

An Efficient Denoising Algorithm for Global Illumination

Michael Mara
Stanford University

Benedikt Bitterli
Dartmouth College

Morgan McGuire
Williams College

Wojciech Jarosz
Dartmouth College

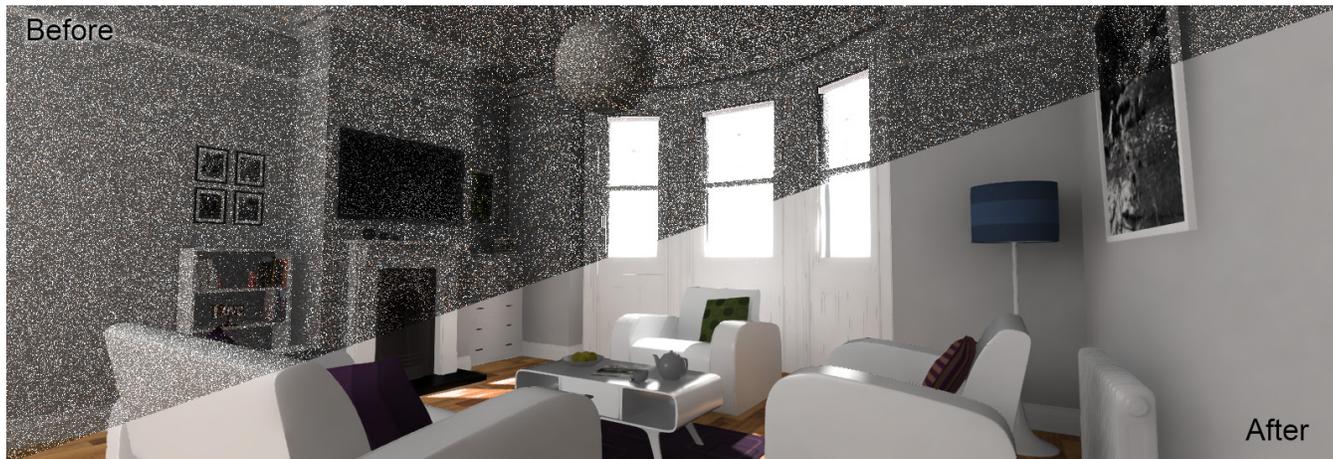


Figure 1: Real-time render with direct light plus two path-traced indirect samples per pixel, and our result denoised in 10 ms.

ABSTRACT

We propose a hybrid ray-tracing/rasterization strategy for real-time rendering enabled by a fast new denoising method. We factor global illumination into direct light at rasterized primary surfaces and two indirect lighting terms, each estimated with one path-traced sample per pixel. Our factorization enables efficient (biased) reconstruction by denoising light without blurring materials. We demonstrate denoising in under 10 ms per 1280×720 frame, compare results against the leading *offline* denoising methods, and include a supplement with source code, video, and data.

CCS CONCEPTS

• Computing methodologies → Rendering;

KEYWORDS

Denoising, reconstruction, real-time path tracing, illumination

ACM Reference format:

Michael Mara, Morgan McGuire, Benedikt Bitterli, and Wojciech Jarosz. 2017. An Efficient Denoising Algorithm for Global Illumination. In *Proceedings of HPG '17, Los Angeles, CA, USA, July 28-30, 2017*, 7 pages. <https://doi.org/10.1145/3105762.3105774>

HPG '17, July 28-30, 2017, Los Angeles, CA, USA

© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of HPG '17, July 28-30, 2017*, <https://doi.org/10.1145/3105762.3105774>.

1 INTRODUCTION

Efficient CPU and GPU parallel ray casting engines such as Embree and OptiX can now path trace global illumination in real time. However, as these systems can currently only afford a few rays per pixel, the results are too noisy for end-user applications and are therefore only suitable for preview in content creation tools.

While ray budgets are likely to increase with GPU and CPU performance, they are unlikely to increase ten-thousand-fold in the near future, as required for convergence in path tracing. Even film rendering does not actually trace to convergence, but instead post-processes images with denoising filters [Goddard 2014]. Previous high-quality denoising filters for path tracing are offline processes that expect hundreds of samples per pixel. In this paper, we describe a fast, new denoising filter based around a specific material (i.e., bidirectional reflectance distribution function; BRDF) factorization as a major step towards real-time path tracing with temporal stability and attractive, robust results. It runs two orders of magnitude faster than leading offline denoisers and yields smoother results without overblurring primary surfaces.

