

Aktuelle Techniken für das Entrauschen von Realtime-Raytracing

Matz Radloff
08.02.2021

Inhalt

- Einleitung
- Raytracing Hintergrund
- Methodenvergleich
- Ausgewählte Beispiele
 - SVGF
 - Denoising Autoencoder
- Limitationen und Zusammenfassung

Einleitung

- Raytracing bzw. Path Tracing lösen Rendergleichung
- Rendergleichung berechnet exakt Intensität, Ort und Richtung von Licht
- Fotorealistische Bilder
- Modelliert natürliches Verhalten von Licht

Einleitung



Abb. 1: "JIM", Quelle: Sefki Ibrahim, Artstation.com [1]

Einleitung - Nachteile von Raytracing

- Sehr hoher Rechenaufwand (oft mehrere Minuten oder Stunden pro Bild)
- In der Praxis bisher nur für "offline"-Anwendungen nutzbar
- CGI in Filmen / Serien
- Architektur-Visualisierung

Raytracing Hintergrund

- Raytracing ist der Oberbegriff für verschiedene ähnliche Techniken
- Virtuelle Strahlen und deren Schnittpunkte mit Objekten
- Vergleichbar mit natürlichem Licht
- Interaktion mit Materie legt fest, wie viele Photonen sich in welche Richtung bewegen
- Strahlen starten meist von der Kamera aus, statt von den Lichtquellen

Raytracing Hintergrund

- Erweiterungen:
 - Lichtbrechung
 - Lichtbeugung
 - Schatten
 - Transmission
- Path Tracing umfasst alle diese Erweiterungen

Raytracing Hintergrund - Path Tracing

- Viele verschiedene Objekte und Lichtquellen können zur finalen Farbe eines Pixels beitragen
- → Mehrere Abtastpunkte pro Pixel notwendig
- Bei zufälliger Positionierung: Monte-Carlo-Path-Tracing
- Wenige Abtastpunkte als Rauschen sichtbar

Raytracing Hintergrund - Path Tracing



Abb. 2:
Blender "Classroom"
Demo Scene
Cycles Path Tracer
5 Samples per Pixel

Raytracing Hintergrund - Path Tracing



Abb. 3:
Blender "Classroom"
Demo Scene
Cycles Path Tracer
300 Samples per Pixel

Raytracing Hintergrund - Path Tracing

- Path Tracing Performance-Optimierungen nicht ausreichend für Berechnung in Echtzeit
- Ansatz: Denoising / Entrauschen
 - Bild mit wenig Abtastpunkten generieren
 - Im Nachhinein Rauschen entfernen

Methodenvergleich

- Filter
- Sampling
- Machine Learning
- Upscaling
- Kombinationen

Methodenvergleich - Filter

- Bekannte Weichzeichnungsmethoden, deren Parameter dynamisch angepasst werden
- Statt jeden Pixel gleich zu behandeln, werden zusätzliche Informationen aus der Rendering-Pipeline verwendet

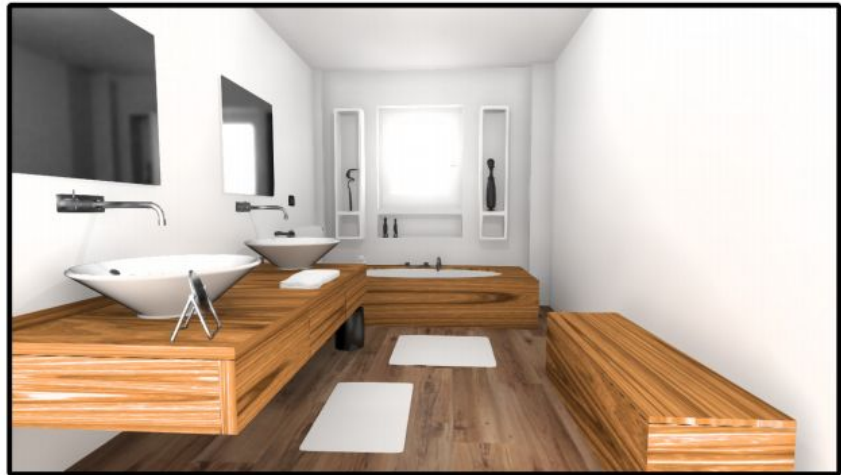


Abb. 4: Mara et al.: *An Efficient Denoising Algorithm for Global Illumination* [2]

Methodenvergleich - Filter

- Schnell zu berechnen
- Verringerte Bildqualität durch unerwünschtes Filtern
- Hochfrequente Informationen gehen oft verloren
- Helligkeit und Farben werden verfälscht

