facebook

Reality Labs

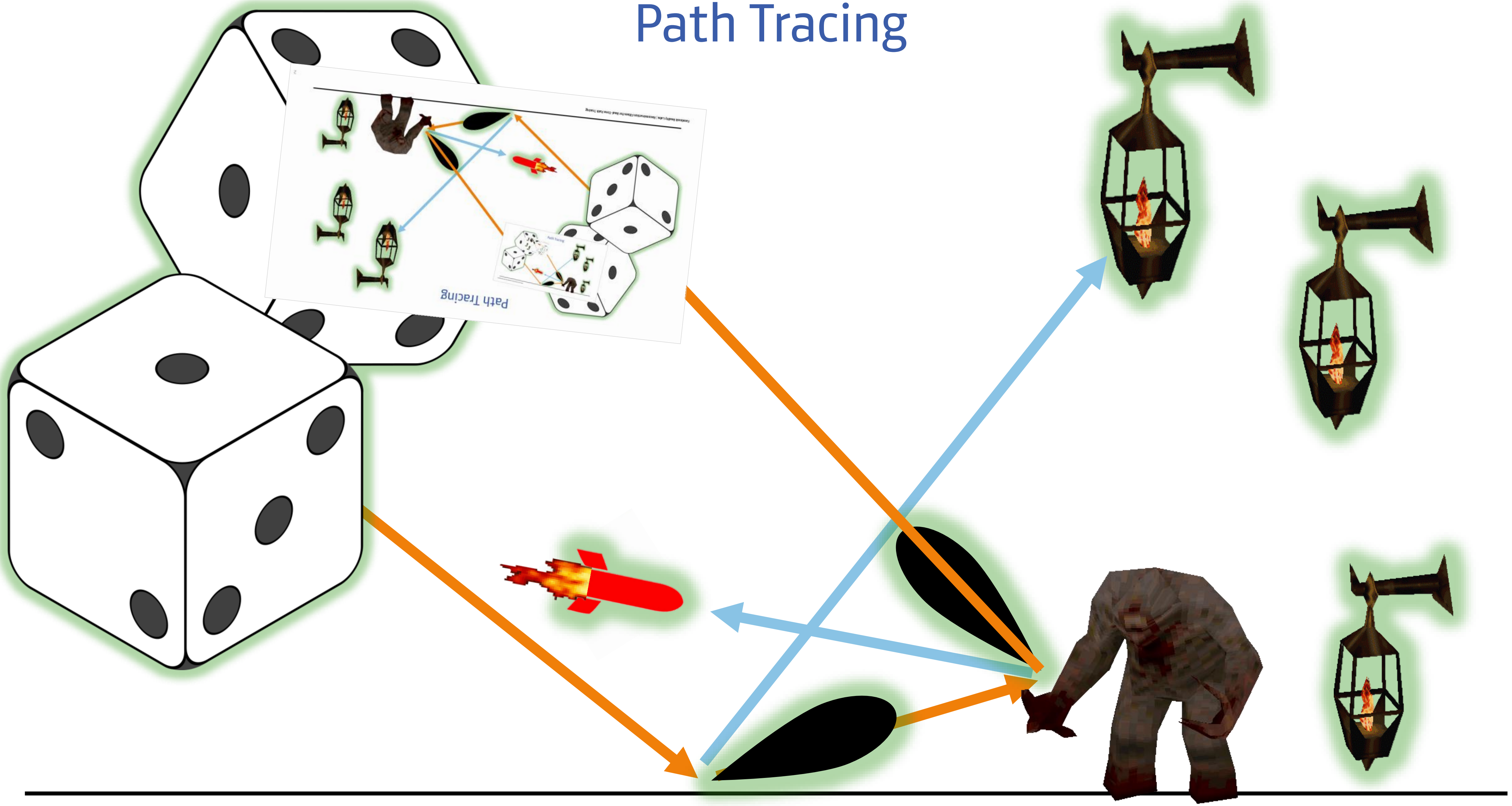
Graphics Research Team

Reconstruction Filters for Real-Time Path Tracing

Christoph Schied

July 28, 2019

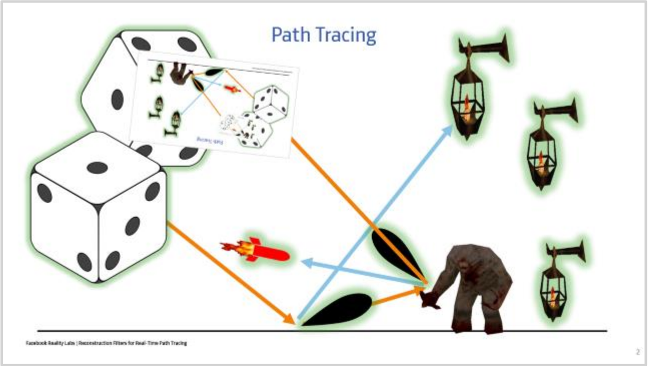
Path Tracing



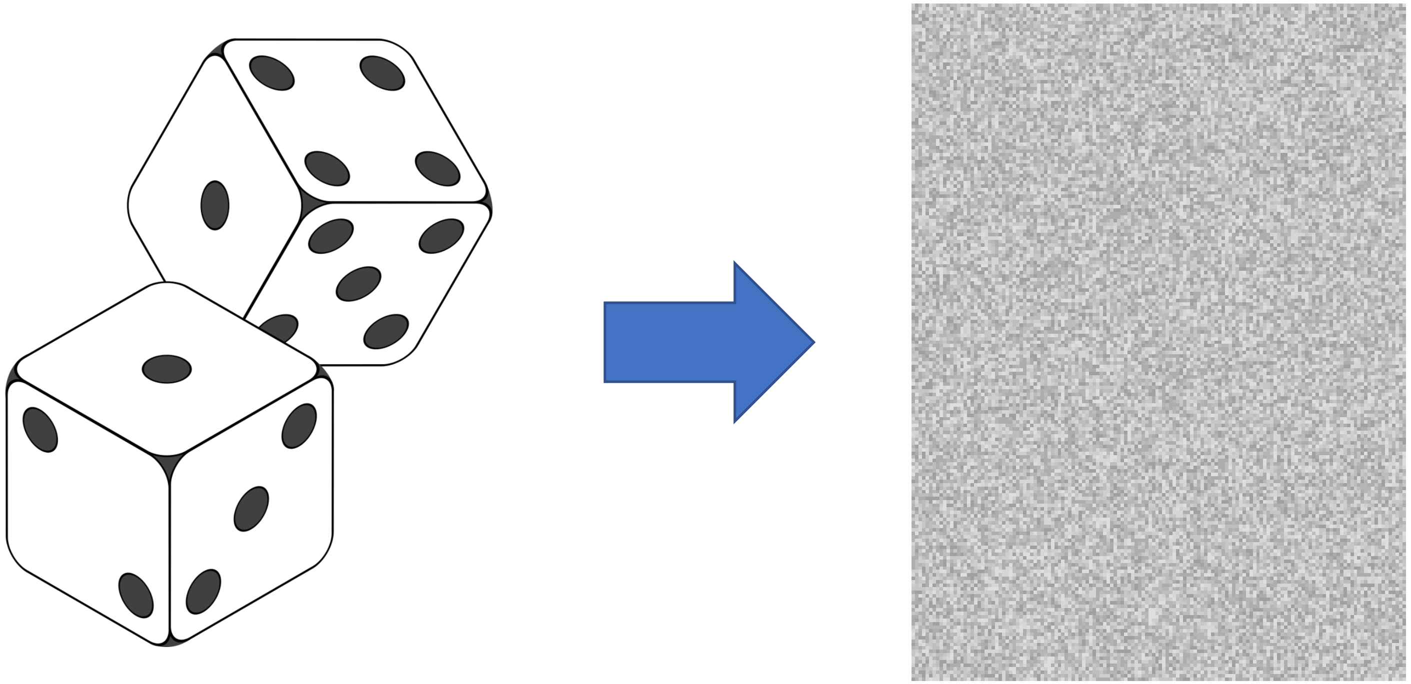
Path Tracing

Happy if we can actually even afford n=1

$$L \approx \frac{1}{n}\sum_{i=1}^{n}$$

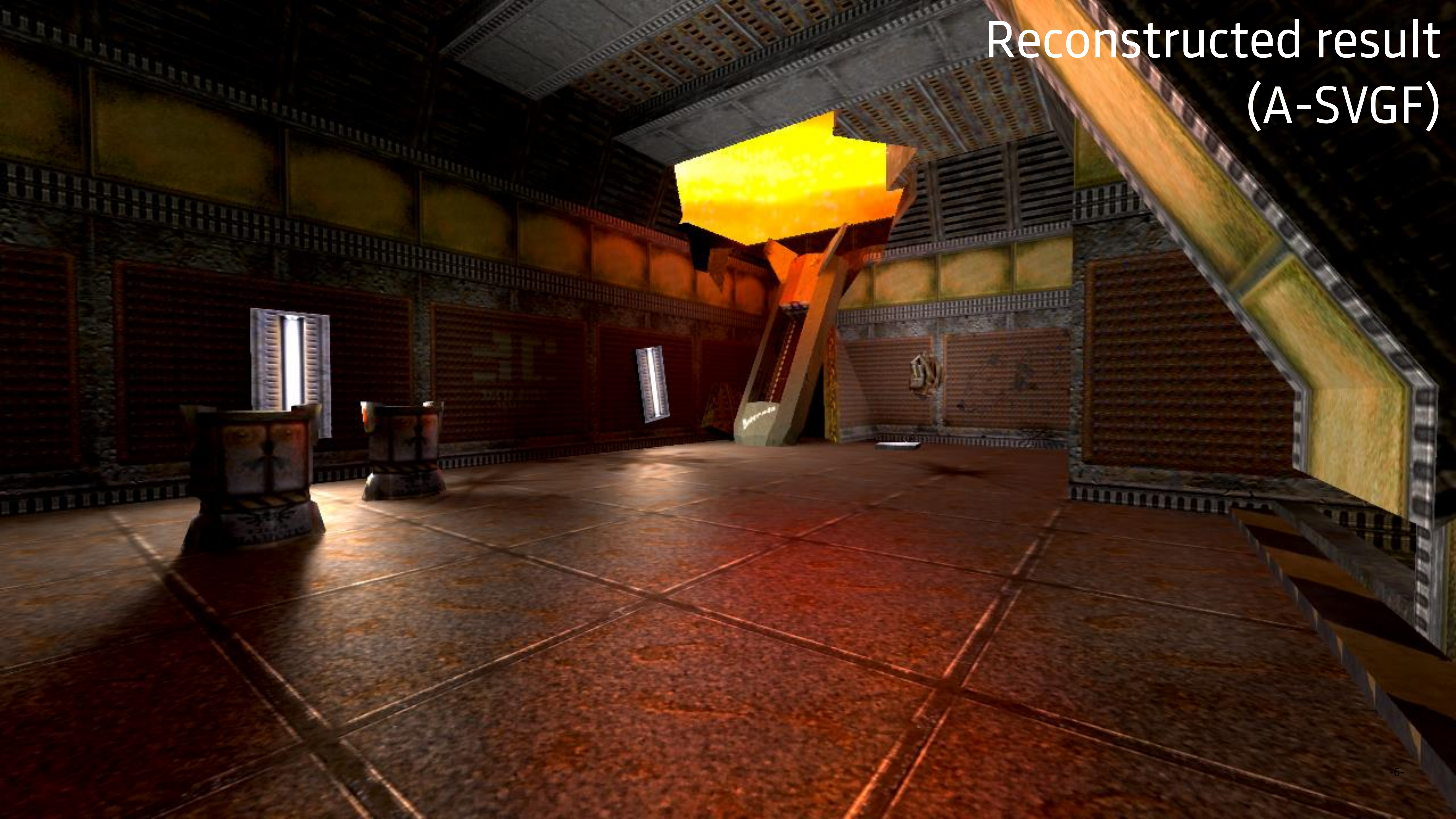


Main challenges



- Better sampling → less noise
- Reconstruction filters (Denoising)

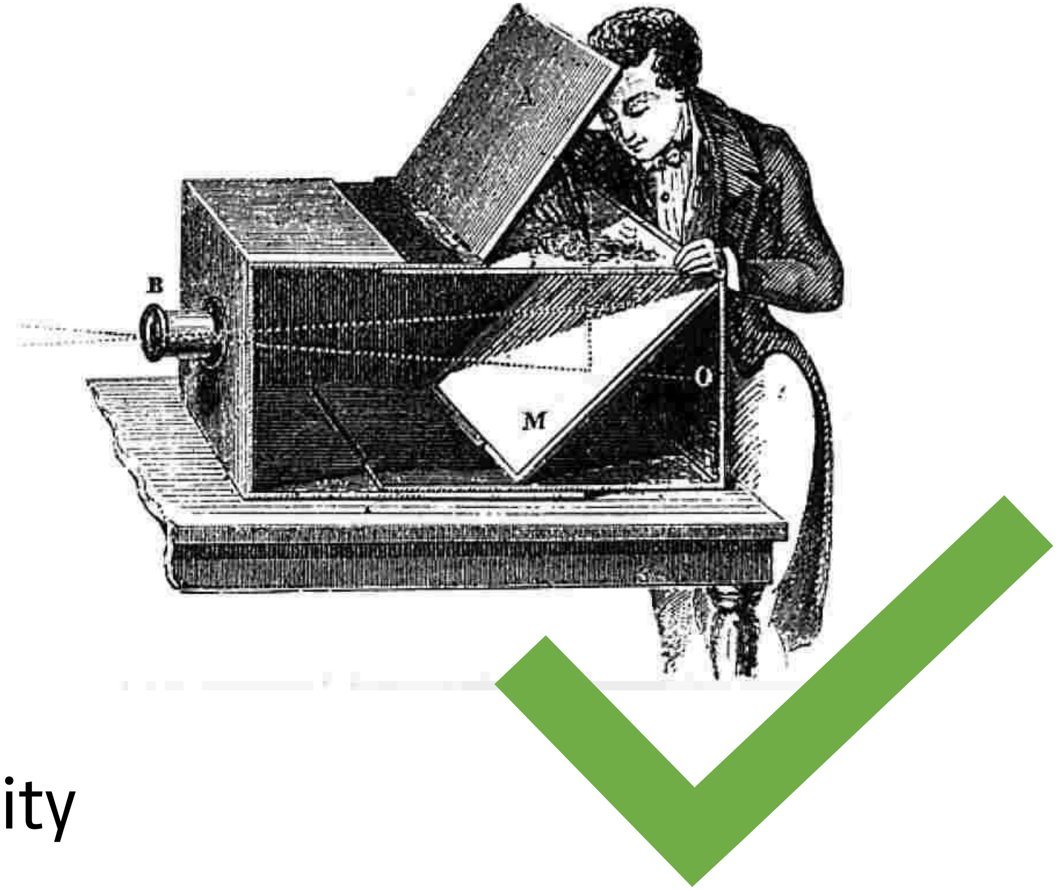

Path tracer output
(1spp)



Reconstructed result (A-SVGF)

Stochastic Ray Tracing vs. Path Tracing

- Current games: stochastic ray tracing
  - Stochastically sample individual effects (e.g. BRDF or area light)
  - Direct illumination at bounces is usually noise-free! (no need to denoise mirrors)
- Path tracing stochastically samples everything
  - Light selection
  - Area lights
  - Indirect Bounces
- In this talk: general reconstruction filters for full path tracing in real-time