The primary contribution of this paper is a real-time spatio-temporal denoising framework, which is derived from a factored approximation of a material-based Monte Carlo integrator. The approximation itself is the core concept that enables a specific way of denoising images. An important key idea is the introduction of a separate filtering path in the historical buffer for temporal denoising. This makes it possible to reduce the bandwidth overhead of wide bilateral filters applied to lighting on matte surfaces without flicker or boiling.

Several principles and observations appropriate for real-time systems underly our method. The detail in reflected light is more

often due to material and direct illumination than to detail in the lighting environment, and blurring is less perceptible for indirect light than material. Under importance sampling, matte surfaces sample the entire hemisphere and thus are very noisy—but that same dispersion means the light can be aggressively denoised. Near-specular reflections exhibit little noise under material importance sampling because all incident rays are tightly clustered. Glossy lobes already band-limit reflections proportional to their roughness, so illumination from rougher surfaces can be denoised using wider filters. Temporal filtering at primary surfaces of reflections must be based on reprojection of the virtual position of reflected objects, not the position of the primary surfaces. Temporal filtering fails under disocclusion, but failure regions are predictable and can be corrected by wide spatial filters.

Our method separates direct light, indirect light, and material sampling at primary surfaces, applies to indirect light spatial and temporal kernels that grow in expected blurry and undersampled regions, and suppresses specific artifacts such as flickering high-lights.

2 RELATED WORK

Zwicker et al. [2015] and Gautron et al. [2014] survey the state of denoising for global illumination, which has historically been considered an offline problem since path tracing was itself an offline process until recently.

At a high level, most methods leverage extremely wide filters inspired by nonlocal means (NLM) [Rousselle et al. 2012], fitting a zeroth- or first-order gradient across a discontinuously segmented image. They determine the weighting of each sample to each term using buffers of geometric, material, and other metadata such as light visibility. We refer to the set of these and a motion vector buffer as a G-buffer. Other edge-avoiding (e.g., Dammertz et al. [2010]) filters and some nascent machine-learning approaches [Kalantari et al. 2015] also rely on wide filters with weights determined by the G-buffer.

Offline denoising strategies also resemble the smaller-extent cross bilateral filters [Eisemann and Durand 2004; Petschnigg et al. 2004] used for computational photography and real-time stochastic transparency. Those have spatially-varying Gaussian kernels with each tap weighted by a function of the G-buffer. Real-time stochastic transparency and antialiasing also rely heavily on temporal filtering and reprojection [Salvi 2016]. Bauszat et al. [2011] was the first to apply those to Monte Carlo rendering for interactive denoising.

We extend that line of real-time temporal and bilateral filter work to the domain of global illumination, derive a real-time sampling strategy for the path tracer based on BRDF frequency decomposition (following Zimmer et al. [2015]), and design a denoising filter chain. Our design is a robust and physically-based extension of ad hoc denoising filters we previously shipped in the Unity 5 game engine (2015) and applied in research papers (including Mara et al. [2016] and McGuire et al. [2017]). The new method is slower, but more robust and physically-based than those earlier methods.

Since our goal is *real-time* path traced indirect illumination, we designed our filter kernels to eliminate visually-undesirable artifacts with high performance; see related literature for matte [Durand et al. 2005; Kontkanen et al. 2004; Soler et al. 2009] and

glossy [Tokuyoshi 2015] reconstruction filters designed instead to minimize bias.

Schied et al. [2017] developed a real-time denoising method that operates on fully-path traced images with one path per pixel. Their research was independent and contemporary with ours and we look forward to a comparison as future work.