Methodenvergleich - Sampling

- Path Tracing ist normalerweise “unbiased”
- → Jeder Abtastpunkt ist gleichwertig
- Verschiedene Pixel benötigen unterschiedlich viel Zeit, um rauschfrei generiert zu werden
- Verschiedene Abtastpunkte tragen unterschiedlich stark zum Ergebnis bei
- Manche Pixel ändern sich in Bilderfolgen kaum
- → Intelligente Auswahl optimiert Berechnung

Methodenvergleich - Machine Learning

- Allgemein gute Ergebnisse beim Entrauschen von Bildern
- Zusätzliche Eingabeinformationen
 - Vorheriges Bild
 - Wichtigkeit der Abtastpunkte
 - Normalen- und Bewegungsvektoren
 - Tiefen- und Geometrie-Informationen

Methodenvergleich - Upscaling

- Generell unabhängig von der Rendermethode
- Überführt ein niedrig auflösendes Bild in ein hochauflösendes
- Je niedriger die Auflösung, desto mehr Abtastpunkte können verwendet werden
- Vor allem mit anderen Methoden zusammen sehr leistungsstark

Methodenvergleich - Kombinationen

- Gleichzeitiges Ausnutzen der Vorteile verschiedener Techniken
- Beispiele:
 - Upscaling mithilfe von Machine Learning
 - Bestimmung der Sampling- und Filter-Parameter mit einem neuronalen Netzwerk
 - Ausgaben der Rendering-Pipeline als Eingaben für ML und adaptives Sampling/Filtern
- Berechnung in Echtzeit mit aktueller Hardware nur mit diesen Kombinationen möglich

SVGF

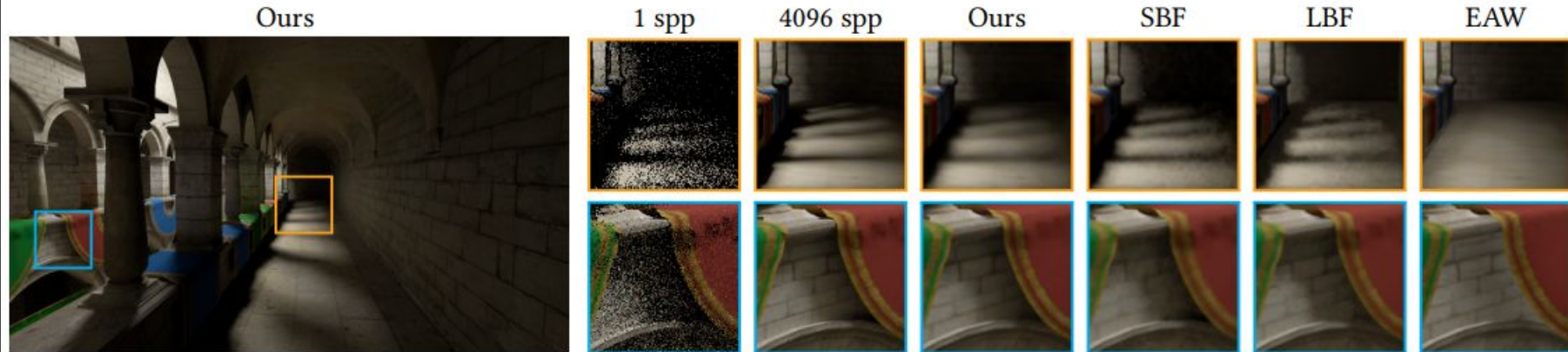


Abb. 5: Schied et al.: *Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination* [3]

SVGF

- Spatio-Temporal Variance Guided Filtering
- Eingabebild verwendet nur einen Abtastpunkt pro Pixel
- Zeitlich stabil → Für Animationen geeignet
- Berechnung eines Bilds in 1080p benötigt 10ms ($\pm 15\%$)

SVGF

- Iterative Wavelet-Filter
 - “Equalizer” für Bilder
 - Hervorheben oder Reduzieren von Details pro Frequenzbereich in 2D
- Temporaler Filter, vergleichbar mit Temporal Anti Aliasing (TAA)
- Parameter aus der Rendering-Pipeline
 - Bewegungs- und Normalenvektoren
 - Farbe
 - Tiefen- und Geometrieinformationen

