video games e.g., Dying Light 2, Watch Dog Legion

*they used a version with non-local means filtering [Rousselle 2013]
Two ways to choose the filter size

2012
SURE-based Optimization for Adaptive Sampling and Reconstruction
Tzu-Mao Li       Yu-Ting Wu       Yung-Yu Chuang
National Taiwan University

used in movie production* e.g., Big Hero 6

they used a version with non-local means filtering [Rousselle 2013]

2017
Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination
Christoph Schied
NVIDIA
Karlsruhe Institute of Technology

Anton Kaplanyan
Chris Wyman
Anjul Patney
NVIDIA

Chakravarty R. Alla Chaitanya
NVIDIA
University of Montreal
McGill University

John Burgess
Shiqiu Liu
NVIDIA

Carsten Dachsacher
Karlsruhe Institute of Technology

Aaron Lefohn
Marco Salvi
NVIDIA

used in video games e.g.,
Dying Light 2, Watch Dog Legion

Figure 1: Our filter takes (left) 1 sample per pixel path-traced input and (center) reconstructs a temporally stable 1530×1080 image in just 39 ms. Compare to (right) a 2000 samples per pixel path-traced reference. Insets compare our input, our filtered results, and a reference on two regions, and show the impact filtered global illumination has over just direct illumination. Given the noisy input, notice the similarity to the reference for glossy reflections, global illumination, and direct soft shadows.
Idea: estimate the filtering error for different filtering parameters

- very low variance, very high bias
- low variance, low bias
- very high variance, very low bias
Estimating filter error
Estimating filter error

\[ y_i \]

\[ \sigma_i^2 \]

mean

variance
Estimating filter error

\[ x_i, y_i, \sigma^2_i \]

converged  \[ \rightarrow \]  mean  \[ \rightarrow \]  variance
Estimating filter error

$$f(y_i) = \frac{\sum w_{ij} y_j}{\sum w_{ij}}$$

Mean

Variance
Estimating filter error

- want to estimate the “mean square error”

\[
E \left[ (f(y_i) - x_i)^2 \right]
\]
Estimating filter error

- want to estimate the “mean square error”

\[ E \left[ (f(y_i) - x_i)^2 \right] \]

don’t know!
Estimating filter error

- want to estimate the “mean square error”

\[
E \left[ (f(y_i) - x_i)^2 \right] = E \left[ \text{SURE}(f(y_i)) \right]
\]

“Stein’s unbiased risk estimator”

*assuming Gaussian noise

quiz: why does the Gaussian noise assumption make sense?
Estimating filter error

- want to estimate the “mean square error”

\[ E \left[ (f(y_i) - x_i)^2 \right] = E \left[ \text{SURE} \left( f(y_i) \right) \right] \]

\[ \text{SURE} \left( f(y_i) \right) = (f(y_i) - y_i)^2 + 2\sigma_i^2 \frac{df(y_i)}{dy_i} - \sigma_i^2 \]

see Wikipedia for a proof [https://en.wikipedia.org/wiki/Stein%27s_unbiased_risk_estimate](https://en.wikipedia.org/wiki/Stein%27s_unbiased_risk_estimate)
Estimating filter error

- want to estimate the “mean square error”

\[ E \left[ (f(y_i) - x_i)^2 \right] = E \left[ \text{SURE} \left( f(y_i) \right) \right] \]

\[
\text{SURE} \left( f(y_i) \right) = (f(y_i) - y_i)^2 + 2\sigma_i^2 \frac{df(y_i)}{dy_i} - \sigma_i^2
\]

can be computed using finite difference or analytical derivatives of bilateral filters

see Wikipedia for a proof https://en.wikipedia.org/wiki/Stein%27s_unbiased_risk_estimate
Variance of the error estimator

SURE has its own variance!

\[ E \left[ (f(y_i) - x_i)^2 \right] = E \left[ \text{SURE} (f(y_i)) \right] \]
Variance of the error estimator

SURE has its own variance!

we used another fixed parameter cross bilateral filter to filter SURE

\[ E \left[ (f(y_i) - x_i)^2 \right] = E \left[ \text{SURE} (f(y_i)) \right] \]
Our method: for each pixel, try out different filter sizes, choose the one with smallest error.
Results & comparison

without denoising
(~82 samples per pixel)

ours
(~40 samples per pixel)

reference
(~4k samples per pixel)
Results & comparison

without denoising
(~82 samples per pixel)

ours
(~40 samples per pixel)

reference
(~4k samples per pixel)
SURE can be directly used for adaptive sampling

renderer → noisy image → denoised → estimate error

more samples in high (SURE) error regions
Two ways to choose the spatial support

2012

SURE-based Optimization for Adaptive Sampling and Reconstruction
Tzu-Mao Li
Yu-Ting Wu
Yung-Yu Chuang
National Taiwan University

used in movie production* e.g., Big Hero 6

2017

Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination
Christoph Schied
NVIDIA
Karlsruhe Institute of Technology
Anton Kaplanyan
Chris Wyman
Anjul Patney
NVIDIA