3 ALGORITHM

3.1 Factored BRDF

We assume the common division of the BRDF f into layered “matte” and “glossy” terms combined by Fresnel coefficients:

$$f(\hat{\omega}_i, \hat{\omega}_o) = m(\hat{\omega}_i, \hat{\omega}_o)(1 - F(\hat{\omega}_i, \hat{\omega}_o))^2 + g(\hat{\omega}_i, \hat{\omega}_o)F(\hat{\omega}_i, \hat{\omega}_o) \quad (1)$$

The glossy term $g()$ may include a specular reflection impulse. The matte term $m()$ must *vary slowly* with respect to $\hat{\omega}_i$ and $\hat{\omega}_o$, e.g., such as the Lambertian and Oren-Nayar models.

As an example, a typical real-time BRDF, Trowbridge-Reitz (a.k.a. “GGX”) microfacet plus Lambertian and Schlick’s Fresnel approximation would be represented in this factorization as:

$$g(\hat{\omega}_i, \hat{\omega}_o) = \frac{G(\hat{\omega}_i, \hat{\omega}_h)G(\hat{\omega}_o, \hat{\omega}_h)D(\hat{\omega}_h)}{4\pi |\hat{n} \cdot \hat{\omega}_o| |\hat{n} \cdot \hat{\omega}_i|}, \quad m(\hat{\omega}_i, \hat{\omega}_o) = \frac{\rho_L}{\pi} \quad (2)$$

$$\text{where} \quad \hat{\omega}_h = (\hat{\omega}_i + \hat{\omega}_o) / \|\hat{\omega}_i + \hat{\omega}_o\|, \quad (3)$$

$$F(\hat{\omega}_i, \hat{\omega}_o) = F_0 + (1 - F_0)(1 - \max(0, \hat{\omega}_h \cdot \hat{\omega}_i))^5, \quad (4)$$

F_0 is the Fresnel coefficient at normal incidence, G is the Smith geometry term, and D is the distribution of normals.

3.2 Factored Monte Carlo Integration

The standard material (vs. light- or multiple-) importance-sampling Monte Carlo integrator for the outgoing radiance L_o at a point X on a surface with primary visibility due to indirect light L_i is:

$$L_o(X, \hat{\omega}_o) = \frac{1}{2N} \sum_i^{2N} L_i(X, \hat{\omega}_i) \frac{f(\hat{\omega}_i, \hat{\omega}_o) |\hat{n} \cdot \hat{\omega}_i|}{p(\hat{\omega}_i)}, \quad (5)$$

where each $\hat{\omega}_i$ of $2N$ incident directions is independently sampled from distribution $p(\hat{\omega})$, and L_i is computed by path tracing.

We can choose any distribution for p that is nonzero where f is nonzero. The integrator will importance sample optimally with respect to the material when $p(\hat{\omega}) \propto f(\hat{\omega}, \hat{\omega}_o) |\hat{n} \cdot \hat{\omega}|$. We can also choose different sampling distributions for the different BRDF terms. For the matte estimator, we choose

$$p_m(\hat{\omega}) = \max(\hat{\omega} \cdot \hat{n}, 0) / \pi, \quad (6)$$

to cancel the numerator. This is nearly optimal importance sampling when $m()$ is nearly constant over the hemisphere. For the glossy term, choose each incident direction $\hat{\omega}_j$ from some distribution p_g that is close to $g()$ but efficient to sample, such as a power-cosine. (We use index j for glossy to make clear that these directions are sampled independently of the diffuse ones indexed by i .) The net estimator is:

$$L_o(X, \hat{\omega}_o) = \frac{\pi}{N} \sum_i^N L_i(X, \hat{\omega}_i) (1 - F(\hat{\omega}_i, \hat{\omega}_o))^2 m(\hat{\omega}_i, \hat{\omega}_o) \quad (7)$$

$$+ \frac{1}{N} \sum_j^N L_i(X, \hat{\omega}_j) \frac{F(\hat{\omega}_j, \hat{\omega}_o) g(\hat{\omega}_j, \hat{\omega}_o) |\hat{n} \cdot \hat{\omega}_j|}{p_g(\hat{\omega}_j)}. \quad (8)$$

3.3 Approximation

Because F and m both vary slowly in incident and outgoing vectors, we can make the approximation of factoring them out of the integrator to separate light and material. To do so, we must choose a single representative incident vector $\hat{\omega}_i$ outside the summation. We select $\hat{\omega}_i = \hat{\omega}_s$, the specular mirror reflection of $\hat{\omega}_o$ (this is a common real-time Fresnel approximation).