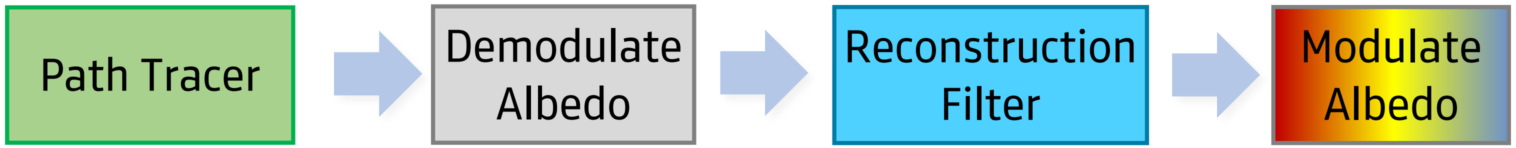


No stochastic sampling of primary visibility

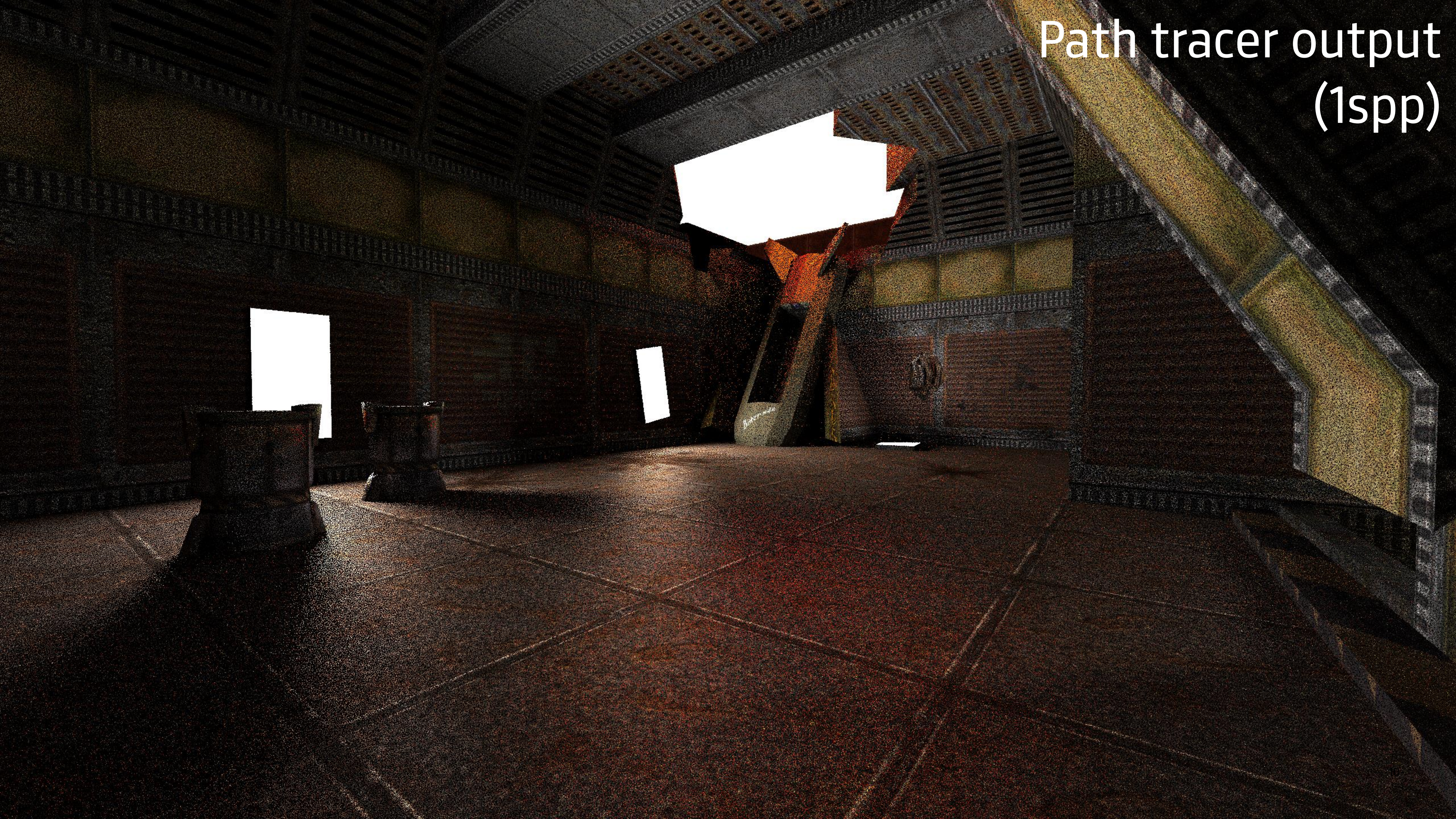
→ Nearby surfaces close in screen-space

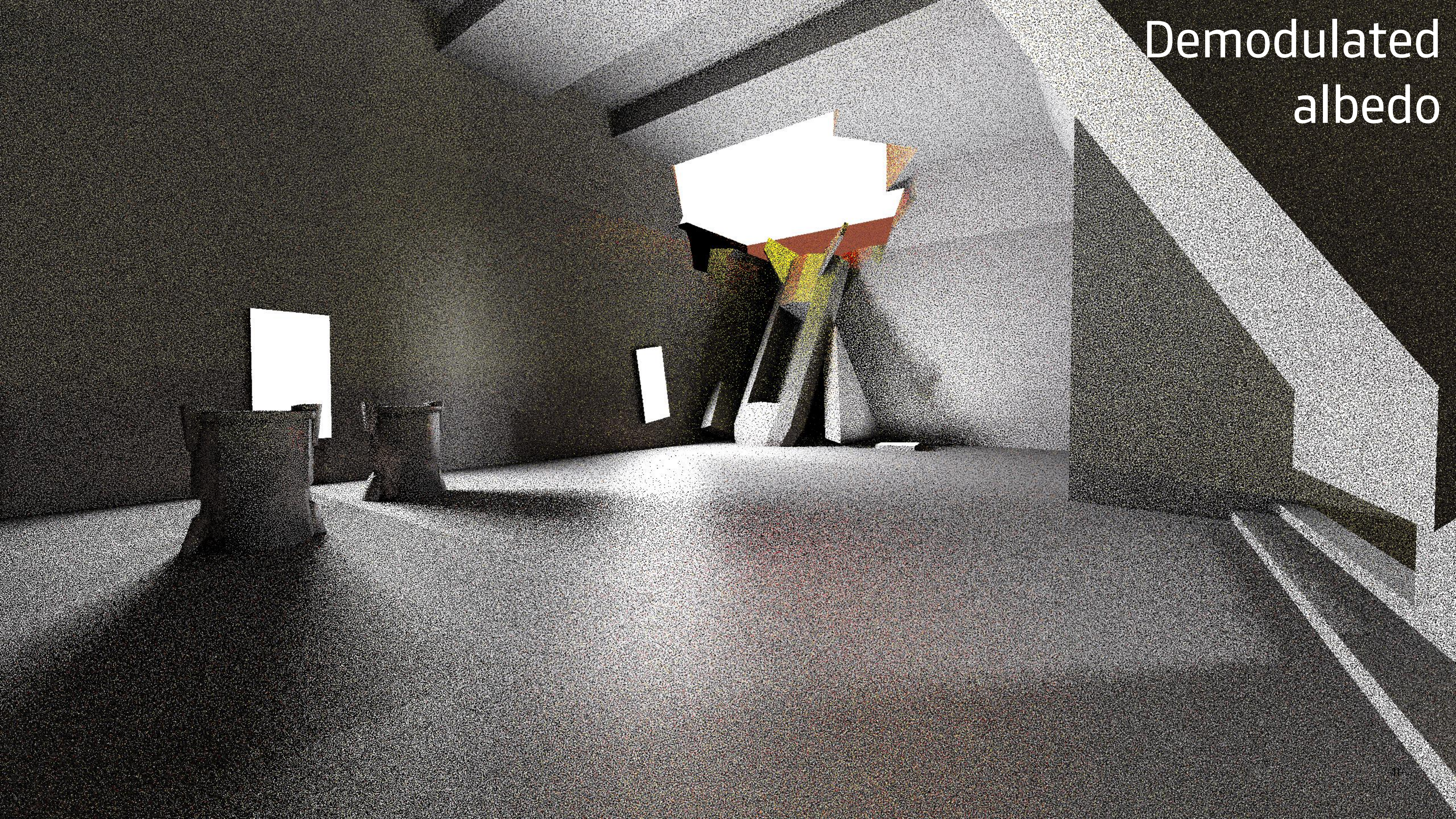
→ G-Buffer noise-free

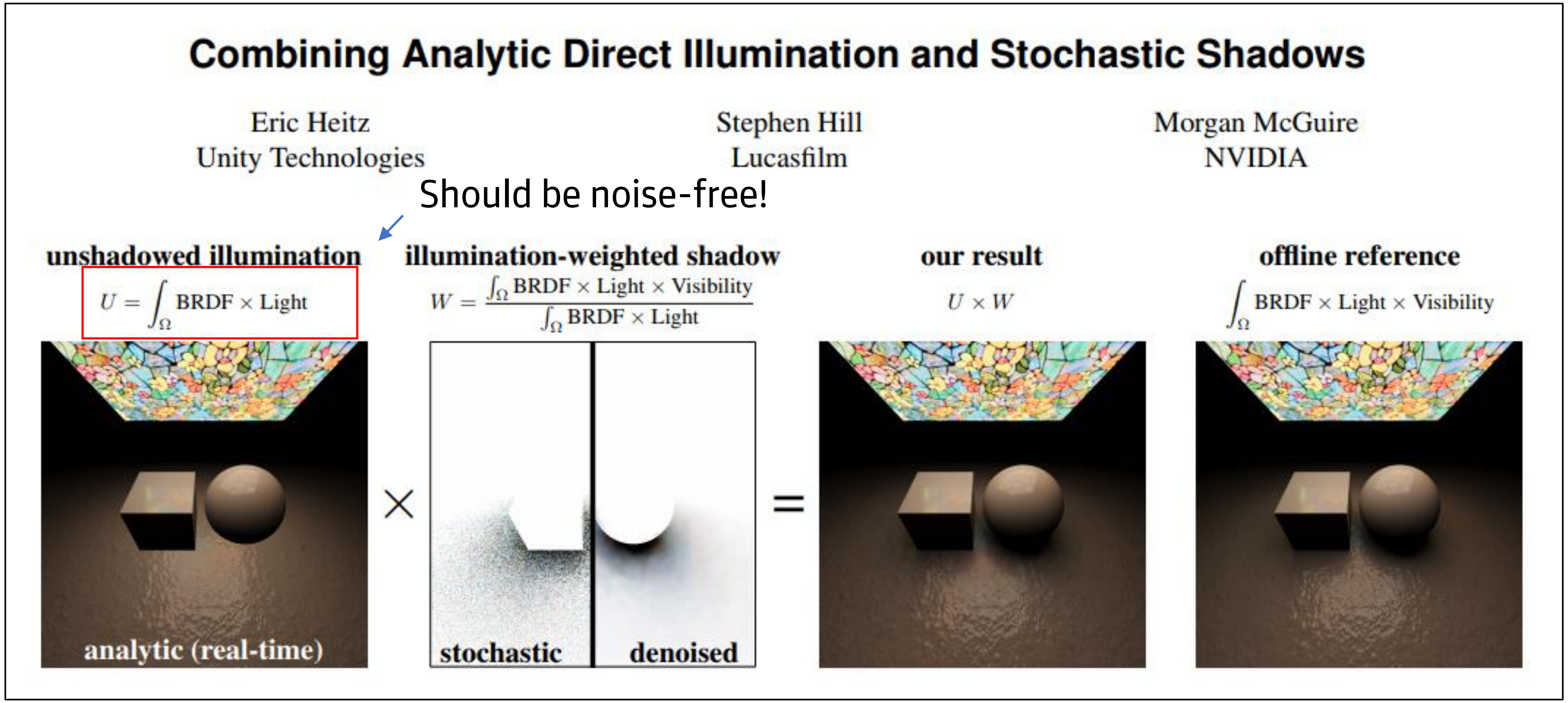
Albedo demodulation



- Essentially used by all of the reconstruction filters
- Much easier to filter “untextured” illumination
- Problems:
  - Fresnel term
  - Layered materials


Path tracer output
(1spp)

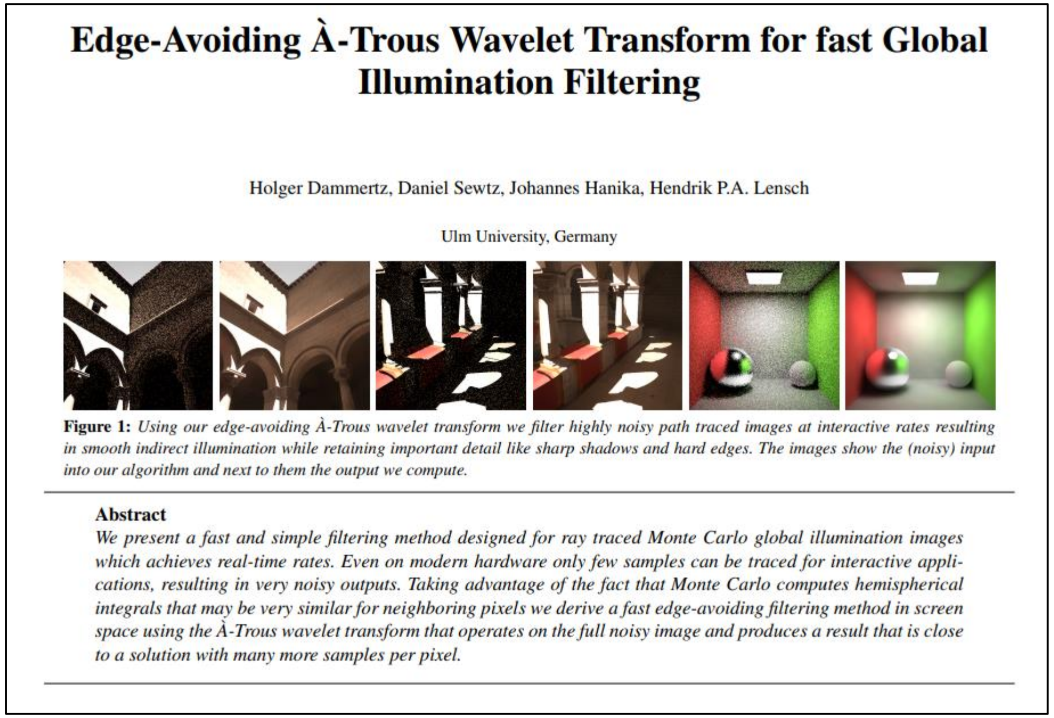

Demodulated
albedo



- Can be thought of as better albedo demodulation

- Cannot trivially be used with stochastic light selection

Wavelet-based reconstruction filters



Edge-Avoiding À-Trous Wavelet Transform for fast Global Illumination Filtering

Holger Dammertz, Daniel Sewtz, Johannes Hanika, Hendrik P.A. Lensch

Ulm University, Germany

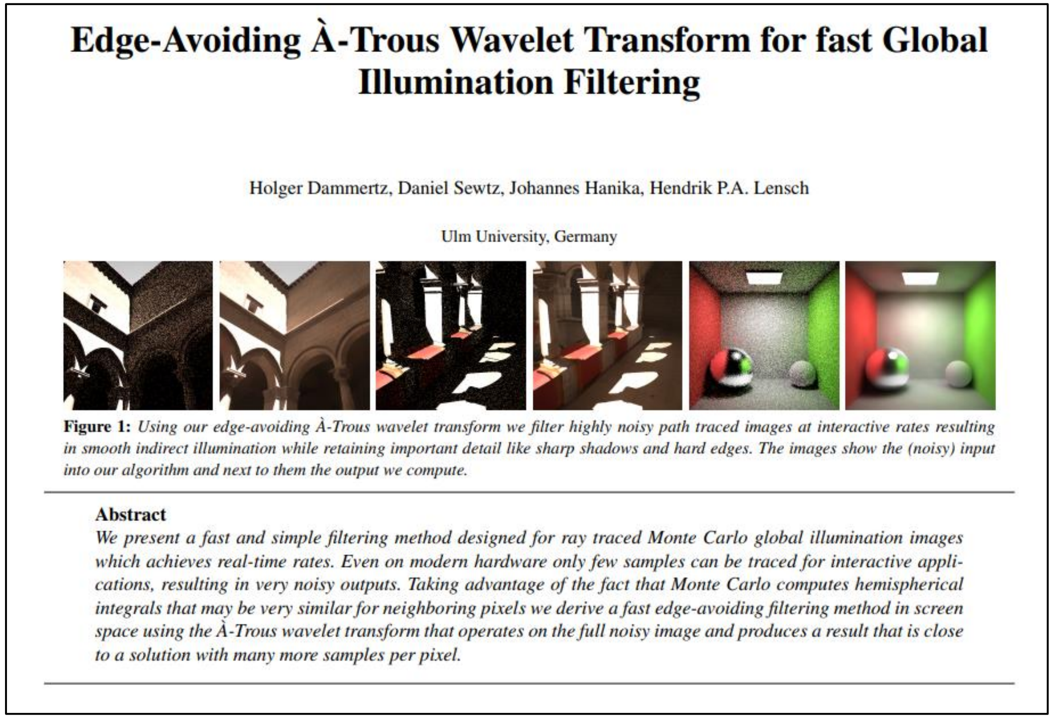
Figure 1: *Using our edge-avoiding À-Trous wavelet transform we filter highly noisy path traced images at interactive rates resulting in smooth indirect illumination while retaining important detail like sharp shadows and hard edges. The images show the (noisy) input into our algorithm and next to them the output we compute.*

Abstract

We present a fast and simple filtering method designed for ray traced Monte Carlo global illumination images which achieves real-time rates. Even on modern hardware only few samples can be traced for interactive applications, resulting in very noisy outputs. Taking advantage of the fact that Monte Carlo computes hemispherical integrals that may be very similar for neighboring pixels we derive a fast edge-avoiding filtering method in screen space using the À-Trous wavelet transform that operates on the full noisy image and produces a result that is close to a solution with many more samples per pixel.

- Fast
- Some artifacts
- Requires parameter tweaking

$\hat{c}_i(q)$



Edge-Avoiding À-Trous Wavelet Transform for fast Global Illumination Filtering

Holger Dammertz, Daniel Sewtz, Johannes Hanika, Hendrik P.A. Lensch

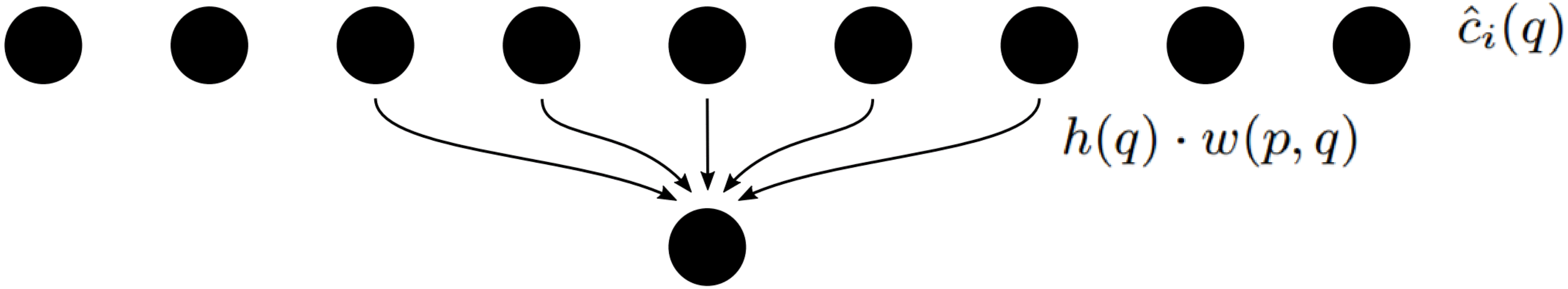
Ulm University, Germany

Figure 1: *Using our edge-avoiding À-Trous wavelet transform we filter highly noisy path traced images at interactive rates resulting in smooth indirect illumination while retaining important detail like sharp shadows and hard edges. The images show the (noisy) input into our algorithm and next to them the output we compute.*

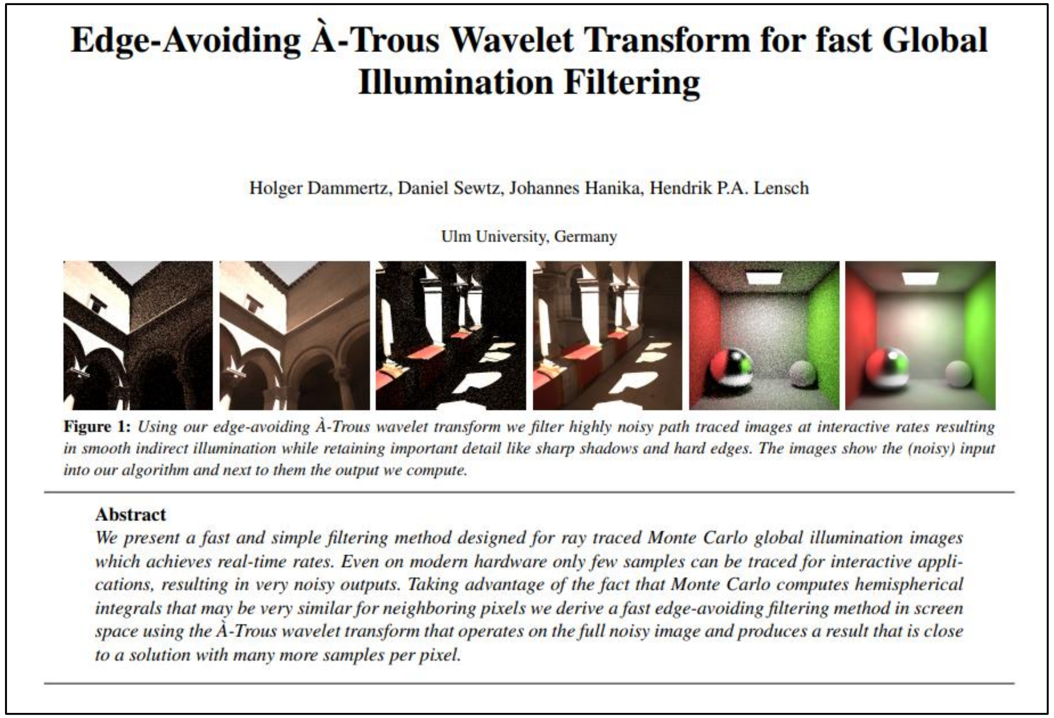
Abstract

We present a fast and simple filtering method designed for ray traced Monte Carlo global illumination images which achieves real-time rates. Even on modern hardware only few samples can be traced for interactive applications, resulting in very noisy outputs. Taking advantage of the fact that Monte Carlo computes hemispherical integrals that may be very similar for neighboring pixels we derive a fast edge-avoiding filtering method in screen space using the À-Trous wavelet transform that operates on the full noisy image and produces a result that is close to a solution with many more samples per pixel.