SVGF

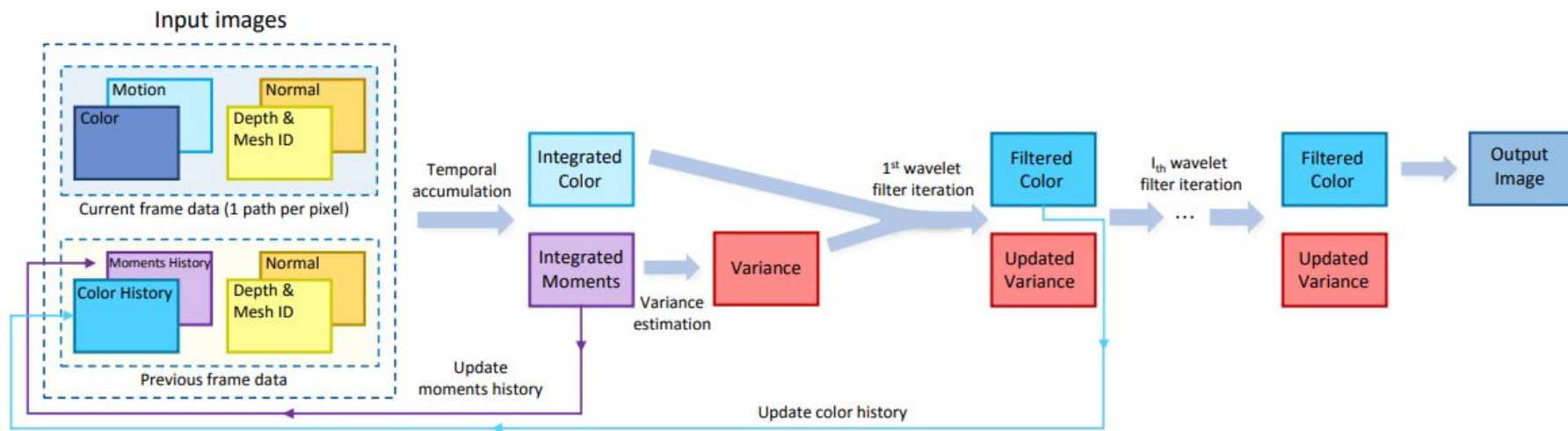


Abb. 6: Schied et al.: *Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination* [3]

SVGF - Temporaler Filter

- 2D-Bewegungsvektors pro Pixel
- Projektion auf vorheriges Bild (Farbe, Tiefe, Normalen, Mesh-ID)
- Je nach Übereinstimmung werden Farbinformationen übernommen
- Keine Übereinstimmung → Verdeckung

SVGF - Varianz der Leuchtkraft

- Farbänderung mehrerer zeitlich nebeneinander liegenden Pixel
- Guter Indikator für Rauschen
- Nicht perfekt, aber gute Basis
- Rauschen erhöht Varianz
- Varianz kann auch ohne Rauschen einen hohen Wert besitzen

SVGF - Wavelet Filter

- 5 Iterationen über verschiedene Größen / Frequenzen
- Ausgabe der ersten Iteration wird als Farbe in der Historie gespeichert
- Varianz legt die Stärke der Filterung fest
- Rasterisierter Tiefenpuffer wird für Kantenerkennung benutzt (rauschfrei)