This reduces the matte portion of the estimator (Eqn. 7) to:

$$\frac{(1 - F(\hat{\omega}_s, \hat{\omega}_o))^2 m(\hat{\omega}_s, \hat{\omega}_o) \pi}{N} \sum_i^N L_i(X, \hat{\omega}_i). \quad (9)$$

We make similar approximations for the glossy estimator. However, because we know that g may be very sensitive to the incident direction $\hat{\omega}_j$ (i.e., it potentially contains narrow lobes), we cannot bring g outside of the summation and must evaluate it for each sample. The glossy portion of the estimator (Eqn. 8) is:

$$\frac{F(\hat{\omega}_s, \hat{\omega}_o)}{N} \sum_j^N L_i(X, \hat{\omega}_j) \frac{g(\hat{\omega}_j, \hat{\omega}_o) |\hat{n} \cdot \hat{\omega}_j|}{p_g(\hat{\omega}_j)}. \quad (10)$$

We compute direct illumination at primary surfaces and then compute the indirect contribution with $N = 1$ by path tracing. However, before combining these terms, we apply a reconstruction filter to the indirect light that effectively allows a very large N .

3.4 Reconstruction

Figure 2 describes our filter chain. Not shown in the diagram to simplify the data paths, all bilateral filters also read the G-buffer to weight taps by the camera-space plane distance in normals and depth from the center sample.

The filter chain gathers illumination from spatio-temporally adjacent samples, effectively varying position X and the implicit time parameter within the integrator. This generally does not overblur the image for three reasons: the approximations from the previous section place the material detail features (matte reflectance m and the glossy magnitude and albedo F_0) outside of the summations; the glossy spatial kernel width is chosen proportional to material roughness; and we correct for motion via reverse reprojection [Nehab et al. 2007; Salvi 2016; Yang et al. 2009]. It does blur out caustics, which we considered too hard to reconstruct from sparse samples.

Matte filtering is on the left side of the diagram; it computes irradiance. The temporal stage (3) reverse-reprojects the position of a sample into the previous frame. It computes a confidence $c \in [0, 1]$ based on how closely the reprojected position matches the expected value by:

$$c = h \cdot \text{smoothstep}(d_0, d_1, |X_{\text{expected}} - X_{\text{actual}}|) \quad (11)$$

The temporal filter then clamps the historical radiance to the distribution of the new 3×3 neighborhood and blends in $h \cdot c$ of the value stored in a historical buffer. Across all scenes, we empirically chose $h = 0.98$, $d_0 = 0.05$ m, $d_1 = 0.07$ m: a smooth transition at 6 cm from 0 to 95% hysteresis. The result passes on to the next stage, but is also filtered (4) and blended back into the historical buffer with hysteresis h .

Irradiance estimated from a small number of samples is extremely noisy, but we also expect the underlying signal to vary slowly on flat surfaces, so it is a good candidate for reconstruction by a wide filter

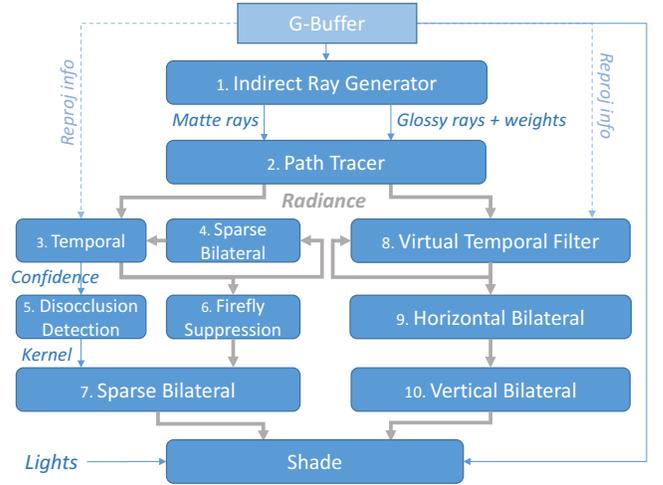


Figure 2: Our rendering system. Thick gray data paths represent radiance buffers, thin blue arrows are G-buffer data.