- Fast
- Some artifacts
- Requires parameter tweaking



$$\hat{c}_{i+1}(p) = \frac{\sum_{q \in \Omega} h(q) \cdot w(p,q) \cdot \hat{c}_i(q)}{\sum_{q \in \Omega} h(q) \cdot w(p,q)}$$



Edge-Avoiding À-Trous Wavelet Transform for fast Global Illumination Filtering

Holger Dammertz, Daniel Sewtz, Johannes Hanika, Hendrik P.A. Lensch

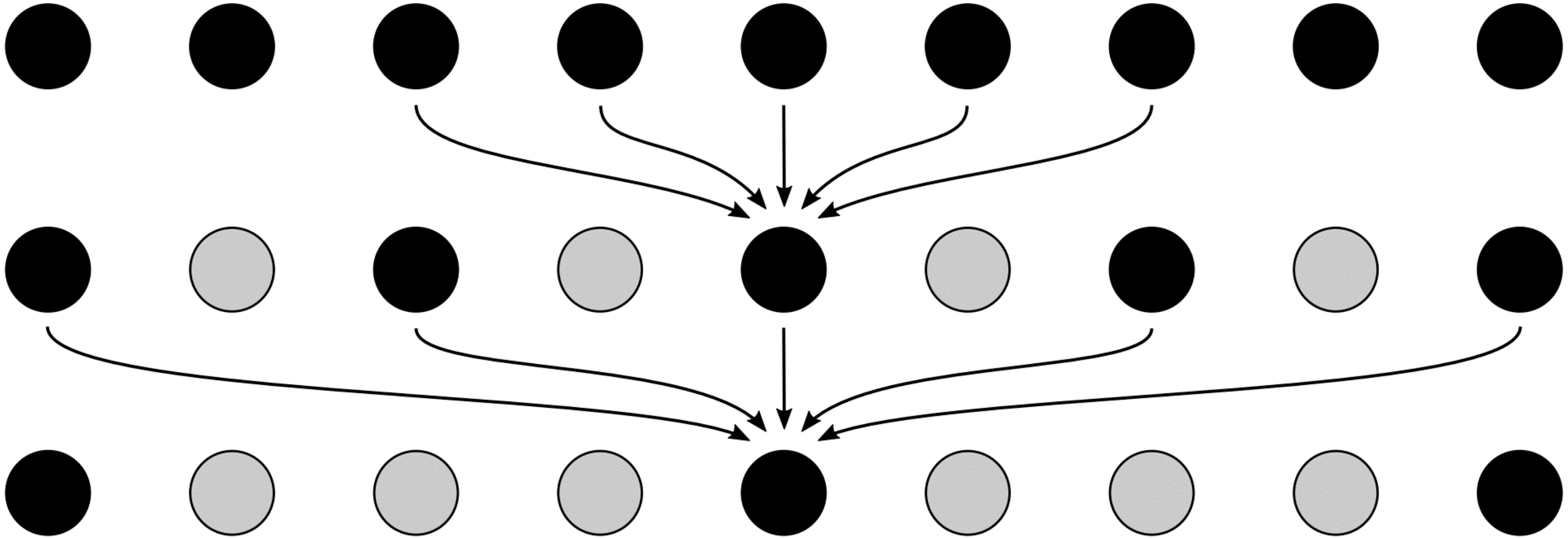
Ulm University, Germany

Figure 1: *Using our edge-avoiding À-Trous wavelet transform we filter highly noisy path traced images at interactive rates resulting in smooth indirect illumination while retaining important detail like sharp shadows and hard edges. The images show the (noisy) input into our algorithm and next to them the output we compute.*

Abstract

We present a fast and simple filtering method designed for ray traced Monte Carlo global illumination images which achieves real-time rates. Even on modern hardware only few samples can be traced for interactive applications, resulting in very noisy outputs. Taking advantage of the fact that Monte Carlo computes hemispherical integrals that may be very similar for neighboring pixels we derive a fast edge-avoiding filtering method in screen space using the À-Trous wavelet transform that operates on the full noisy image and produces a result that is close to a solution with many more samples per pixel.

- Fast
- Some artifacts
- Requires parameter tweaking



Edge-stopping functions

$$\hat{c}_{i+1}(p) = \frac{\sum_{q\in\Omega} h(q) \cdot \boxed{w(p,q)} \cdot \hat{c}_i(q)}{\sum_{q\in\Omega} h(q) \cdot w(p,q)}$$

Input | No edge-stopping | RT-Buffer | + Normal | + Position

[Dammertz et al. 2010]

Builds on Edge-avoiding À-trous Wavelets

- Improved geometric edge-stopping functions
    - Adapt automatically
- Hierarchical noise estimation
- Integrates temporal filtering

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

Figure 1: Our filter takes (left) 1 sample per pixel path-traced input and (center) reconstructs a temporally stable 1920×1080 image in just 10 ms. Compare to (right) a 2048 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.

ABSTRACT

We introduce a reconstruction algorithm that generates a temporally stable sequence of images from one path-per-pixel global illumination. To handle such noisy input, we use temporal accumulation to increase the effective sample count and spatiotemporal luminance variance estimates to drive a hierarchical, image-space wavelet filter [Dammertz et al. 2010]. This hierarchy allows us to distinguish between noise and detail at multiple scales using local luminance variance.

Physically based light transport is a long-standing goal for real-time computer graphics. While modern games use limited forms of ray tracing, physically based Monte Carlo global illumination does not meet their 30 Hz minimal performance requirement. Looking ahead to fully dynamic real-time path tracing, we expect this to only be feasible using a small number of paths per pixel. As such, image reconstruction using low sample counts is key to bringing path tracing to real-time. When compared to prior interactive reconstruction filters, our work gives approximately 10× more temporally stable results, matches reference images 5–47% better (according to SSIM), and runs in just 10 ms (± 15%) on modern graphics hardware at 1920×1080 resolution.

HPG '17, Los Angeles, CA, USA

CCS CONCEPTS

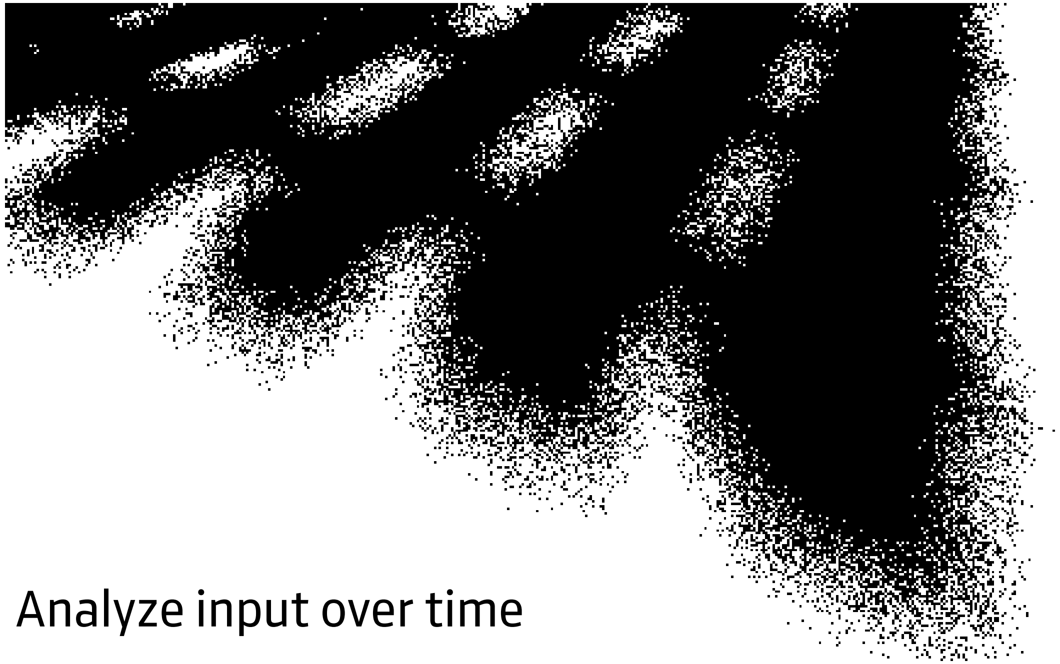
•**Computing methodologies** →Ray tracing;

KEYWORDS

global illumination, reconstruction, real-time rendering

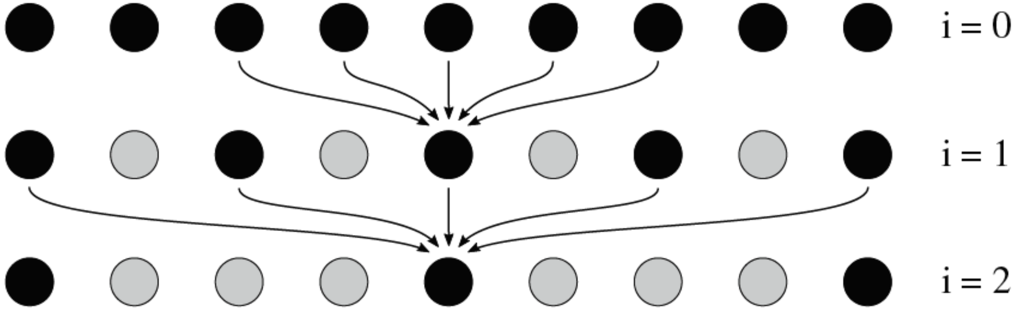
ACM Reference format:
Christoph Schied, Anton Kaplanyan, Chris Wyman, Anjul Patney, Chakravarty R. Alla Chaitanya, John Burgess, Shiqiu Liu, Carsten Dachsbacher, Aaron Lefohn, and Marco Salvi. 2017. Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination. In *Proceedings of HPG '17, Los Angeles, CA, USA, July 28-30, 2017*, 12 pages.
DOI: 10.1145/3105762.3105770

Spatiotemporal Variance-Guided Filter (SVGF)



Analyze input over time

- Temporally unstable ⟶ blur more
- Temporally stable ⟶ blur less

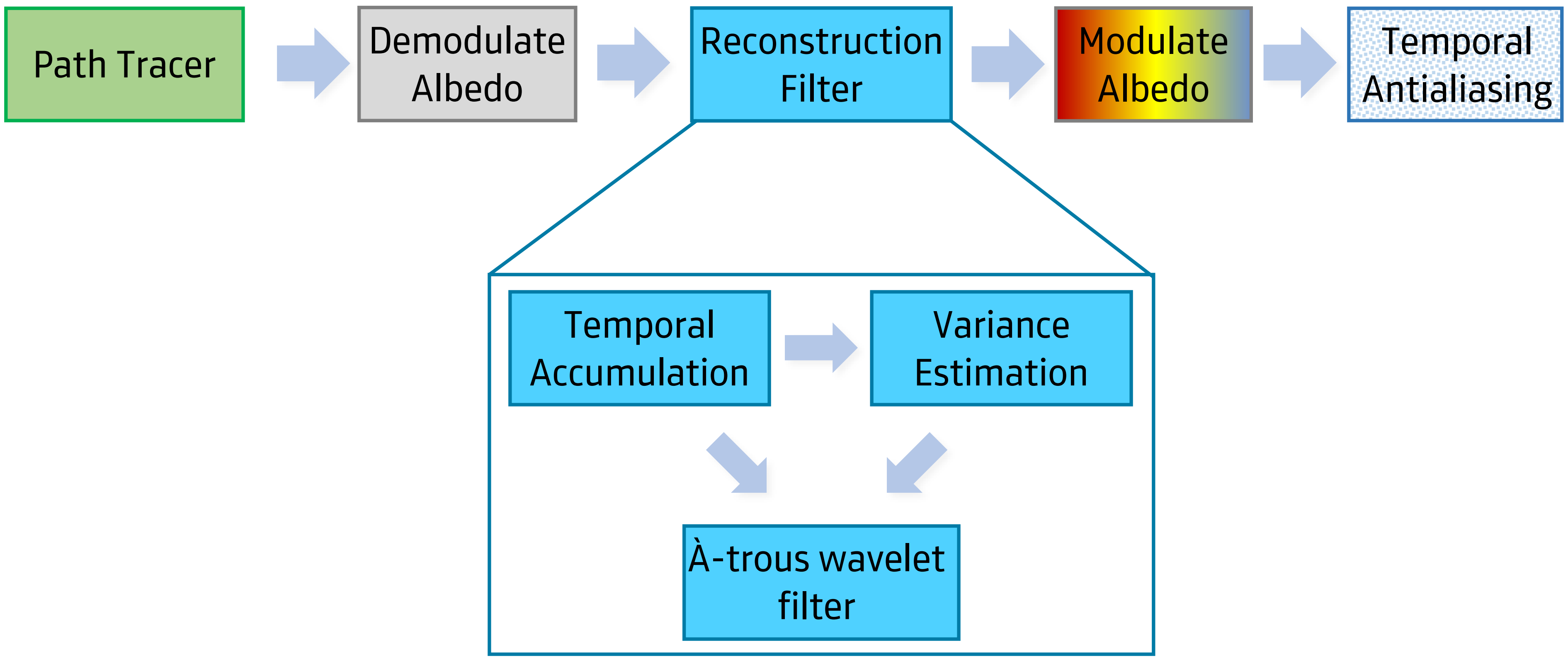


Filter hierarchically, starting small

- Estimate temporal stability after each filter iteration

⟶ Strong blur more likely in early iterations

SVGF

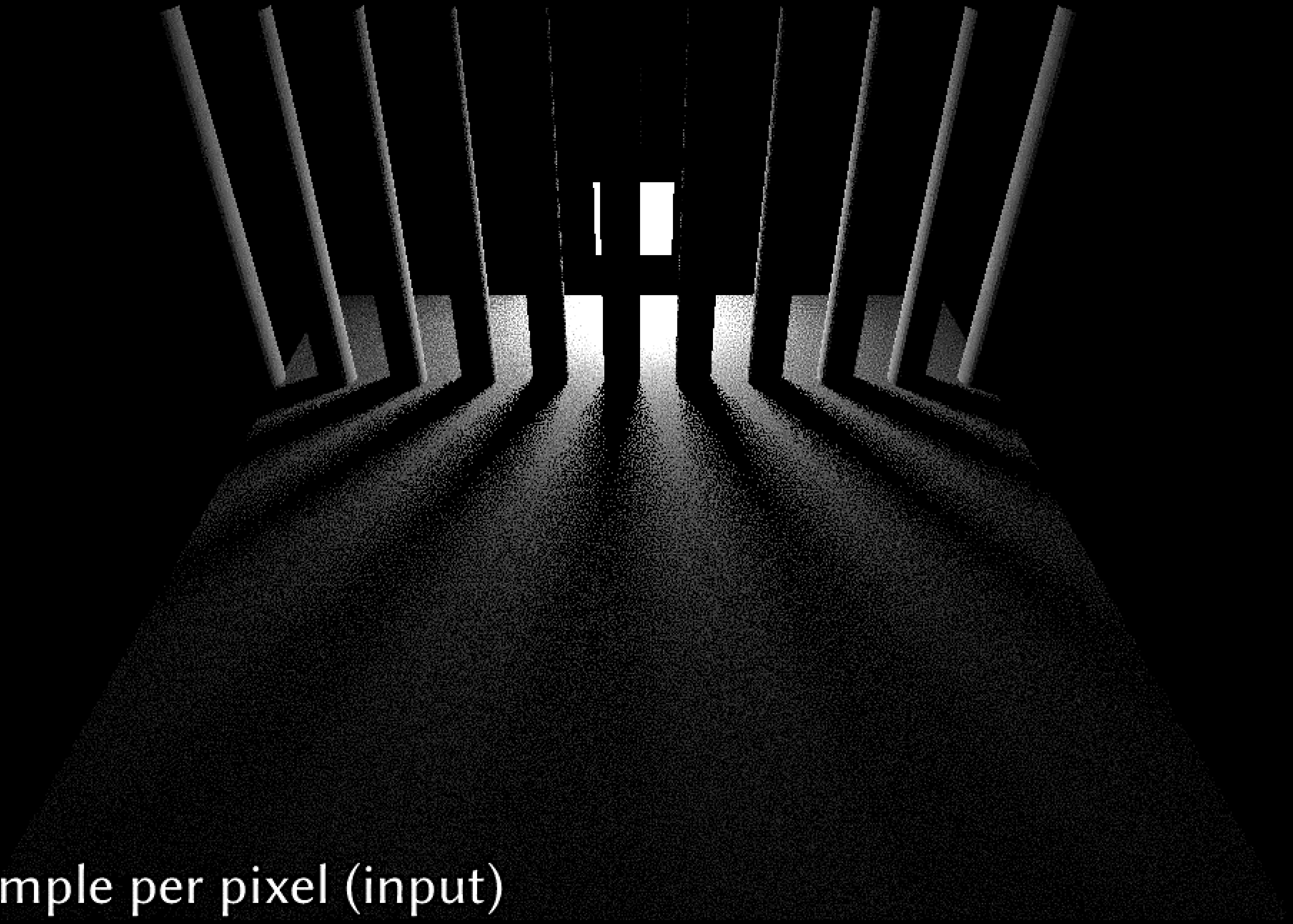

Path Tracer
Demodulate Albedo
Reconstruction Filter
Modulate Albedo
Temporal Antialiasing
Temporal Accumulation
Variance Estimation
À-trous wavelet filter

Reference

SVGF

Overblurring at high-frequency normal maps and materials

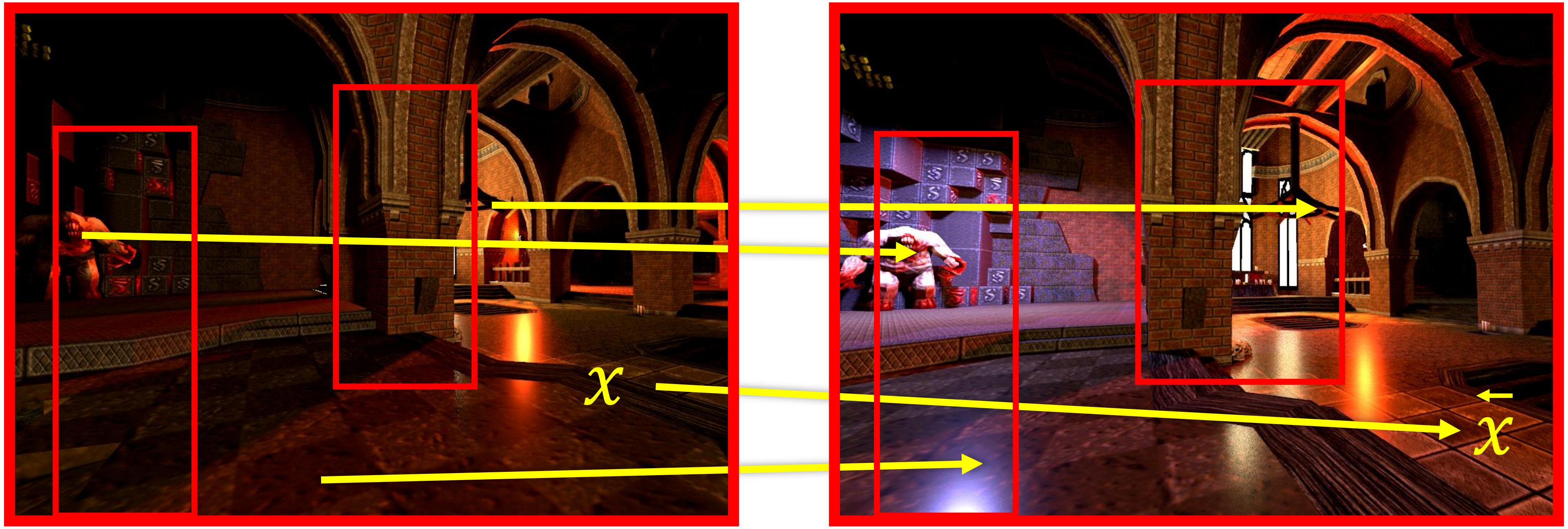
[Schied et al. 2017]