John Burgess
Shiqiu Liu
NVIDIA

Carsten Dachsacher
Karlsruhe Institute of Technology

Chakravarty R. Alla Chaitanya
NVIDIA
University of Montreal
McGill University

Aaron Lefohn
Marco Salvi
NVIDIA

used in video games e.g.,
Dying Light 2, Watch Dog Legion

*they used a version with non-local means filtering [Rousselle 2013]
Idea: use the variance directly to determine filter size

very noisy -> large filter
moderately noisy -> mid-size filter
not noisy -> small filter

quiz: when will this go wrong?
Scale the bilateral Gaussian variance using pixel variance

\[ G\left( \frac{\| C - C' \|^2}{\sigma_c^2} \right) \]

\[ \sigma_c^2 = \sigma_c^2 \sigma_i^2 + \epsilon \]
Using temporal information

very crucial for real-time rendering!

aka “temporal antialiasing”

https://www.lei-xu.com/post/taa/
Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination

Christoph Schied\textsuperscript{1,2}
Anjul Patney\textsuperscript{1}
Shiqiu Liu\textsuperscript{1}
Anton Kaplanyan\textsuperscript{1}
Chakravarty R. Alla Chaitanya\textsuperscript{1,3}
Carsten Dachsbacher\textsuperscript{2}
Marco Salvi\textsuperscript{1}
\textsuperscript{1}NVIDIA
\textsuperscript{2}Karlsruhe Institute of Technology
\textsuperscript{3}University of Montreal and McGill University

Chris Wyman\textsuperscript{1}
John Burgess\textsuperscript{1}
Aaron Lefohn\textsuperscript{1}
Neural networks

starting to take over the MC denoising world recently

color & variance

“G-buffer”

normal  albedo  depth

denoised image
Neural networks

starting to take over the MC denoising world recently

• in general:
  • at lower sample count, neural nets tend to produce higher quality results
  • at higher sample count, all methods tend to work similarly
  • neural nets are still somewhat slow for real-time rendering, and often produce temporal flickering
Neural networks

2017

Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings

STEVE BAKO*, University of California, Santa Barbara
THIJS VOGELS*, ETH Zürich & Disney Research
BRIAN MCWILLIAMS, Disney Research
MARK MEYER, Pixar Animation Studios
JAN NOVÁK, Disney Research
ALEX HARYILL, Pixar Animation Studios
PRADEEP SEN, University of California, Santa Barbara
TONY DEROSE, Pixar Animation Studios
FABRICE ROUSSELLE, Disney Research

Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder

CHAKRAVARTY R. ALLA CHAITANYA, NVIDIA, University of Montreal and McGill University
ANTON S. KAPLANYAN, NVIDIA
CHRISTOPH SCHIED, NVIDIA and Karlsruhe Institute of Technology
MARCO SALVI, NVIDIA
AARON LEFOHN, NVIDIA
DEREK NOWROUZEZAHRAI, McGill University
TIMO AILA, NVIDIA

(a) 1pp noisy input
(b) Edge-avoiding wavelets
(c) SURE-based filter
(d) Recurrent autoencoder
(e) Reference

offline

real-time
Sample-level denoising

we can also filter individual samples before aggregating them into pixels
Sample-level denoising pioneered by UCSD students!

Multidimensional Adaptive Sampling and Reconstruction for Ray Tracing

Toshiya Hachisuka*  Wojciech Jarosz*  Richard Peter Weistroffer†  Kevin Dale†
Greg Humphreys§  Matthias Zwicker*  Henrik Wann Jensen*

*UC San Diego  †Harvard University  ‡University of Virginia

2008

of course I also worked on this

Sample-based Monte Carlo Denoising using a Kernel-Splatting Network

MICHAËL GHARBI, Adobe and MIT CSAIL
TZU-MAO LI, MIT CSAIL
MIKA AIITALA, MIT CSAIL
JAAKKO LEHTINEN, Aalto University and NVIDIA
FÉDOR DURAND, MIT CSAIL

2019
Frequency analysis of light transport

analyze how light transport (reflection/shadow/etc) changes the bandwidth of the signal
Frequency analysis of light transport can be used for deriving filter kernels.

Axis-Aligned Filtering for Interactive Physically-Based Diffuse Indirect Lighting

Soham Uday Mehta\textsuperscript{1}  Brandon Wang\textsuperscript{1}  Ravi Ramamoorthi\textsuperscript{1}  Fredo Durand\textsuperscript{2}

\textsuperscript{1}University of California, Berkeley  \textsuperscript{2}MIT CSAIL

(a) 1-bounce indirect lighting, Our Method average 63 samples per pixel (spp)
(b) Adaptive Sampling and Filtering
(c) unfiltered 63 spp adaptively sampled
(d) unfiltered 63 spp uniformly sampled
(e) Our Method 63 adap. sample, filter
(f) Equal error 324 spp
Irradiance filtering vs irradiance caching

Irradiance Filtering for Monte Carlo Ray Tracing

Janne Kontkanen¹, Jussi Räsänen¹,², and Alexander Keller³
Denoising has become a necessary component of modern path tracing pipelines. Adaptive sampling is commonly used in offline rendering, but not quite so in real-time rendering.

NVIDIA Real-Time Denoiser Delivers Best-in-Class Denoising in Ubisoft’s Watch Dogs Legion

By Phillip Singh

Tags: featured, Game Development, News, ray tracing
Next: variance reduction