kernel. While small cross-bilateral filter kernels can be separated for efficiency into horizontal and vertical passes with only a small amount of directional bias near edges, large filters produce large streaks if separated. So, we apply a *sparse* cross-bilateral filter in stage 4. This uses only 50 taps in a disk with diameter equal to 1% of the screen width. The taps have temporally-varying positions given by Mara et al.’s [Mara et al. 2016] AO pattern, which is an efficient approximation of a blue noise distribution. Over several frames, this gives quality comparable to the dense filter but is about 2.5× faster.

We apply that sparse cross-bilateral filter only to data cycled back in to the *historical buffer*. This ordering is important—the cost of blurring is amortized over time by the temporal filter, but newly-sampled paths are never blurred out at this stage and noise from the sparse taps has one additional frame to settle. This process means that historical buffer is also repeatedly spatially filtered; since the kernel underlying the bilateral weighting is Gaussian, it converges to an ideal much wider Gaussian in flat regions such as ceilings. Combined with the sparsity of the filter, that is similar to what non-local means filtering accomplishes but with much lower bandwidth and computation per frame.

Stage (6) is a 3×3 median filter. This is to remove transient, single-sample bright “firefly” values from observation. We placed it outside the history feedback loop to ensure those samples still affect the running average but do not appear on screen; otherwise the images would be too dark.

The final sparse bilateral filter (7) removes low-frequency noise and fills temporal disocclusions. That filter’s radius r is determined by the disocclusion detection stage (5), which computes $r = (1 - \min_{2R \times 2R} \text{box}(c)) \times R$ by two simple 1D min passes. We empirically chose maximum radius $R = \text{screen width}/20$. As this is the last place to fill disocclusion holes, we expand the radius in (usually very small) regions where temporal confidence c is low. Our results were rendered with a multiplier of six where $c < 1/2$, but these values were ad hoc and a more principled tuning is merited.

Table 1: Run-time in milliseconds for real-time filter pipeline stages at 1280×720.

Stage	Time	Stage	Time
3. Matte Temporal	0.18	7. Matte Bilateral	3.36
4. Matte Pre-Bilateral	3.32	8. Glossy Temporal	0.23
5. Disocclusion Detection	0.76	9. Glossy H. Bilateral	0.34
6. Firefly Suppression	0.55	10. Glossy V. Bilateral	0.43

The glossy filtering on the right side of Figure 2 applies a temporal filter (8) to the *virtual* positions of reflected objects [Zimmer et al. 2015], avoiding the ghosting of previous real-time reflection reprojection [Salvi 2016]. It applies the same color clamping and confidence measures as for (3). The output is then cross-bilaterally filtered. After some experimentation, we chose for the underlying Gaussian kernel a standard deviation in pixels of $\sigma = 7\text{roughness}^{0.1} + \epsilon$. This is independent of resolution and distance to the surface because having an optimally sharp filter for the projection of the BRDF’s impulse response would leave too few taps in the distance, producing flicker.

This filter (9-10) is optimized for performance differently from the matte spatial ones. Despite our reflected reprojection, temporal filtering is less effective for glossy when objects are in motion. A sparse cross-bilateral filter such as the ones employed on the matte side of the process would leave visible sparse tap artifacts (appearing like dithering patterns) for glossy terms when reprojection fails, so we instead optimize the glossy spatial filter by separating it into two 1D passes.

Note that there is no firefly suppression on the glossy side. Fireflies in path tracing generally arise from caustics (which only appear on matte surfaces) and paths that were sampled with low probability and then greatly weighted upwards by importance sampling to compensate (which we limit within the path tracer, as is standard production practice). The remaining bright spots in glossy reflections are often genuine glossy highlights, i.e., reflections of small emissive objects, and should be preserved.

4 RESULTS

We evaluated our algorithm on several scenes from the literature. Most were previously rigged with explicit area light sources for “windows.” To make more realistic test cases for dynamic, real-time rendering, we remodeled the windows as holes and inserted a sun and bright sky. This is a harder scenario to denoise because there is minimal direct illumination; most light is due to indirect paths that happen to eventually pass through a window. Our implementation used the open source path tracer, BSDF, and deferred shader from the G3D Innovation Engine (<https://casual-effects.com/g3d>) in their default configurations plus the code from our supplement.