One sample per pixel (input)

Temporal filtering

Screen-space Reprojection



Current frame c_i Previous filtered frame $\hat{c}_{i-1}$

$$\hat{c}_i(x) = \alpha \cdot \boxed{c_i}\boxed{(x)} + (1-\alpha) \cdot \boxed{\hat{c}_{i-1}}\boxed{(\overleftarrow{x})}$$

$$\hat{c}_i(x) = \boxed{\alpha} \cdot c_i(x) + \boxed{(1-\alpha)} \cdot \hat{c}_{i-1}(\overleftarrow{x})$$

- Set α according to changes of the shading function
  - Moving shadows, glossy highlights, flickering light sources, …

- Make α per-pixel weight for local adaptivity

Gradient Estimation for Real-Time Adaptive Temporal Filtering

CHRISTOPH SCHIED, CHRISTOPH PETERS, and CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

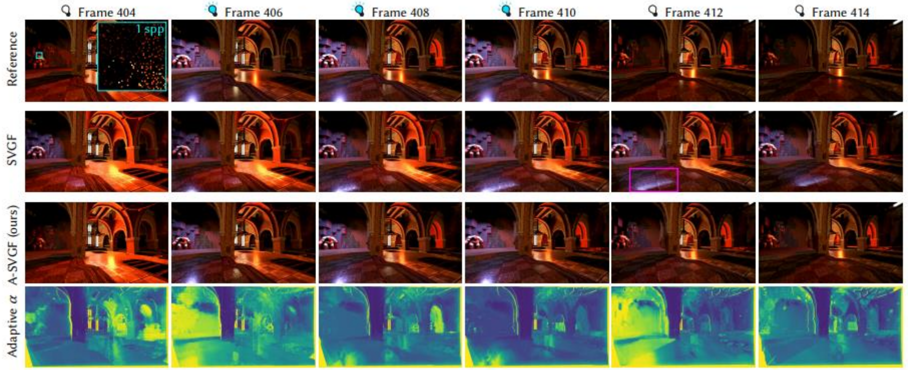
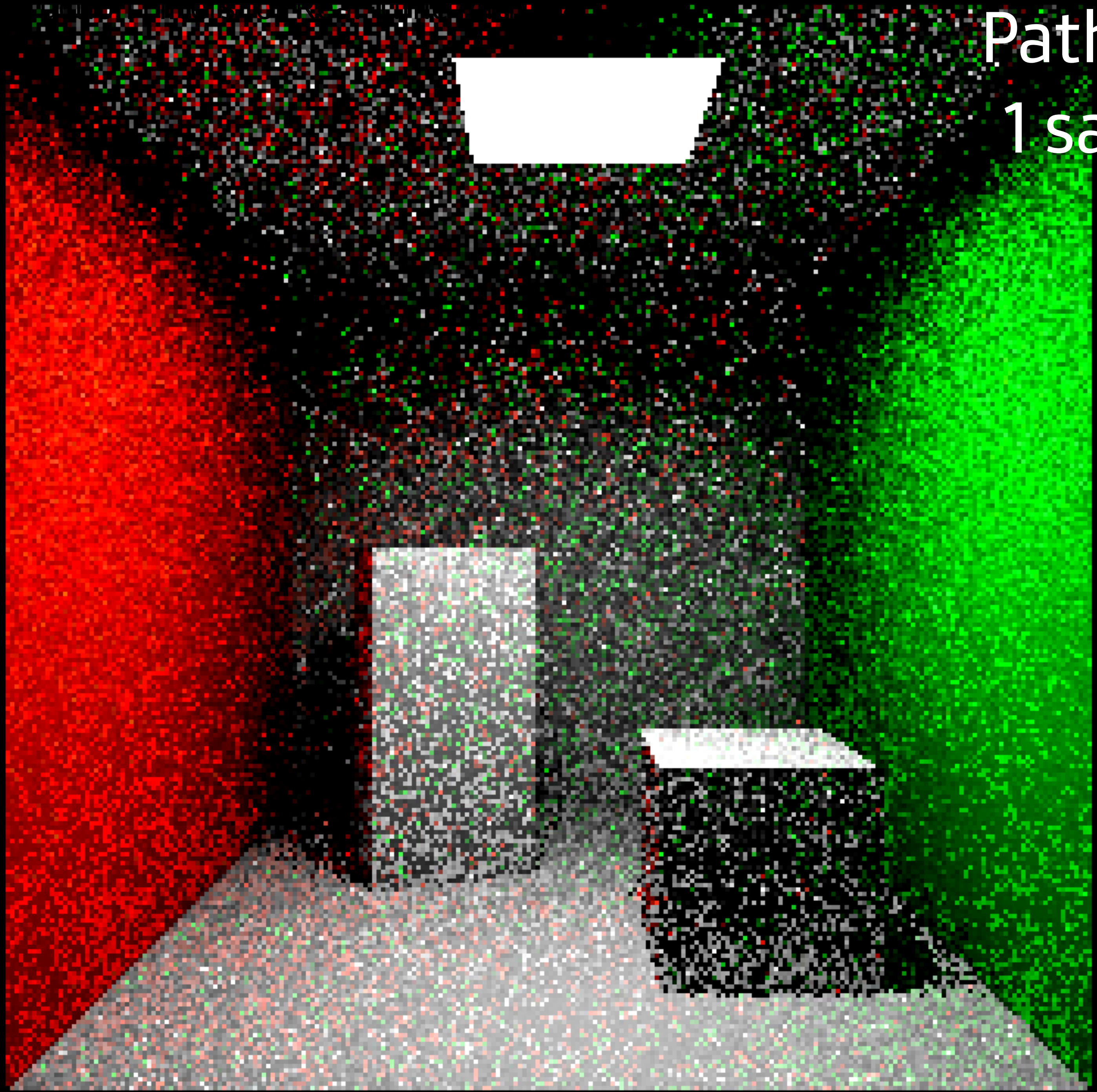


Fig. 1. Results of our novel spatio-temporal reconstruction filter (A-SVGF) for path tracing at one sample per pixel (cyan inset in frame 404) with a resolution of 1280×720. The animation includes a moving camera and a flickering, blue area light. Previous work (SVGF) [Schied et al. 2017] introduces temporal blur such that lighting is still present when the light source is off and glossy highlights leave a trail (magenta box in frame 412). Our temporal filter estimates and reconstructs sparse temporal gradients and uses them to adapt the temporal accumulation factor α per pixel. For example, the regions lit by the flickering blue light have a large α in frames 406 and 412 where the light has been turned on or off. Glossy highlights also receive a large α due to the camera movement. Overall, stale history information is rejected reliably.

With the push towards physically based rendering, stochastic sampling of shading, e.g. using path tracing, is becoming increasingly important in real-time rendering. To achieve high performance, only low sample counts are viable, which necessitates the use of sophisticated reconstruction filters. Recent research on such filters has shown dramatic improvements in both quality and performance. They exploit the coherence of consecutive frames by reusing temporal information to achieve stable, denoised results. However, existing temporal filters often create objectionable artifacts such as ghosting and lag. We propose a novel temporal filter which analyzes the signal over time to derive adaptive temporal accumulation factors per pixel. It repurposes a subset of the shading budget to sparsely sample and reconstruct the temporal gradient. This allows us to reliably detect sudden changes of the sampled signal and to drop stale history information. We create gradient samples through forward-projection of surface samples from the previous frame into the current frame and by reevaluating the shading samples using the same random sequence. We apply our filter to improve real-time path tracers. Compared to previous work, we show a significant reduction of lag and ghosting as well as

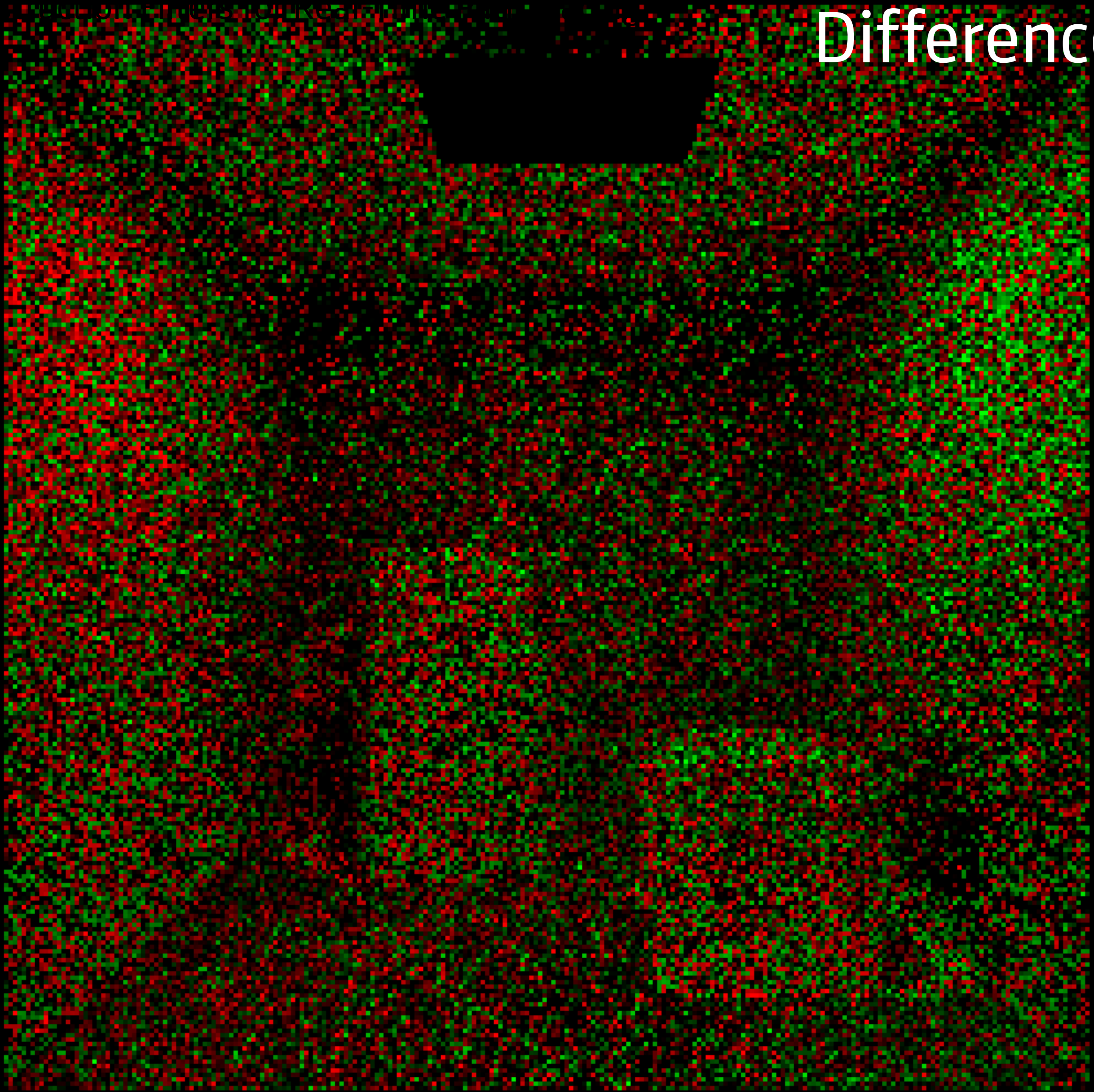
Authors' address: Christoph Schied, schied@kit.edu; Christoph Peters, christoph.peters@kit.edu; Carsten Dachsbacher, dachsbacher@kit.edu, Karlsruhe Institute of Technology, Am Fasanengarten 5, 76131, Karlsruhe, Germany.

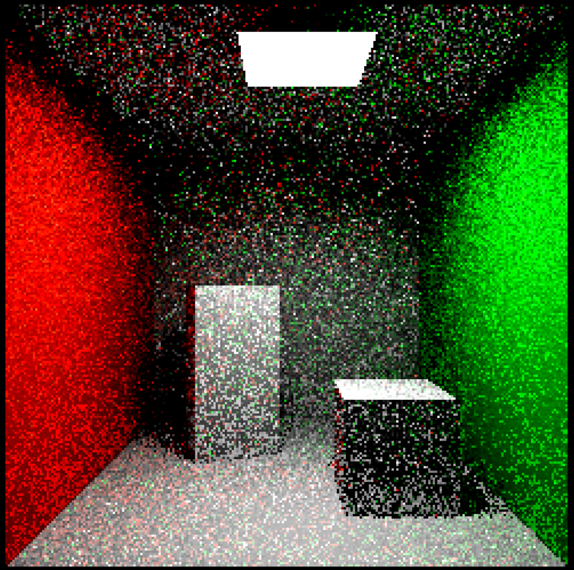
Proc. ACM Comput. Graph. Interact. Tech., Vol. 1, No. 2, Article 24. Publication date: August 2018.