SVGF

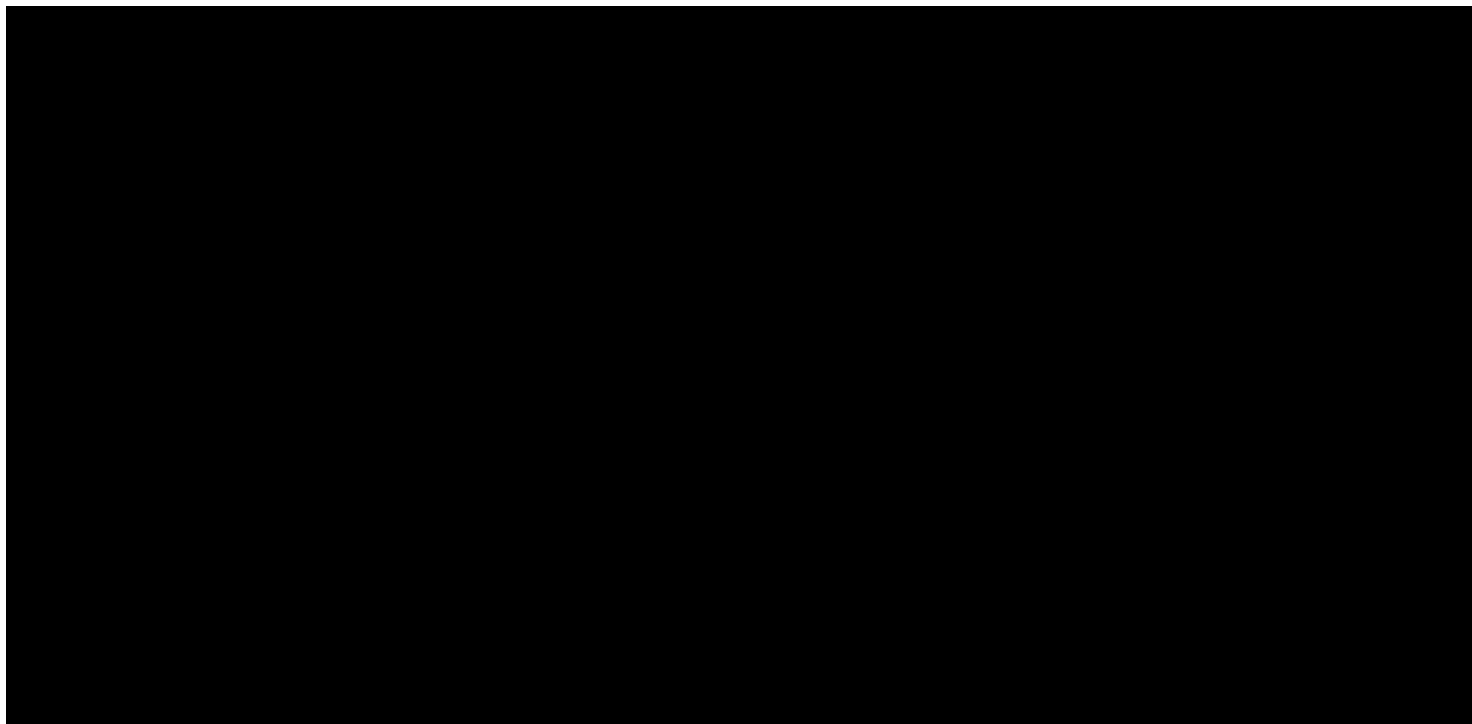


Abb. 7: SVGF Video Demo, Nvidia, Quelle:

https://research.nvidia.com/publication/2017-07_Spatiotemporal-Variance-Guided-Filtering%3A

Denoising Autoencoder



Abb. 8: Chaitanya et al.: *Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder* [4]

Denoising Autoencoder

- Ein Abtastpunkt pro Pixel
- Benötigt 54.9ms um ein Bild in 720p zu berechnen
- Recurrent Convolutional Neural Network (RCNN) basierter Autoencoder

Denoising Autoencoder - RCNN

- Recurrent Neural Networks besitzen Rückkopplungen zwischen verschiedenen Berechnungsschritten (hier: zwischen mehreren Bildern in einer Folge)
- Convolutional Neural Networks verwenden Faltungsmatrizen, um wichtige Informationen zu integrieren bzw. zu extrahieren (Encoder, Decoder)
- Encoder + Decoder = Autoencoder
- Allgemein gut für Bildtransformationen geeignet