Unless otherwise noted, all results are at 1280×720 resolution with three-bounce illumination via one primary sample and two indirect paths per pixel, and the denoising takes about 9 ms on a NVIDIA Titan X GPU. Performance varies by $\pm 1\text{ms}$ with motion and viewpoint because the filter widths change in response. Table 1 gives a representative breakdown by stage, using the Horse Room scene under fast camera motion. Ray generation time and the incremental cost to deferred shading are negligible.

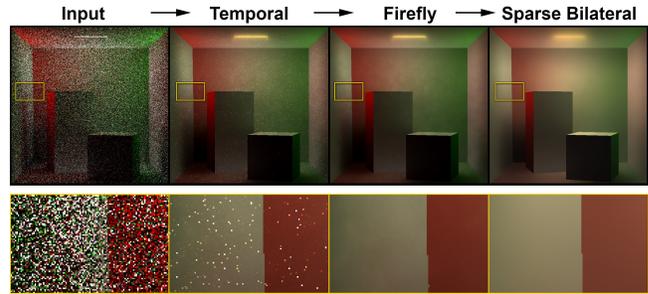
**Figure 3: Matte filter stage outputs, with details underneath.****Figure 4: Indirect illumination input and output terms of the denoising process for the Livingroom scene in Figure 1.**

Figure 3 shows the output of each matte filter stage for the Cornell Box. The initial spatio-temporal loop (3+4) removes significant amounts of high-frequency noise. Firefly suppression (6) removes flickering samples; while median denoising is not itself novel, placing it here ensures that very bright and dark samples are correctly incorporated into the historical average without also appearing immediately on the screen and flickering. The final sparse bilateral stage (7) removes low-frequency noise.

Figure 4 shows the indirect illumination input and output for Figure 1, with detail views zoomed in the top and bottom rows.

Figure 5 shows the disocclusion detection stage (5) expanding the matte sparse bilateral kernel in areas where reprojection fails, avoiding noise and ghosting (a-c), but slightly overblurring the mirror (d). The motion here is $10\times$ faster than in our video.

Figure 6 shows six scenes before and after real-time denoising with a mixture of realistic materials and lighting configurations. The brightest pixels in the left column are out of gamut, so those image appear darker even though the average energy in each neighborhood is the same before and after denoising. The ceilings are

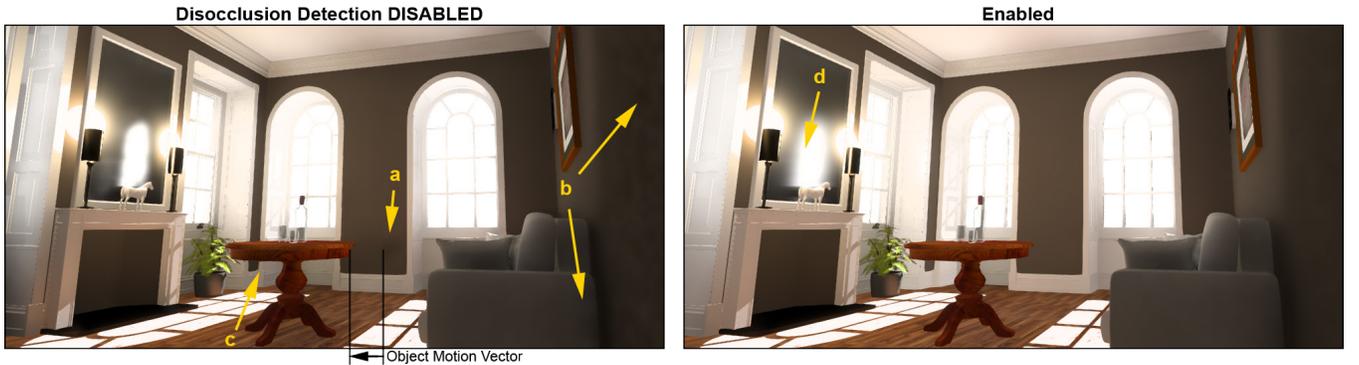


Figure 5: Horse room with rapid camera motion. Left) Noise and ghosting *without* disocclusion detection. Right) Improved.

the most difficult areas in these scenes. They are matte and have relatively uniform indirect light, no direct light, and no texture. This means that the samples are extremely noisy and any errors are highly perceptible. These areas necessitated our wide matte filter kernels. As future work, performance could be increased by applying much narrower filters in other regions where texture or direct illumination that masks the perception of indirect light noise.