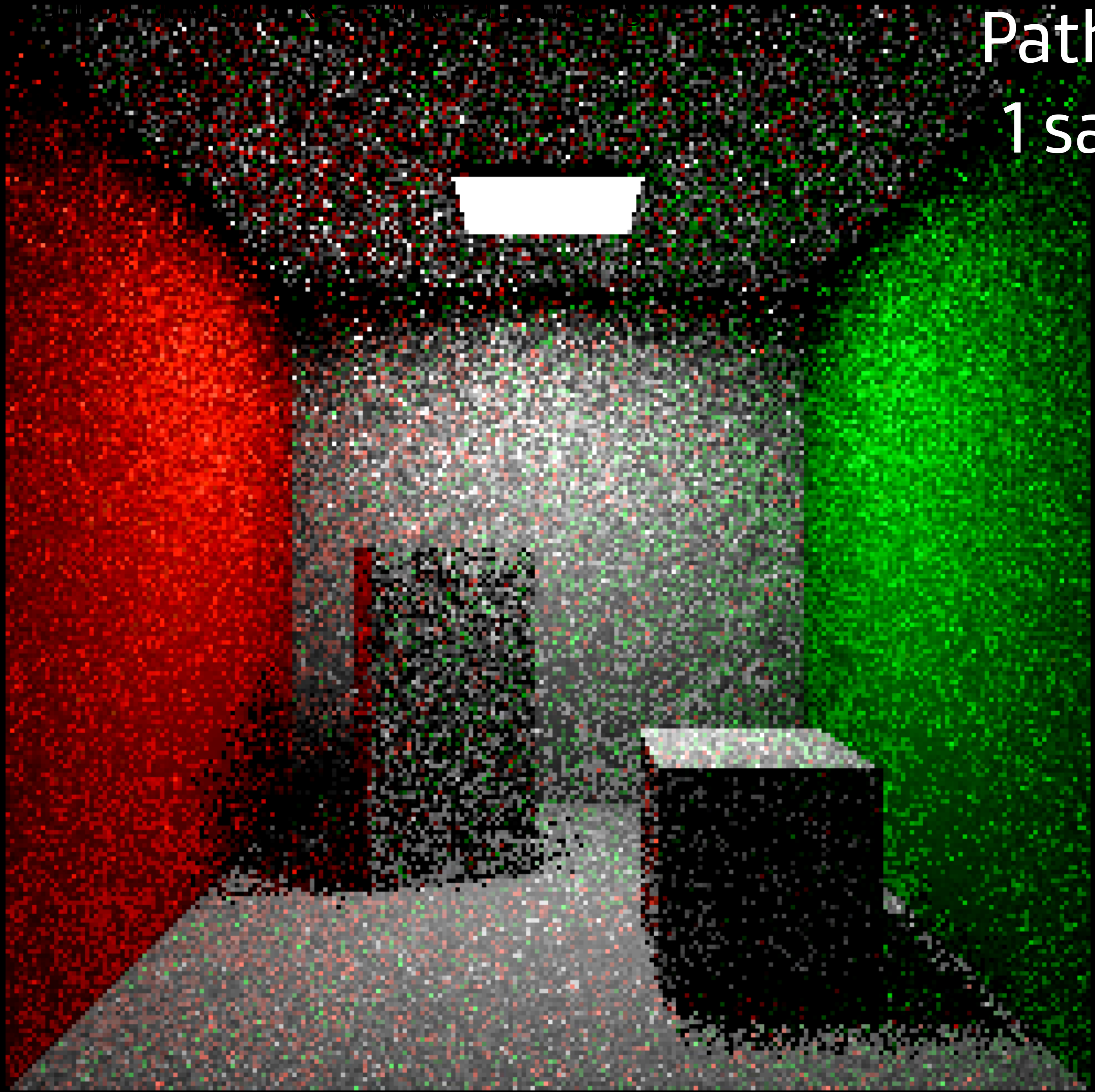
Path tracer output
1 sample per pixel

Difference of luminance
green positive
red negative

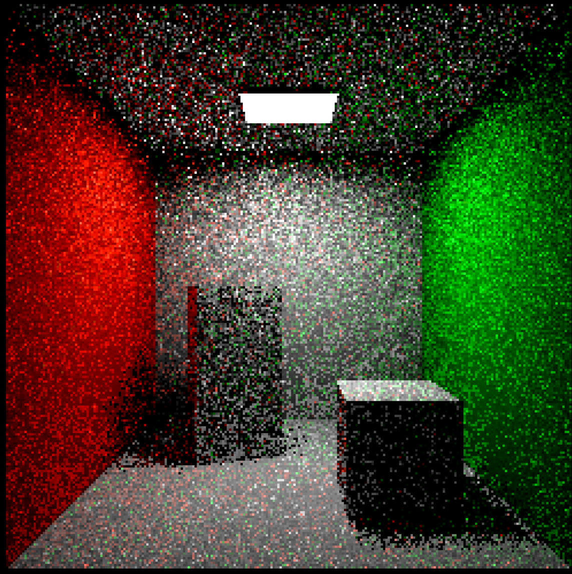


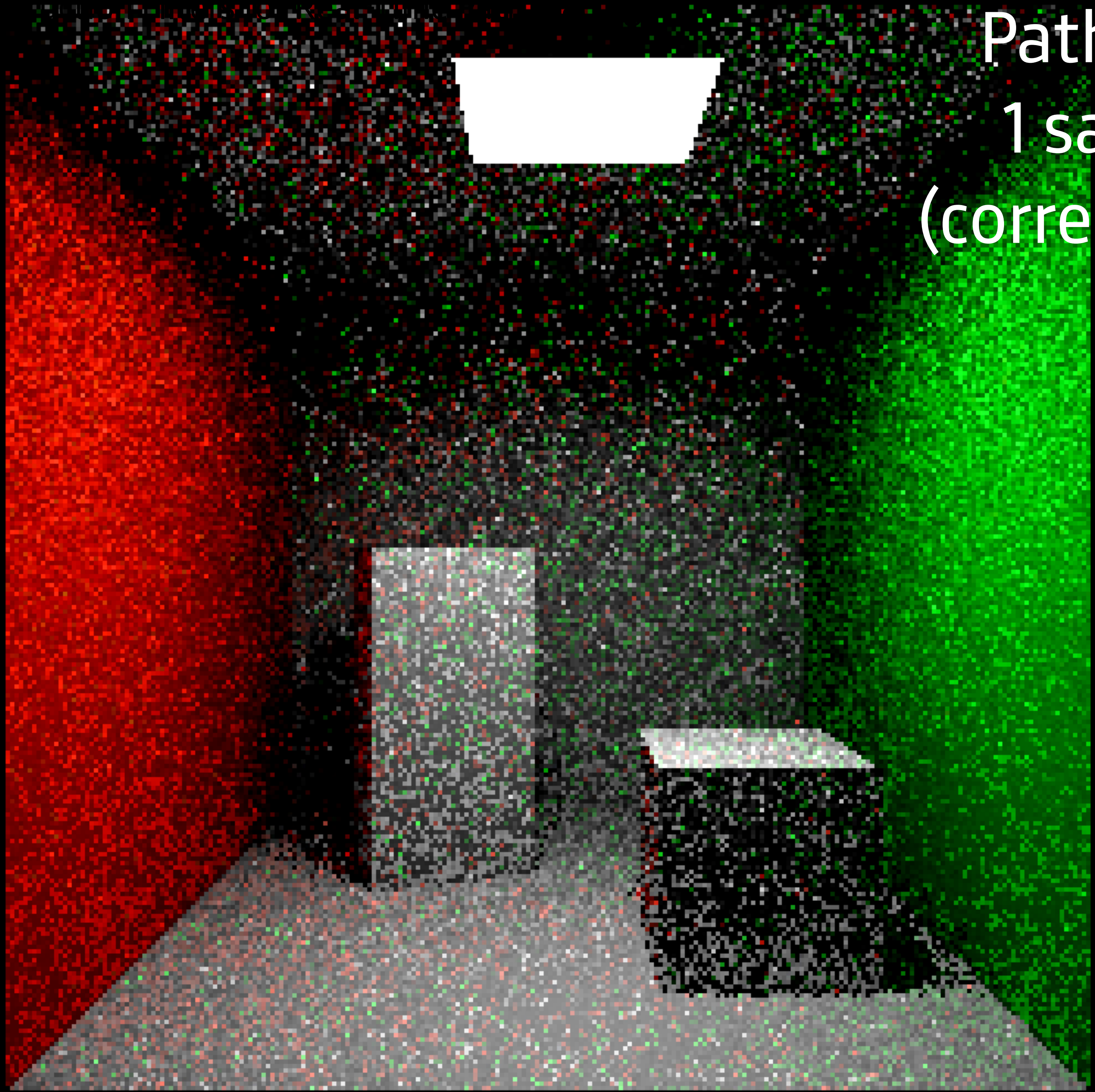


Path tracer output
1 sample per pixel



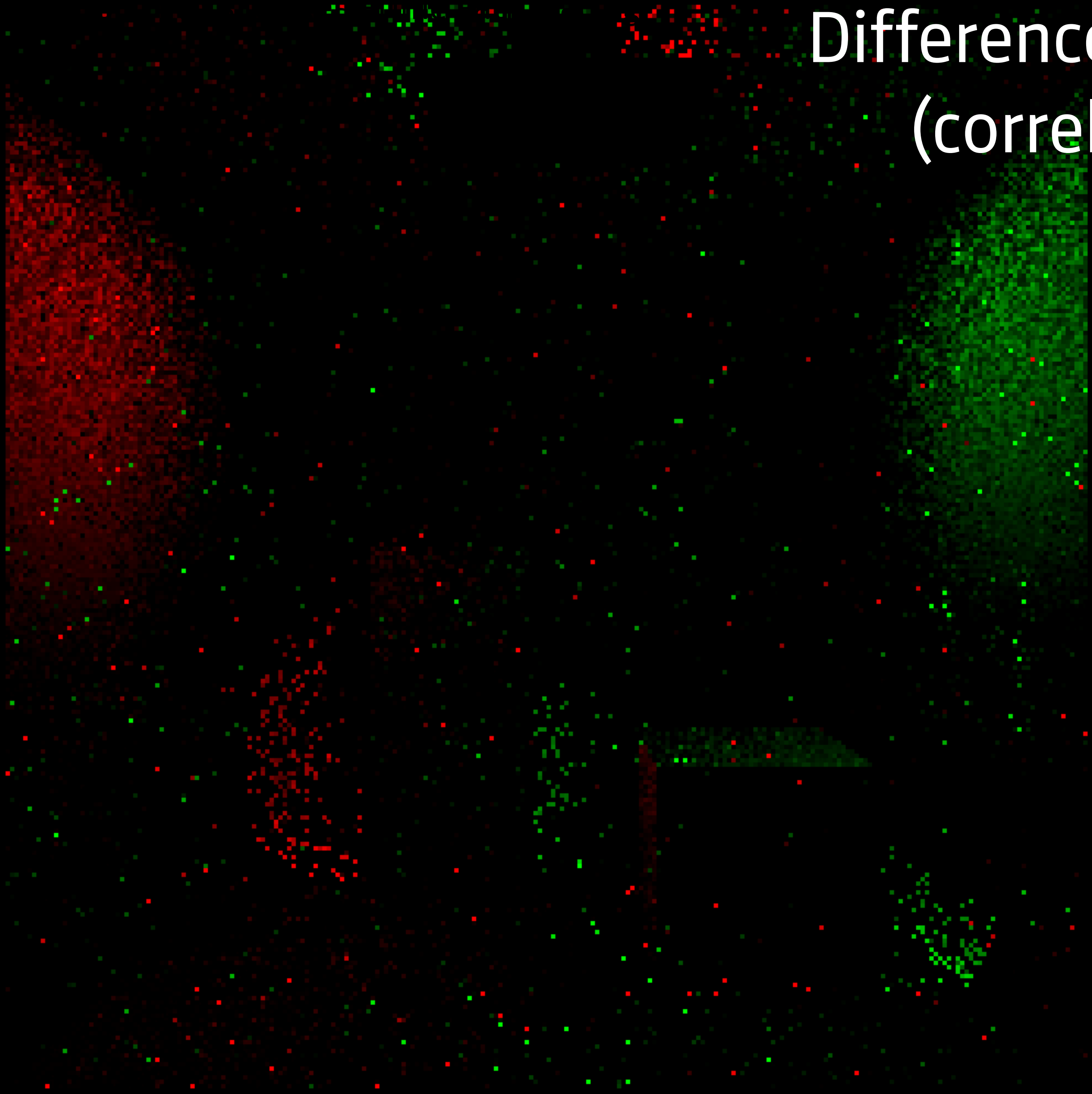
Difference of luminance
green positive
red negative

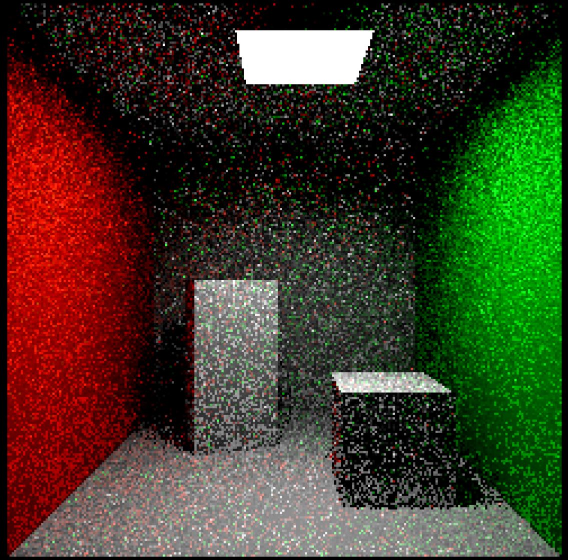




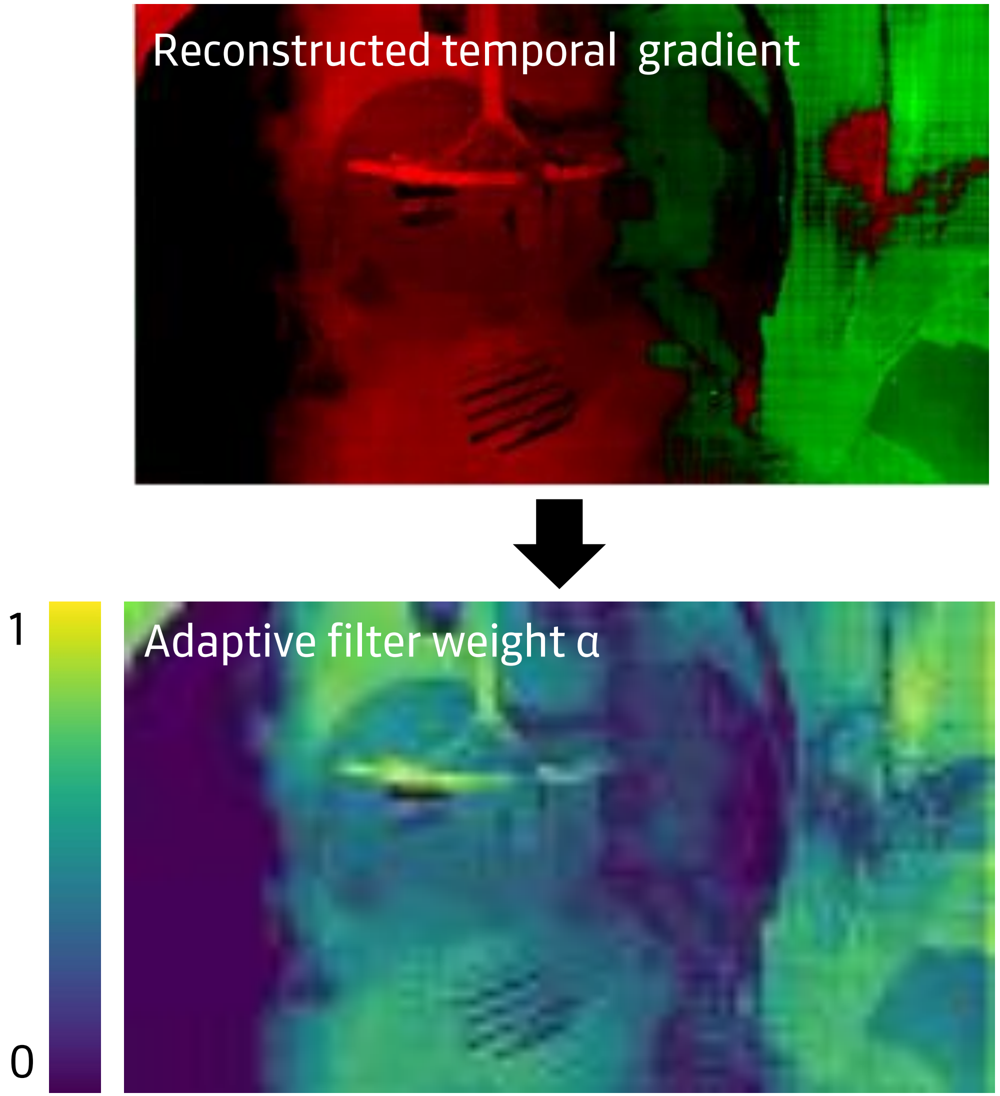
Path tracer output
1 sample per pixel
(correlated samples)

Difference of luminance
(correlated samples)
green positive
red negative



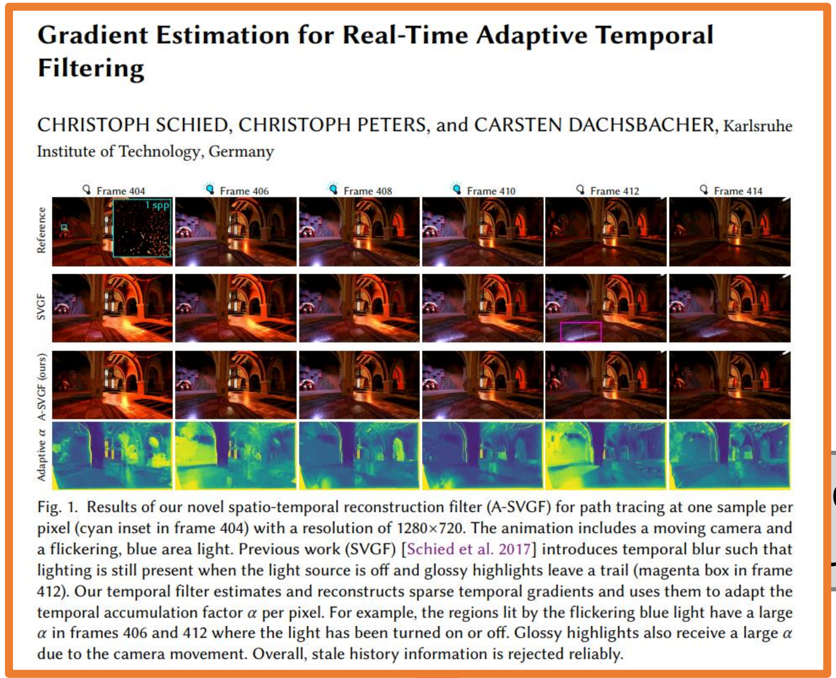


Adaptive temporal filter weight



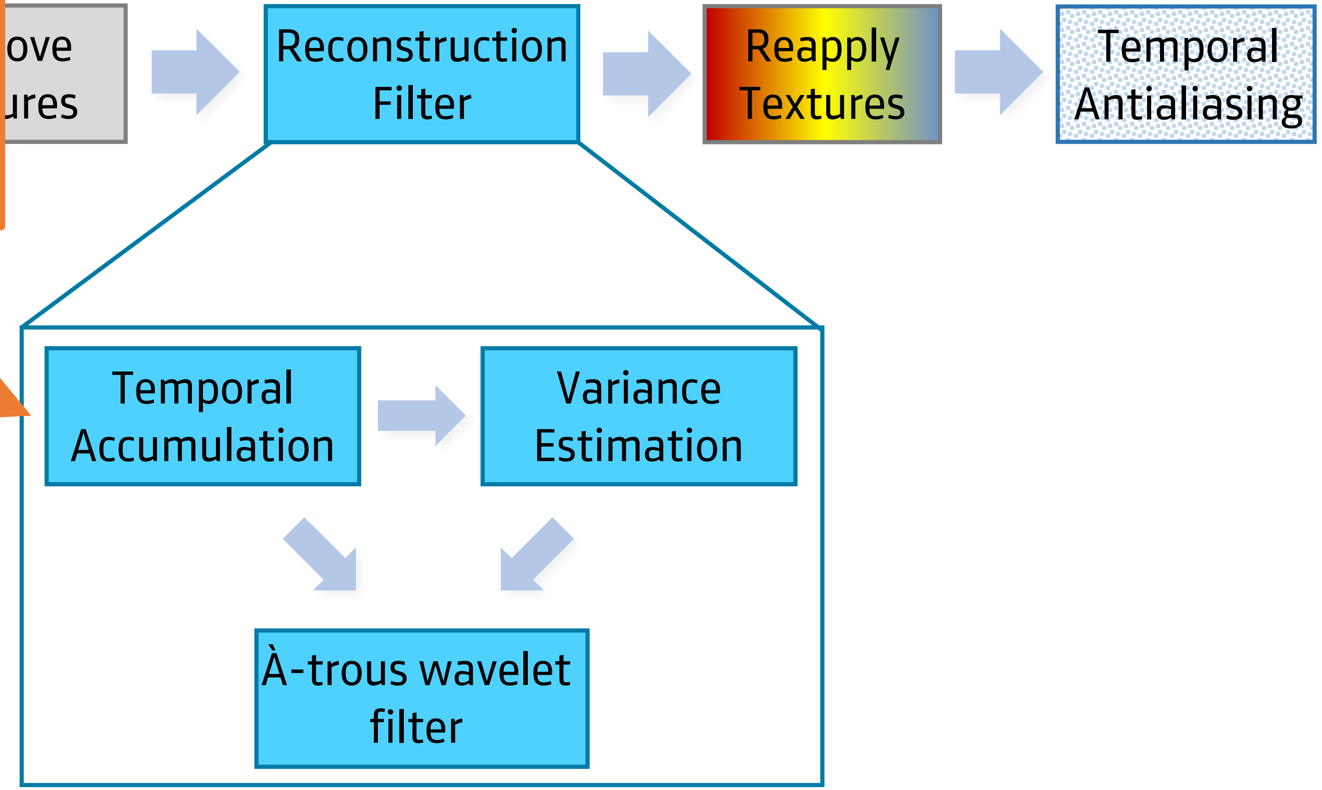
- Sample and reconstruct temporal gradient
- Change α according to relative rate of change