Denoising Autoencoder - RCNN

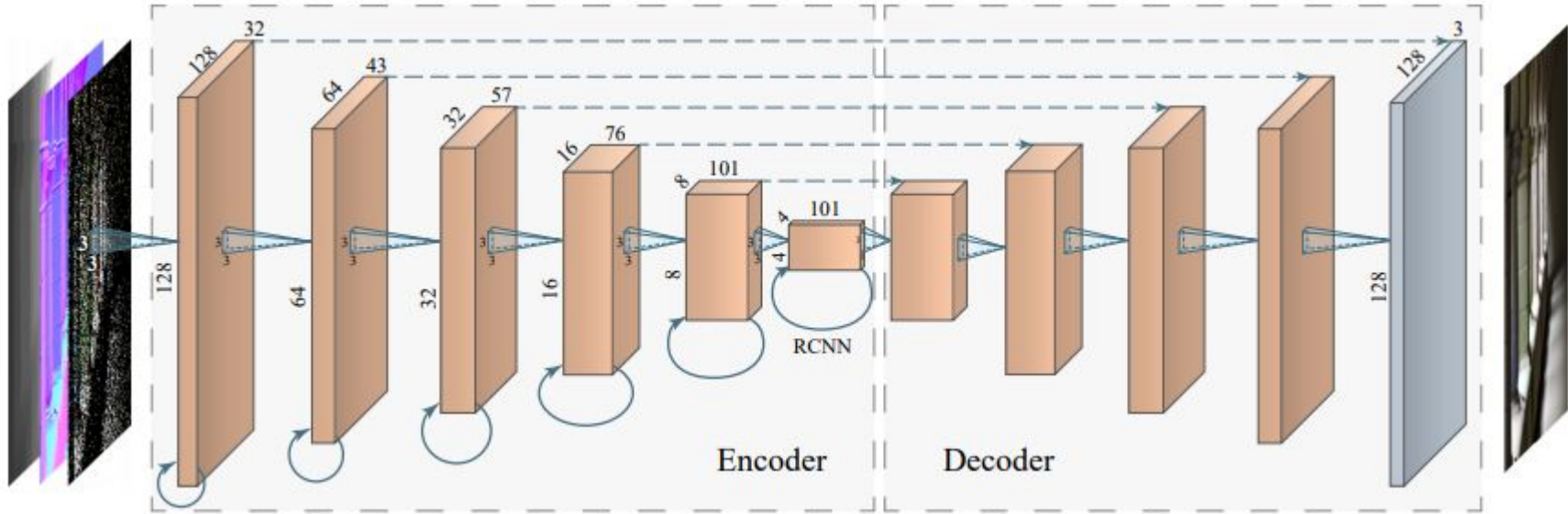


Abb. 9: Chaitanya et al.: *Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder* [4]

Denoising Autoencoder

- Verbesserung der Leistung durch weitere Informationen als Eingabe
 - Normalenvektoren (2D, view-space)
 - Tiefeninformationen
 - Materialrauhigkeit
- Insgesamt 3 Farbwerte + 4 Zusatzwerte → 7 Eingabewerte pro Pixel
- Sehr rauscharmes Bild bei geeigneter Wahl der Parameter
- Mit 54.9ms pro Bild ist die Performance nicht für Echtzeitanwendungen geeignet aber nicht weit entfernt

Limitationen und Zusammenfassung

- Path Tracing qualitativ bestmögliche Methode für die Generierung von fotorealistischen Bildern
- Geringe Anzahl von Abtastpunkten und anschließendes Entrauschen stellt einen praktikablen Ansatz dar
 - → Höher Qualität bei gleicher Performance
 - → Bessere Performance bei vergleichbarer Qualität

Limitationen und Zusammenfassung

- Größte Limitation ist weiterhin die benötigte Rechenzeit
- Auch mit aktueller Hardware müssen viele Kompromisse eingegangen werden
- Szenenkomplexität <> Bildqualität <> Performance
- Aktuelle Beispiele in der Praxis:

Limitationen und Zusammenfassung

- Quake II RTX (Adaptive-SVGF), Minecraft RTX (SVGF-Variante), Teardown
 - Verwenden echtes Path Tracing
 - Stark vereinfachte Welten
 - Meist nur in Kombination mit Upscaling flüssig spielbar
- Battlefield V, Control, Call of Duty Modern Warfare
 - Einzelne Aspekte der Rasterisierungs-Pipeline werden mithilfe von Raytracing berechnet
 - Schatten, Globale Beleuchtung, Reflektionen
 - Meist werden nur einzelne Methoden benutzt

Limitationen und Zusammenfassung

- Benötigter Anstieg der Rechenleistung ca 2-4x
- → Bei ca 20% mehr Leistung pro Hardware-Generation 4-8 Jahre
- Wahrscheinlich weniger durch Raytracing-spezifische Hardware

Literatur und Quellen

- [1] Sefki Ibrahim: “JIM”

<https://www.artstation.com/artwork/A93VGm>

Letzter Zugriff: 04.02.2021

- [2] Michael Mara, McGuire, Morgan and Bitterli, Benedikt and Jarosz, Wojciech:

An Efficient Denoising Algorithm for Global Illumination

<https://cs.dartmouth.edu/~wjarosz/publications/mara17towards.pdf>

Juli 2017

Letzter Zugriff: 16.01.2021

Literatur und Quellen

- [3] Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn und Marco Salvi:
Spatiotemporal Variance-Guided Filtering: Real-Time
Reconstruction for Path-Traced Global Illumination
https://cg.ivd.kit.edu/publications/2017/svgf/svgf_preprint.pdf
Juli 2017
Letzter Zugriff: 17.01.2021

Literatur und Quellen

- [4] Chakravarty R. Alla Chaitanya, Anton Kaplanyan, Christoph Schied, Marco Salvi,
Aaron Lefohn, Derek Nowrouzezahrai, Timo Aila:
Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder
https://research.nvidia.com/sites/default/files/publications/dnn_denoise_author.pdf
Juli 2017
Letzter Zugriff: 17.01.2021