Figure 7 compares the quality of the new real-time method against leading offline methods: Nonlocal means (NLM) [Rousselle et al. 2012], first-order weighted linear regression (WRL) [Moon et al. 2014], and nonlinearly weighted first-order regression (NFOR) [Bitterli et al. 2016]. We reimplemented NLM and used the original authors' implementations for WRL and NFOR.

Because the offline methods were developed to work with multiple samples per pixel, we rendered these results with sixteen samples per pixel. For our method, we simply ran at 4×4 higher resolution and then box filtered each pixel. Our method integrates historical data, so we ran it for 100 frames. We did not do this for the offline methods because they have no designed path for incorporating such data. The figure and insets are at full resolution to support zooming in the electronic version of this paper. Our supplement also contains both 8-bit tone-mapped PNG images and the original EXR radiance files for these results. Run times for processing these high-resolution images are given in Table 2.

We suggest NFOR and our new method produce the best results of the methods evaluated. Comparing just those two, our weakest result is the overblurred statue face on Glossy Sponza (b) and our strongest results are preserved material details in the Livingroom (e) and indirect reflection detail in the Horse Room (c). Across all experiments, our filters perform much better than previous work on the walls and ceilings (see supplement), which is consistent with residual low-frequency blotches for those methods in the literature.

Our results are much faster and generally smoother than the previous methods, without losing detail in mirror reflections or materials. This is possible through simultaneous co-design of the renderer, indirect sampler, and filter.

However, the offline methods were designed to denoise primary visibility and direct illumination as well and to run with hundreds of samples per pixel. They are also able to denoise refraction, which our method cannot. Transmissive objects in our results are rendered with order-independent transparency.

Table 2: Run times in seconds for offline tracing at $1280 \times 720 \times 16$.

Scene	NLM	WLR	NFOR	Ours
(a) Cornell Box	55	54	110	0.16
(b) Glossy Sponza	63	54	138	0.28
(c) Horse Room	60	52	129	0.31
(d) Bathroom	56	50	126	0.32
(e) Livingroom	58	50	119	0.31

In a scene with dynamic objects, the reprojection tests in the temporal filters fail to eliminate stale illumination on surfaces that are themselves unchanged in the camera view, but which should now have different lighting conditions. However, the color distribution clamping limits the error in this situation. There is some latency for diffuse indirect illumination, but it occurs primarily on matte surfaces. Sharp reflections, such as in mirrors, experience minimal latency because they require little filtering in the first place. See our video for examples.

5 DISCUSSION

5.1 Limitations

Where insufficient information is present due to high variance or disocclusion, our design and tuning favor blurring to either high or low frequency noise in space or time. For example, it overblurs caustics and areas where shading is dominated by indirect visibility instead of material or geometry. Distant glossy reflections are also blurrier than close ones because the glossy spatial filter's width is independent of distance to the surface, which we found preferable to distant flicker. We render refraction solely using forward rasterization and a displacement shader.

5.2 Future Work

We anticipate a future in which ray tracing improvements yield a budget of six to ten rays per pixel per real-time frame. Then, fast denoising will be needed to make those results spatially smooth and stable for entertainment applications.

Our algorithm is orders of magnitude faster and expects orders of magnitude fewer rays than previous methods; it enables real-time global illumination on a high-end machine today. However, 8-10 ms

is still expensive at 60 fps. Our method is bandwidth-limited, so we hypothesize that the best route for further enhancing performance is tuning for lower precision (e.g., R11G11B10F is twice as fast as our RGB16F format, because of alignment constraints), and stratified, quasi-Monte Carlo sampling for both the sparse filters and the path tracer to speed convergence.