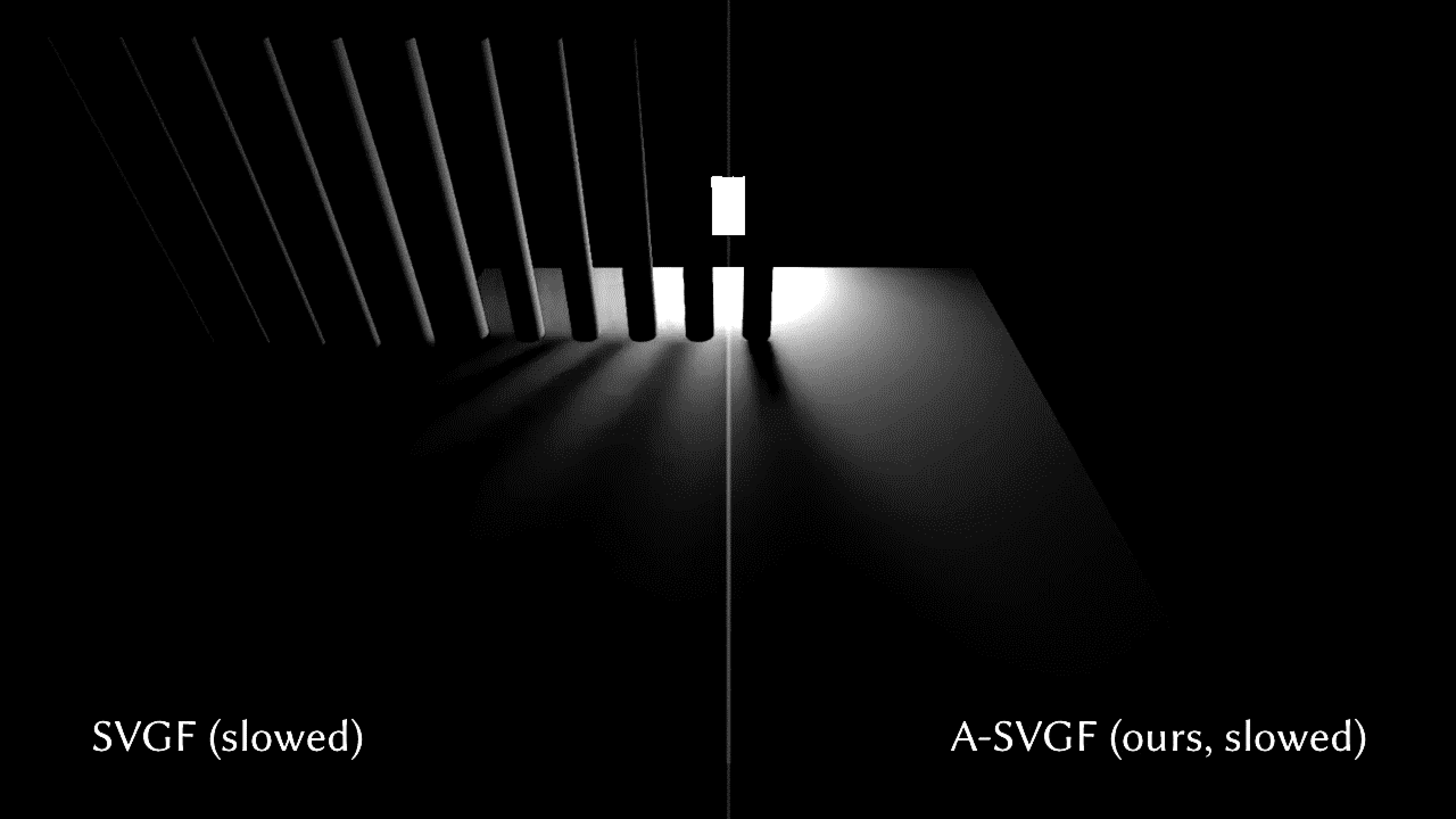
A-SVGF


Gradient Estimation for Real-Time Adaptive Temporal Filtering

CHRISTOPH SCHIED, CHRISTOPH PETERS, and CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

Fig. 1. Results of our novel spatio-temporal reconstruction filter (A-SVGF) for path tracing at one sample per pixel (cyan inset in frame 404) with a resolution of 1280×720. The animation includes a moving camera and a flickering, blue area light. Previous work (SVGF) [Schied et al. 2017] introduces temporal blur such that lighting is still present when the light source is off and glossy highlights leave a trail (magenta box in frame 412). Our temporal filter estimates and reconstructs sparse temporal gradients and uses them to adapt the temporal accumulation factor α per pixel. For example, the regions lit by the flickering blue light have a large α in frames 406 and 412 where the light has been turned on or off. Glossy highlights also receive a large α due to the camera movement. Overall, stale history information is rejected reliably.




SVGF (slowed)
A-SVGF (ours, slowed)

Q2VKPT

Quake 2 Vulkan Path tracer

- Experiment: how far can we push the state of the art
- Basis for NVIDIA's Quake 2 RTX
- A-SVGF for denoising
- Why Quake 2:
    - Lot's of area light sources
    - Manageable complexity for a prototype


100


100


frame time 15.24 ms
instance geometry 0.02 ms
bvh update 0.52 ms
asvgf gradient samples 0.28 ms
path tracer 10.89 ms
asvgf full 3.44 ms
asvgf reconstruct gradient 0.27 ms
asvgf temporal 0.68 ms
asvgf atrous 2.16 ms
asvgf taa 0.32 ms
2560x1440, RTX2080 Ti

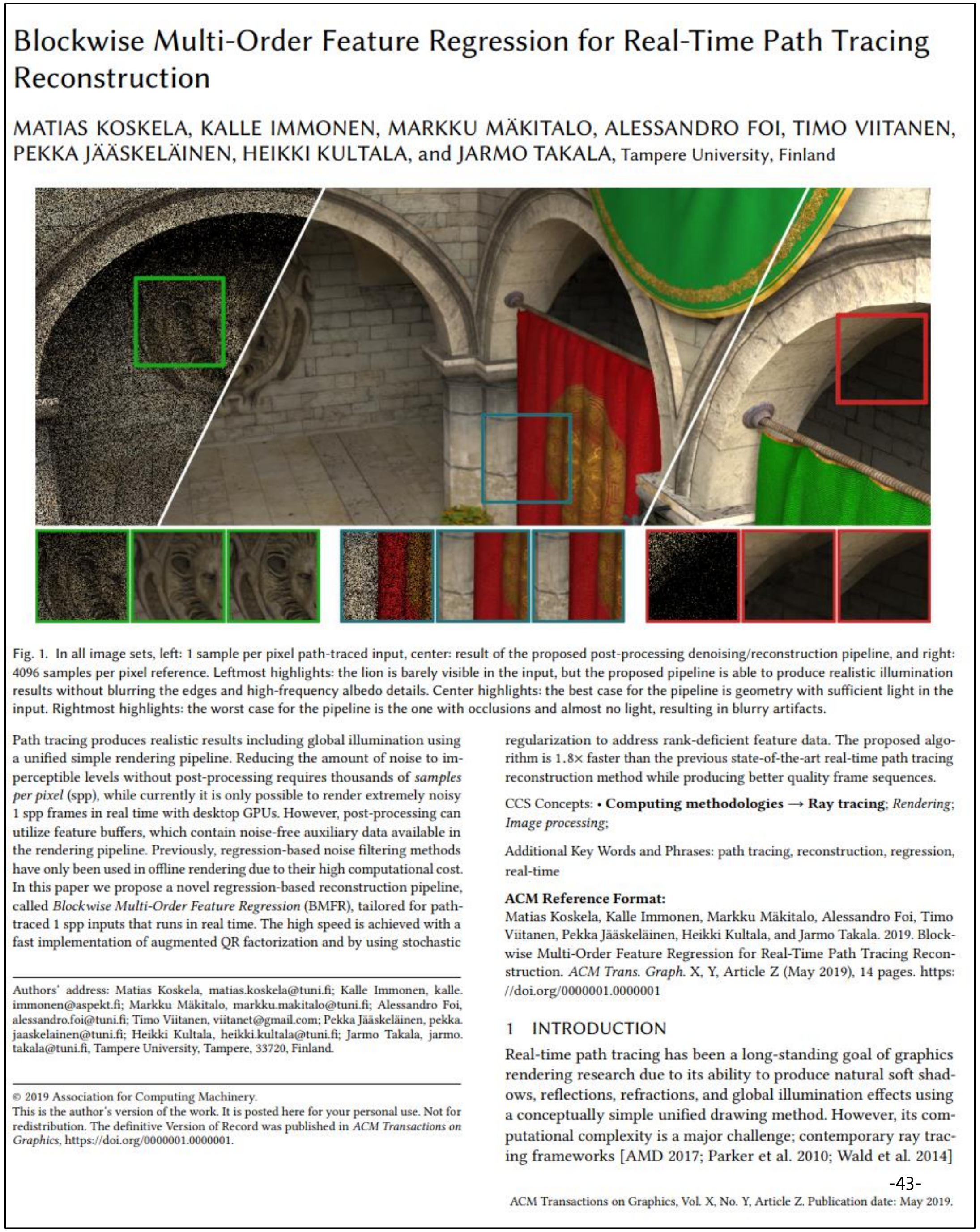

No Transmission
• Need Albedo demodulation in reflection to prevent overblurring
• Would need denoising of sampling of Fresnel


Noise where the filter lacks history


STROGGOS

Regression-based reconstruction

- Screen-space multi-order regression in tiles
- Fast (2.4ms at 720p, Titan X Pascal)
- Two temporal filters
  - Pre- and post-filtering the Regression result



Blockwise Multi-Order Feature Regression for Real-Time Path Tracing Reconstruction

MATIAS KOSKELA, KALLE IMMONEN, MARKKU MÄKITALO, ALESSANDRO FOI, TIMO VIITANEN, PEKKA JÄÄSKELÄINEN, HEIKKI KULTALA, and JARMO TAKALA, Tampere University, Finland

Fig. 1. In all image sets, left: 1 sample per pixel path-traced input, center: result of the proposed post-processing denoising/reconstruction pipeline, and right: 4096 samples per pixel reference. Leftmost highlights: the lion is barely visible in the input, but the proposed pipeline is able to produce realistic illumination results without blurring the edges and high-frequency albedo details. Center highlights: the best case for the pipeline is geometry with sufficient light in the input. Rightmost highlights: the worst case for the pipeline is the one with occlusions and almost no light, resulting in blurry artifacts.

Path tracing produces realistic results including global illumination using a unified simple rendering pipeline. Reducing the amount of noise to imperceptible levels without post-processing requires thousands of *samples per pixel* (spp), while currently it is only possible to render extremely noisy 1 spp frames in real time with desktop GPUs. However, post-processing can utilize feature buffers, which contain noise-free auxiliary data available in the rendering pipeline. Previously, regression-based noise filtering methods have only been used in offline rendering due to their high computational cost. In this paper we propose a novel regression-based reconstruction pipeline, called *Blockwise Multi-Order Feature Regression* (BMFR), tailored for path-traced 1 spp inputs that runs in real time. The high speed is achieved with a fast implementation of augmented QR factorization and by using stochastic regularization to address rank-deficient feature data. The proposed algorithm is 1.8× faster than the previous state-of-the-art real-time path tracing reconstruction method while producing better quality frame sequences.

CCS Concepts: • **Computing methodologies → Ray tracing**; *Rendering*; *Image processing*;

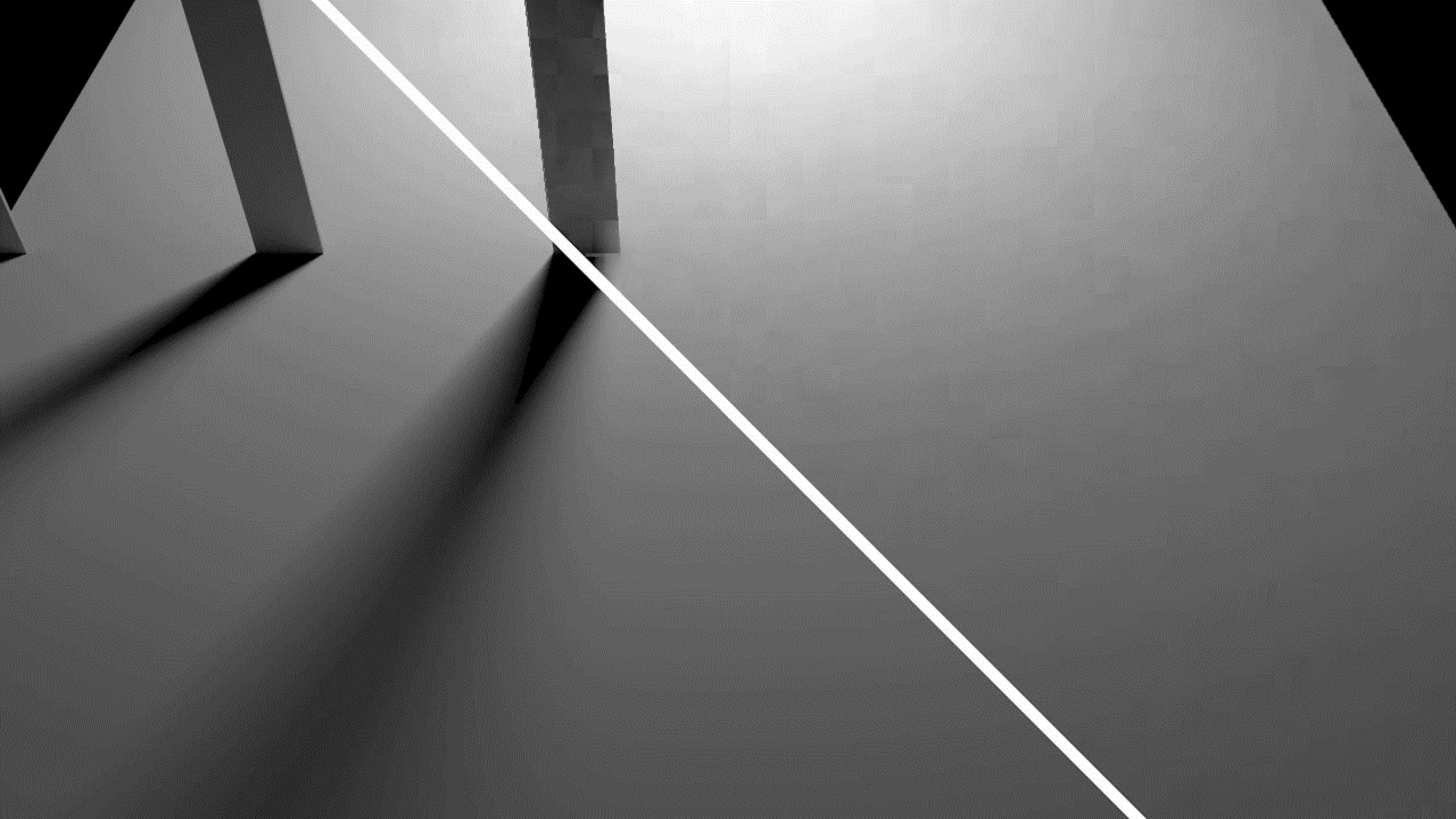
Additional Key Words and Phrases: path tracing, reconstruction, regression, real-time

ACM Reference Format:
Matias Koskela, Kalle Immonen, Markku Mäkitalo, Alessandro Foi, Timo Viitanen, Pekka Jääskeläinen, Heikki Kultala, and Jarmo Takala. 2019. Blockwise Multi-Order Feature Regression for Real-Time Path Tracing Reconstruction. *ACM Trans. Graph.* X, Y, Article Z (May 2019), 14 pages. https://doi.org/0000001.0000001

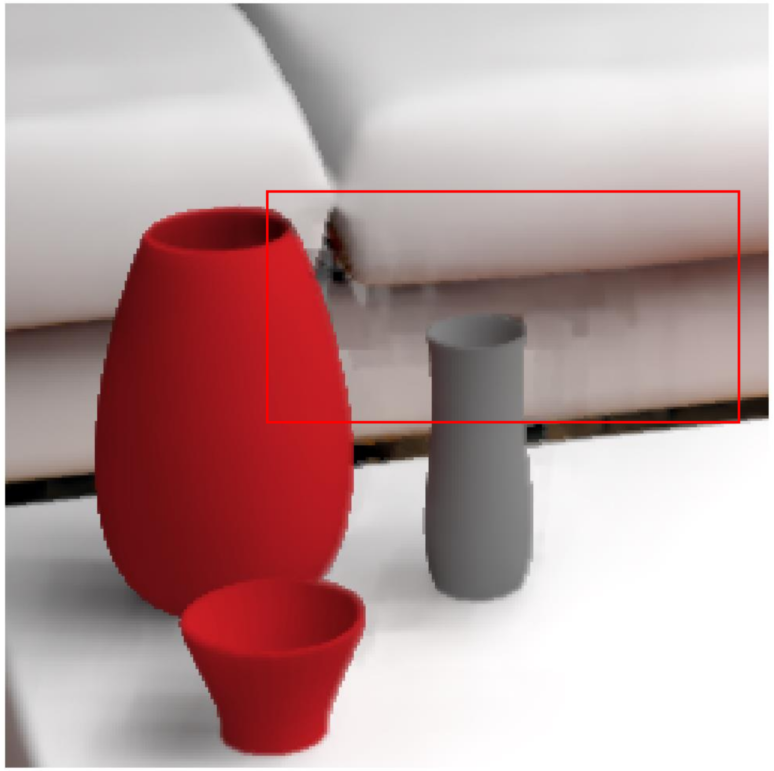
1 INTRODUCTION

Real-time path tracing has been a long-standing goal of graphics rendering research due to its ability to produce natural soft shadows, reflections, refractions, and global illumination effects using a conceptually simple unified drawing method. However, its computational complexity is a major challenge; contemporary ray tracing frameworks [AMD 2017; Parker et al. 2010; Wald et al. 2014]

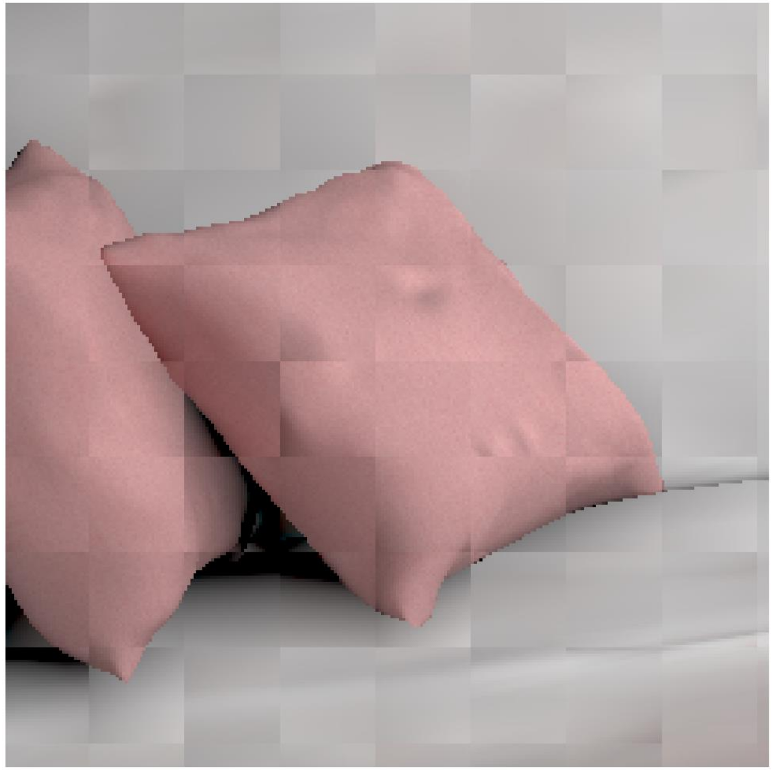
Authors' address: Matias Koskela, matias.koskela@tuni.fi; Kalle Immonen, kalle.immonen@aspekt.fi; Markku Mäkitalo, markku.makitalo@tuni.fi; Alessandro Foi, alessandro.foi@tuni.fi; Timo Viitanen, viitanet@gmail.com; Pekka Jääskeläinen, pekka.jaaskelainen@tuni.fi; Heikki Kultala, heikki.kultala@tuni.fi; Jarmo Takala, jarmo.takala@tuni.fi, Tampere University, Tampere, 33720, Finland.



Under motion

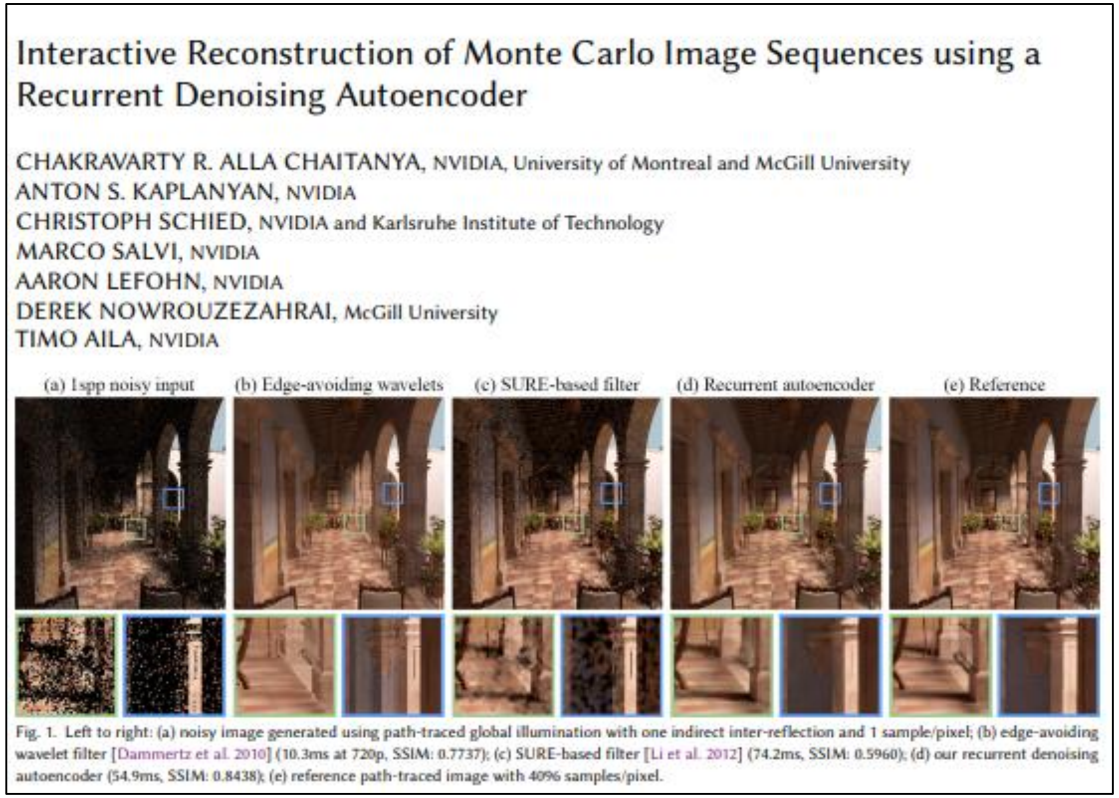


Without temporal filter



Converged temporal filter

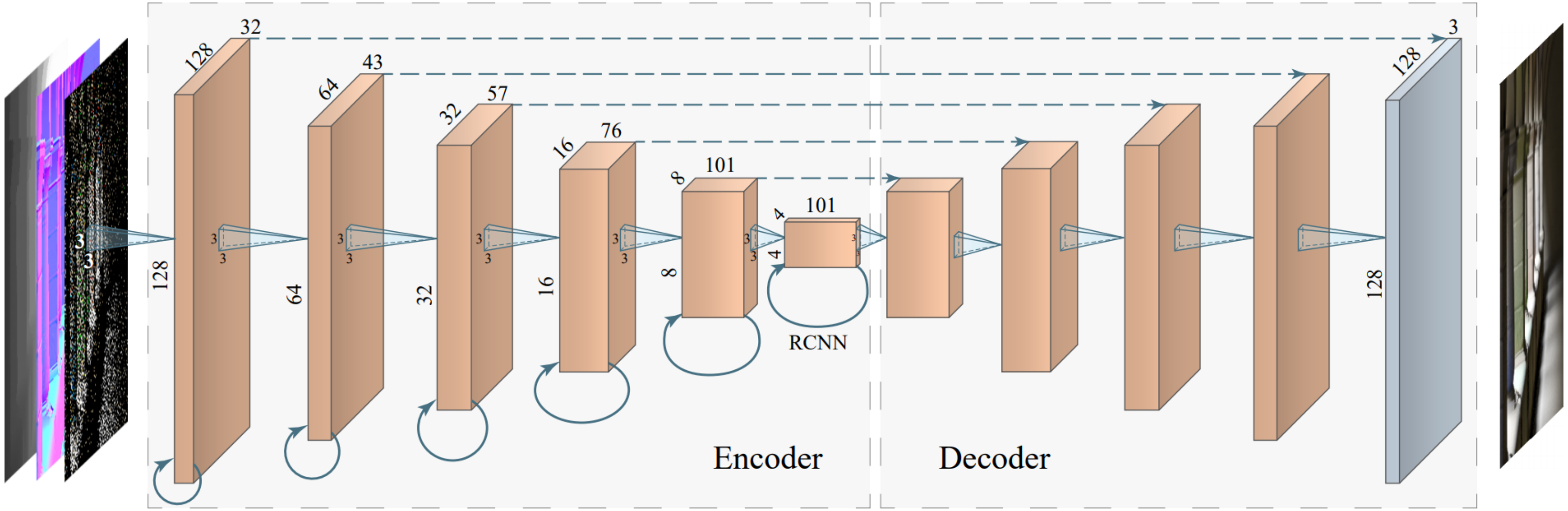
Deep Learning approaches

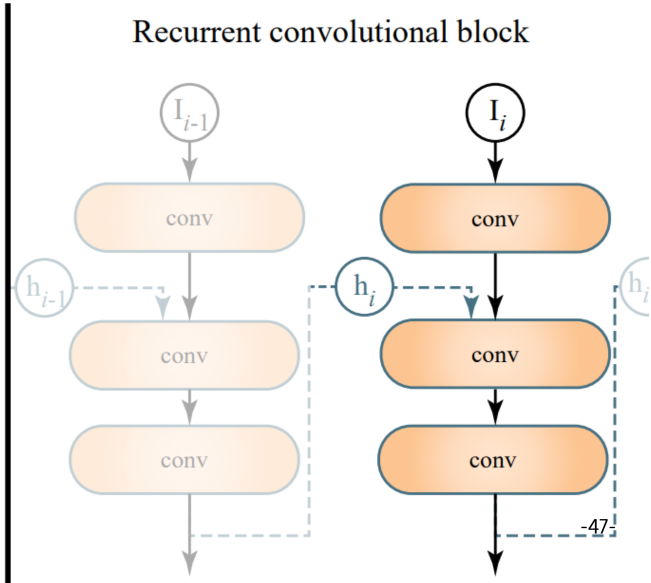

Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder

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

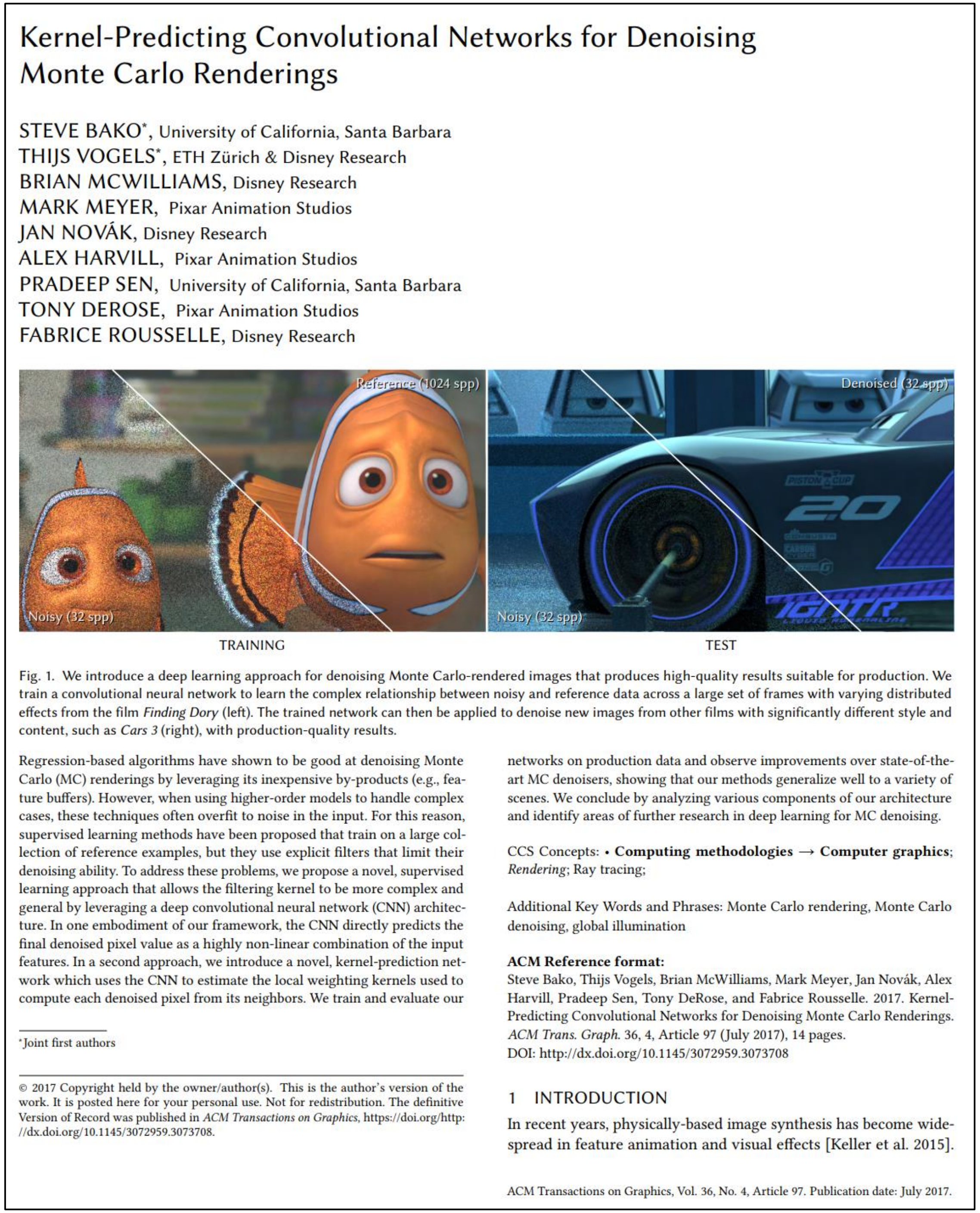
Fig. 1. Left to right: (a) noisy image generated using path-traced global illumination with one indirect inter-reflection and 1 sample/pixel; (b) edge-avoiding wavelet filter [Dammertz et al. 2010] (10.3ms at 720p, SSIM: 0.7737); (c) SURE-based filter [Li et al. 2012] (74.2ms, SSIM: 0.5960); (d) our recurrent denoising autoencoder (54.9ms, SSIM: 0.8438); (e) reference path-traced image with 4096 samples/pixel.

- Direct prediction with U-Net architecture (Autoencoder with skips)
- Recurrent blocks for temporal stability
- 55ms @ 720p, NVIDIA Titan X (Pascal)






RAE
A-RAE (ours)



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 HARVILL, Pixar Animation Studios
PRADEEP SEN, University of California, Santa Barbara
TONY DEROSE, Pixar Animation Studios
FABRICE ROUSSELLE, Disney Research

Reference (1024 spp) | Denoised (32 spp)
Noisy (32 spp) | Noisy (32 spp)

TRAINING | TEST

Fig. 1. We introduce a deep learning approach for denoising Monte Carlo-rendered images that produces high-quality results suitable for production. We train a convolutional neural network to learn the complex relationship between noisy and reference data across a large set of frames with varying distributed effects from the film *Finding Dory* (left). The trained network can then be applied to denoise new images from other films with significantly different style and content, such as *Cars 3* (right), with production-quality results.

Regression-based algorithms have shown to be good at denoising Monte Carlo (MC) renderings by leveraging its inexpensive by-products (e.g., feature buffers). However, when using higher-order models to handle complex cases, these techniques often overfit to noise in the input. For this reason, supervised learning methods have been proposed that train on a large collection of reference examples, but they use explicit filters that limit their denoising ability. To address these problems, we propose a novel, supervised learning approach that allows the filtering kernel to be more complex and general by leveraging a deep convolutional neural network (CNN) architecture. In one embodiment of our framework, the CNN directly predicts the final denoised pixel value as a highly non-linear combination of the input features. In a second approach, we introduce a novel, kernel-prediction network which uses the CNN to estimate the local weighting kernels used to compute each denoised pixel from its neighbors. We train and evaluate our networks on production data and observe improvements over state-of-the-art MC denoisers, showing that our methods generalize well to a variety of scenes. We conclude by analyzing various components of our architecture and identify areas of further research in deep learning for MC denoising.

CCS Concepts: • **Computing methodologies** → **Computer graphics**; *Rendering*; Ray tracing;

Additional Key Words and Phrases: Monte Carlo rendering, Monte Carlo denoising, global illumination

ACM Reference format:
Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony DeRose, and Fabrice Rousselle. 2017. Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings. *ACM Trans. Graph.* 36, 4, Article 97 (July 2017), 14 pages.
DOI: http://dx.doi.org/10.1145/3072959.3073708

*Joint first authors

1 INTRODUCTION

In recent years, physically-based image synthesis has become widespread in feature animation and visual effects [Keller et al. 2015].

- Output of the network is a per-pixel convolution kernel
- Filter kernel could be computed at lower precision
  - Int4/int8 inference?
- Has not been demonstrated for real-time / interactive yet

- Training with non-converged references is possible

- Training data is much cheaper
  - More variation in data → better generalization

- Online-learning for specializing denoiser to current inputs possible

arXiv:1803.04189v3 [cs.CV] 29 Oct 2018

Noise2Noise: Learning Image Restoration without Clean Data

Jaakko Lehtinen[1 2] Jacob Munkberg[1] Jon Hasselgren[1] Samuli Laine[1] Tero Karras[1] Miika Aittala[3] Timo Aila[1]

Abstract

We apply basic statistical reasoning to signal reconstruction by machine learning – learning to map corrupted observations to clean signals – with a simple and powerful conclusion: it is possible to learn to restore images by only looking at corrupted examples, at performance at and sometimes exceeding training using clean data, without explicit image priors or likelihood models of the corruption. In practice, we show that a single model learns photographic noise removal, denoising synthetic Monte Carlo images, and reconstruction of undersampled MRI scans – all corrupted by different processes – based on noisy data only.

1. Introduction

Signal reconstruction from corrupted or incomplete measurements is an important subfield of statistical data analysis. Recent advances in deep neural networks have sparked significant interest in avoiding the traditional, explicit a priori statistical modeling of signal corruptions, and instead *learning* to map corrupted observations to the unobserved clean versions. This happens by training a regression model, e.g., a convolutional neural network (CNN), with a large number of pairs $(\hat{x}_i, y_i)$ of corrupted inputs $\hat{x}_i$ and clean targets y_i and minimizing the empirical risk

$$\underset{\theta}{\operatorname{argmin}} \sum_i L\left(f_\theta(\hat{x}_i),\, y_i\right), \tag{1}$$

where f_θ is a parametric family of mappings (e.g., CNNs), under the loss function L. We use the notation $\hat{x}$ to underline the fact that the corrupted input $\hat{x} \sim p(\hat{x}|y_i)$ is a random variable distributed according to the clean target. Training data may include, for example, pairs of short and long exposure photographs of the same scene, incomplete and complete k-space samplings of magnetic resonance images, fast-but-noisy and slow-but-converged ray-traced renderings of a synthetic scene, etc. Significant advances have been reported in several applications, including Gaussian denoising, de-JPEG, text removal (Mao et al., 2016), super-resolution (Ledig et al., 2017), colorization (Zhang et al., 2016), and image inpainting (Iizuka et al., 2017). Yet, obtaining clean training targets is often difficult or tedious: a noise-free photograph requires a long exposure; full MRI sampling precludes dynamic subjects; etc.

In this work, we observe that we can often *learn to turn bad images into good images by only looking at bad images*, and do this just as well – sometimes even better – as if we were using clean examples. Further, we require neither an explicit statistical likelihood model of the corruption nor an image prior, and instead learn these indirectly from the training data. (Indeed, in one of our examples, synthetic Monte Carlo renderings, the non-stationary noise cannot be characterized analytically.) In addition to denoising, our observation is directly applicable to inverse problems such as MRI reconstruction from undersampled data. While our conclusion is almost trivial from a statistical perspective, it significantly eases practical learned signal reconstruction by lifting requirements on availability of training data.

The reference TensorFlow implementation for Noise2Noise training is available on GitHub.[1]

2. Theoretical Background

Assume that we have a set of unreliable measurements $(y_1, y_2, \ldots)$ of the room temperature. A common strategy for estimating the true unknown temperature is to find a number z that has the smallest average deviation from the measurements according to some loss function L:

$$\underset{z}{\operatorname{argmin}}\, \mathbb{E}_y\{L(z, y)\}. \tag{2}$$

For the L_2 loss $L(z, y) = (z - y)^2$, this minimum is found at the arithmetic mean of the observations:

$$z = \mathbb{E}_y\{y\}. \tag{3}$$

The L_1 loss, the sum of absolute deviations $L(z, y) = |z - y|$, in turn, has its optimum at the median of the observations. The general class of deviation-minimizing estimators are

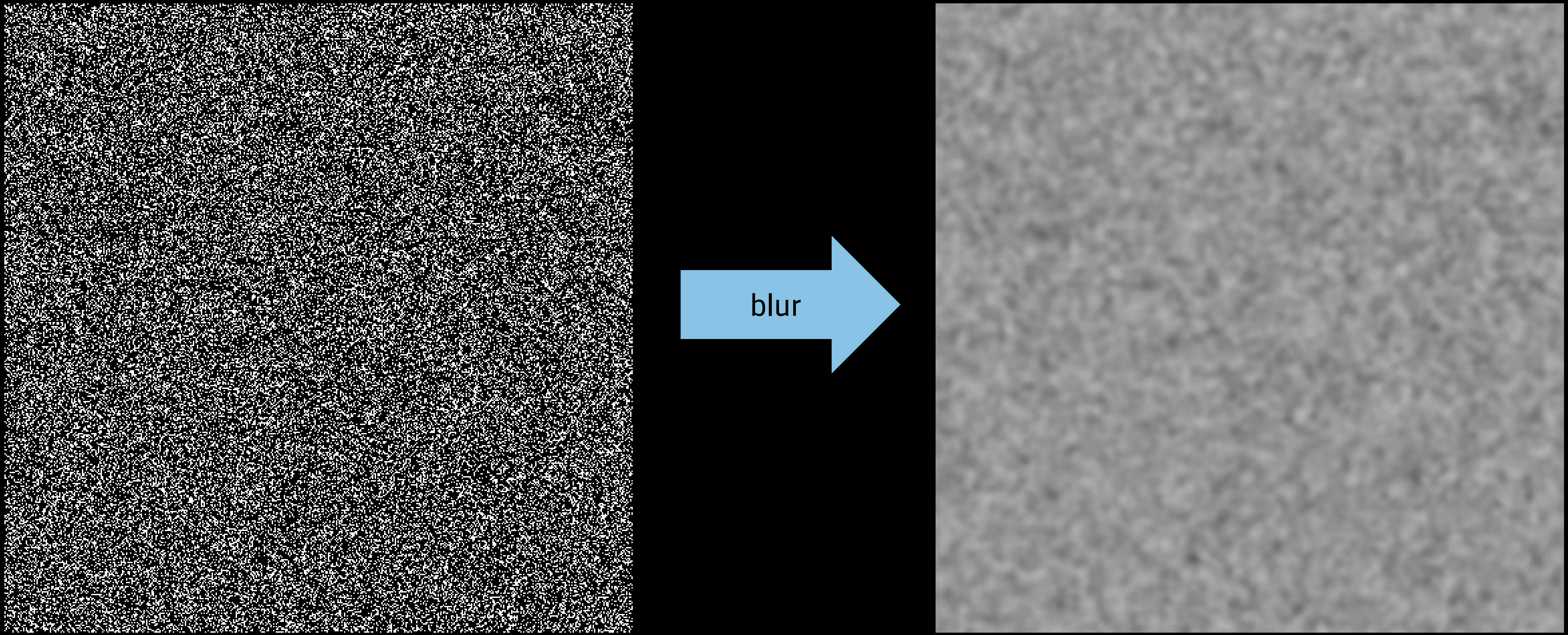
[1]NVIDIA [2]Aalto University [3]MIT CSAIL. Correspondence to: Jaakko Lehtinen <jlehtinen@nvidia.com>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

[1]https://github.com/NVlabs/noise2noise

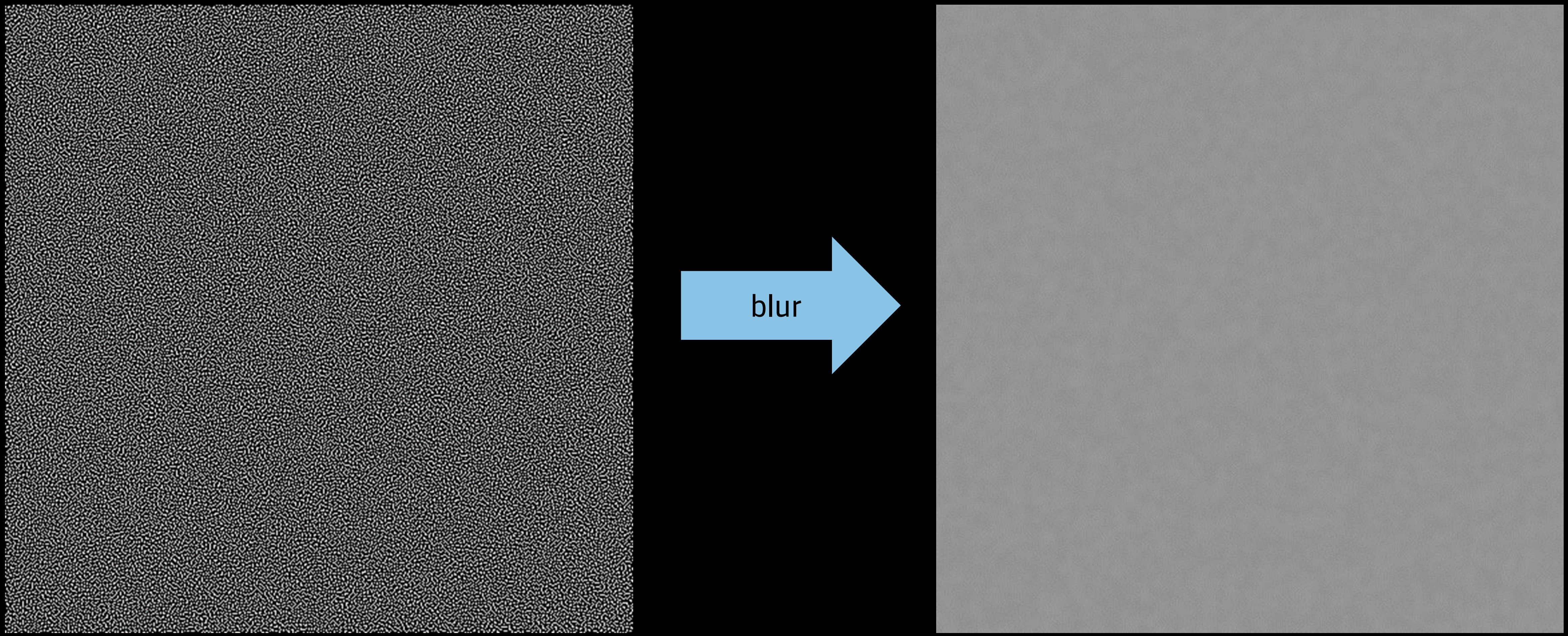
Sampling

White noise



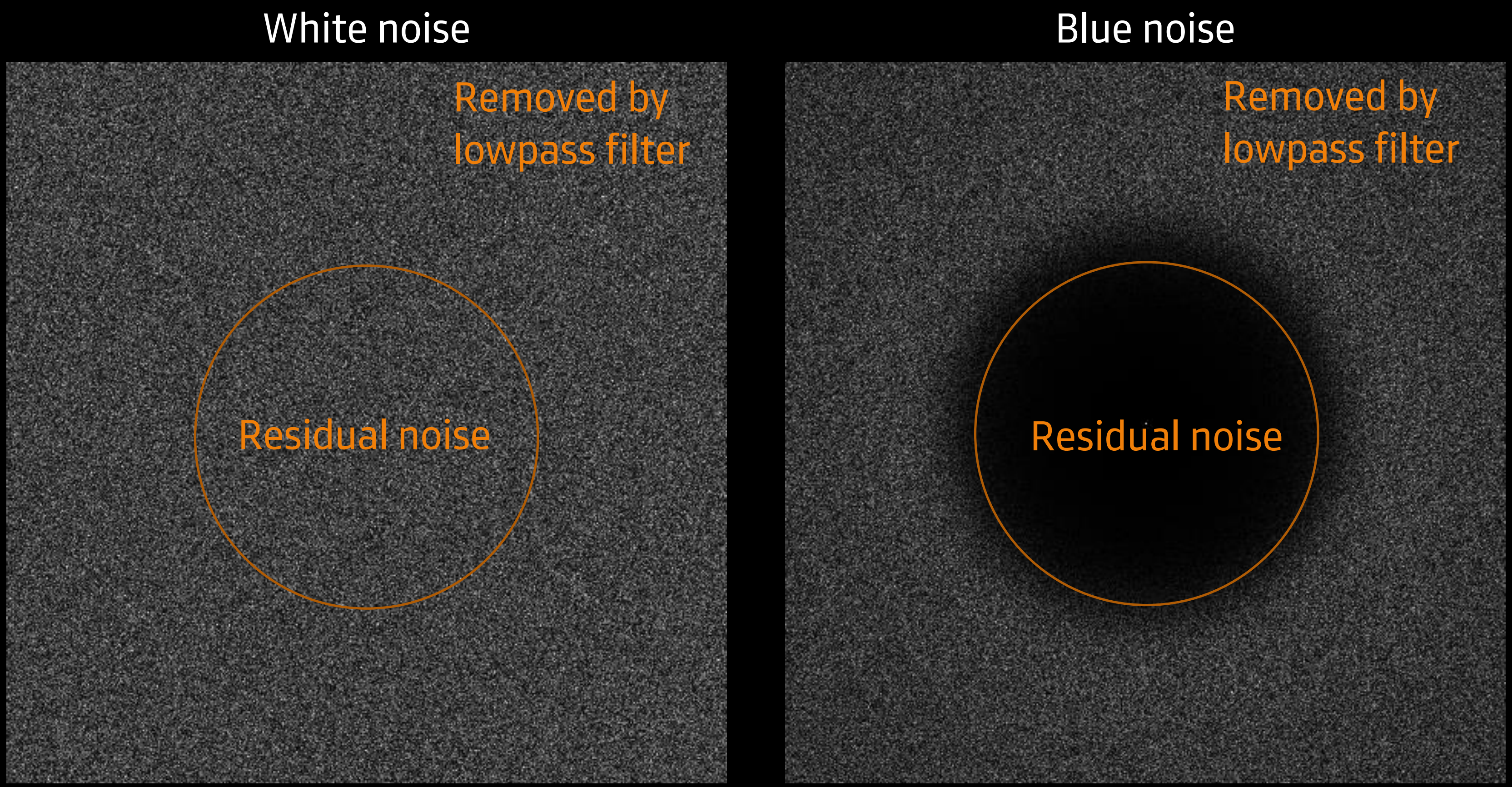
Source: http://momentsingraphics.de/?p=127

Blue noise



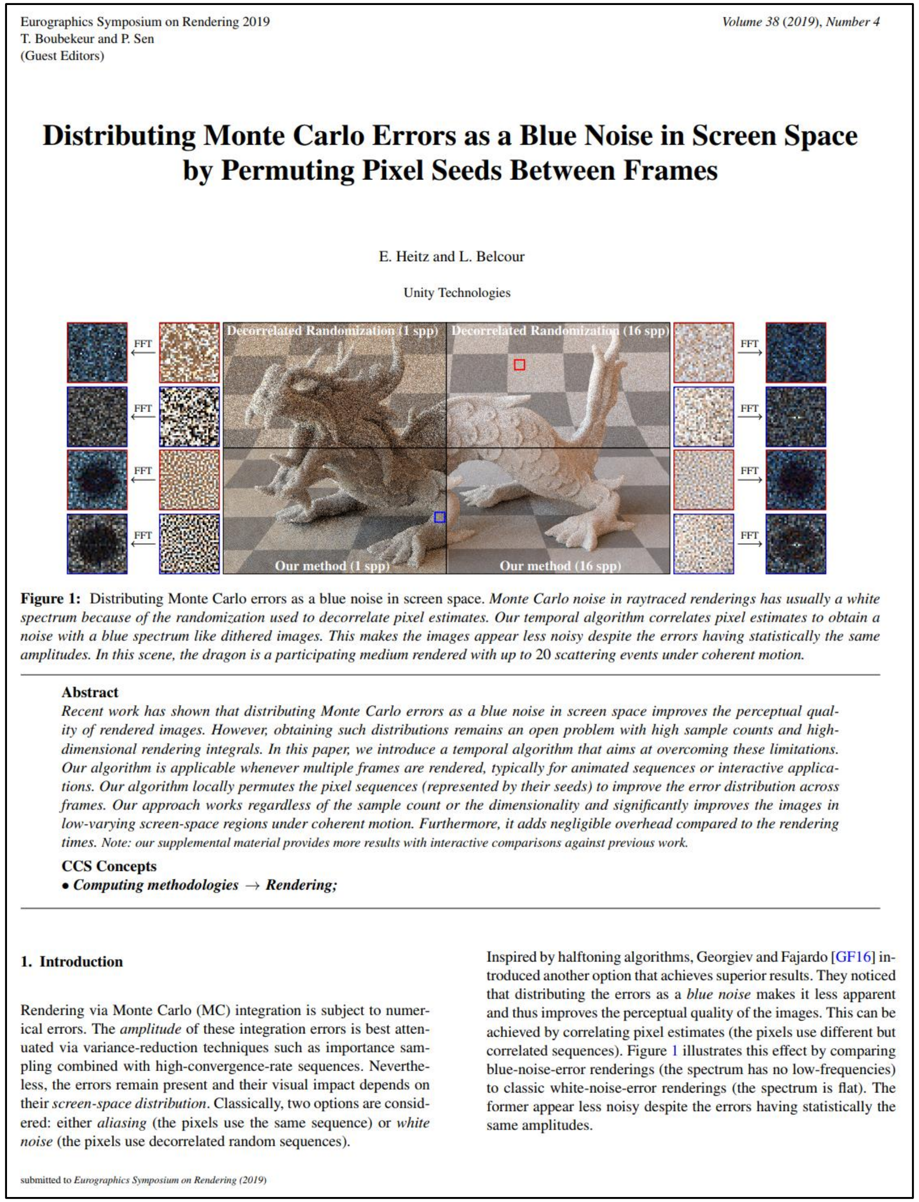
Source: http://momentsingraphics.de/?p=127

Magnitude of Fourier Transform



Source: http://momentsingraphics.de/?p=127

- Impose blue-noise characteristic on Monte Carlo Error


Eurographics Symposium on Rendering 2019
T. Boubekeur and P. Sen
(Guest Editors)

Volume 38 (2019), Number 4

Distributing Monte Carlo Errors as a Blue Noise in Screen Space by Permuting Pixel Seeds Between Frames

E. Heitz and L. Belcour

Unity Technologies

Decorrelated Randomization (1 spp) | Decorrelated Randomization (16 spp) | Our method (1 spp) | Our method (16 spp) | FFT

Figure 1: Distributing Monte Carlo errors as a blue noise in screen space. *Monte Carlo noise in raytraced renderings has usually a white spectrum because of the randomization used to decorrelate pixel estimates. Our temporal algorithm correlates pixel estimates to obtain a noise with a blue spectrum like dithered images. This makes the images appear less noisy despite the errors having statistically the same amplitudes. In this scene, the dragon is a participating medium rendered with up to 20 scattering events under coherent motion.*

Abstract
Recent work has shown that distributing Monte Carlo errors as a blue noise in screen space improves the perceptual quality of rendered images. However, obtaining such distributions remains an open problem with high sample counts and high-dimensional rendering integrals. In this paper, we introduce a temporal algorithm that aims at overcoming these limitations. Our algorithm is applicable whenever multiple frames are rendered, typically for animated sequences or interactive applications. Our algorithm locally permutes the pixel sequences (represented by their seeds) to improve the error distribution across frames. Our approach works regardless of the sample count or the dimensionality and significantly improves the images in low-varying screen-space regions under coherent motion. Furthermore, it adds negligible overhead compared to the rendering times. Note: our supplemental material provides more results with interactive comparisons against previous work.

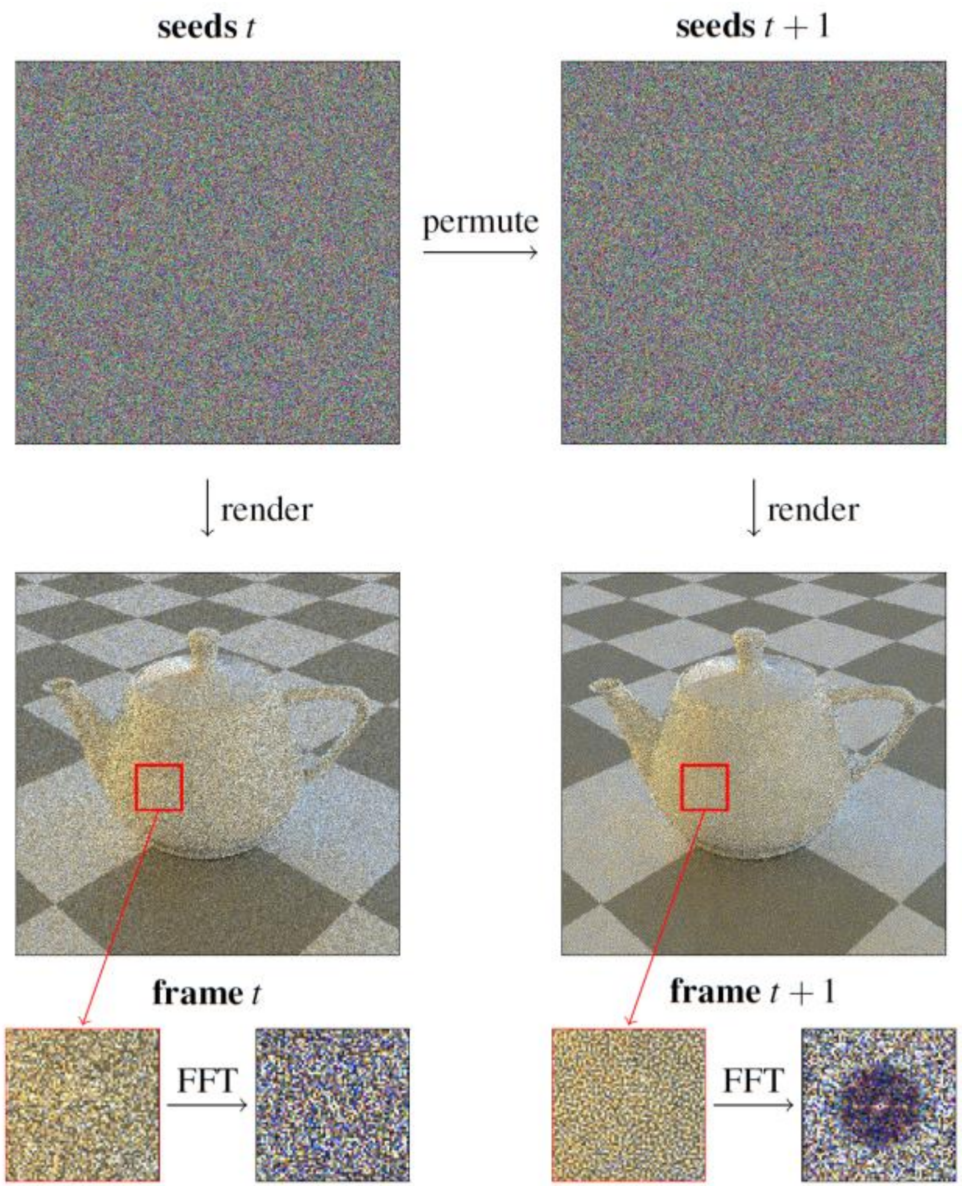
CCS Concepts
• *Computing methodologies* → *Rendering;*

1. Introduction

Rendering via Monte Carlo (MC) integration is subject to numerical errors. The *amplitude* of these integration errors is best attenuated via variance-reduction techniques such as importance sampling combined with high-convergence-rate sequences. Nevertheless, the errors remain present and their visual impact depends on their *screen-space distribution*. Classically, two options are considered: either *aliasing* (the pixels use the same sequence) or *white noise* (the pixels use decorrelated random sequences).

Inspired by halftoning algorithms, Georgiev and Fajardo [GF16] introduced another option that achieves superior results. They noticed that distributing the errors as a *blue noise* makes it less apparent and thus improves the perceptual quality of the images. This can be achieved by correlating pixel estimates (the pixels use different but correlated sequences). Figure 1 illustrates this effect by comparing blue-noise-error renderings (the spectrum has no low-frequencies) to classic white-noise-error renderings (the spectrum is flat). The former appear less noisy despite the errors having statistically the same amplitudes.

submitted to *Eurographics Symposium on Rendering (2019)*



[Heitz and Belcour 2019]

Not there yet!

Reconstruction

- Robustness (no history, high variance)
- Mirrors
- Overblurred materials

Viable path forward: CNN

- Temporal stability
- Performance

Path Tracing

- Robustness
- Offline techniques do not directly translate to real-time

Sampling

- Design patterns with reconstruction filters in mind