Acknowledgements

We thank anonymous Reviewer 1 for the description of the paper’s contributions that appears in the introduction; Carsten Benthin and Ingo Wald at Intel and the NVIDIA OptiX team for their technical and financial support of this research; and the artists Guillermo M. Leal Llaguno, Marko Dabrovic, Nacimus Ait Cherif, “Jay Artist”, Nic Hull, and Guedis Cardenas for their scenes.

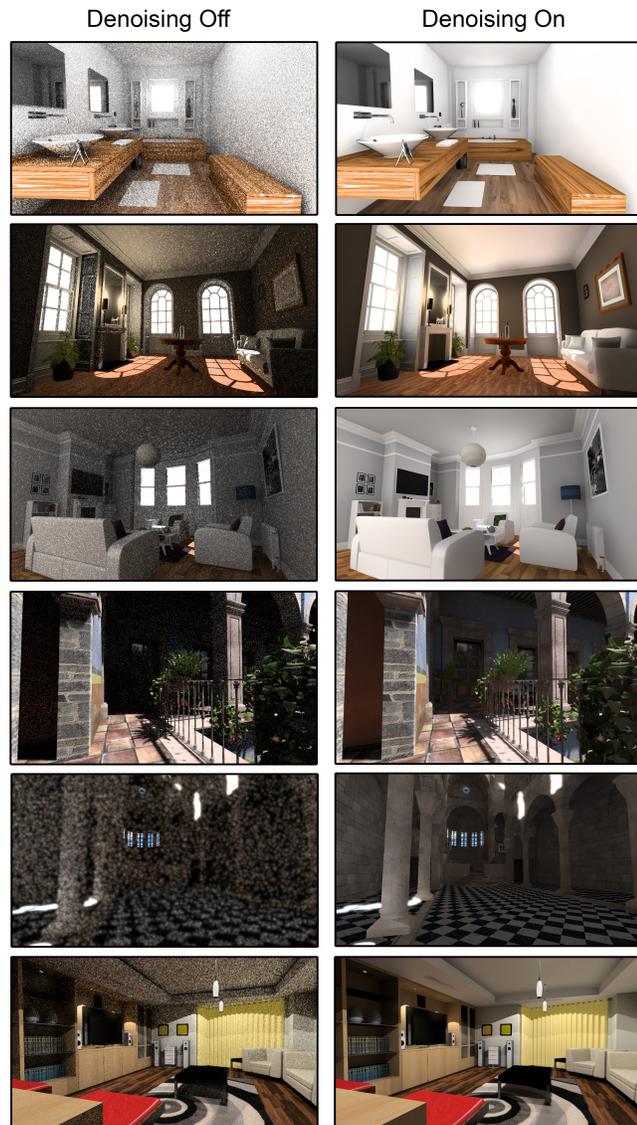


Figure 6: Real-time results with the raw path-traced samples, and after denoising.

REFERENCES

Pablo Bauszat, Martin Eisemann, Marcus Magnor, and Naveed Ahmed. 2011. Guided Image Filtering for Interactive High-quality Global Illumination. *Proc. of EGSR* 30, 4 (Jun 2011), 1361–1368.

B. Bitterli, F. Rousselle, B. Moon, J. A. Iglesias-Gutián, D. Adler, K. Mitchell, W. Jarosz, and J. Novák. 2016. Nonlinearly Weighted First-order Regression for Denoising Monte Carlo Renderings. In *Proc. of EGSR*, Vol. 35.

H. Dammertz, D. Sewtz, J. Hanika, and H. P. A. Lensch. 2010. Edge-avoiding À-Trous Wavelet Transform for Fast Global Illumination Filtering. In *Proc. of HPG*, 67–75.

Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. 2005. A Frequency Analysis of Light Transport. *ACM Trans. Graph.* 24, 3 (July 2005), 1115–1126.

Elmar Eisemann and Frédo Durand. 2004. Flash Photography Enhancement via Intrinsic Relighting. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 673–678.

Pascal Gautron, Marc Droske, Carsten Wächter, Lutz Kettner, Alexander Keller, Nikolaus Binder, and Ken Dahm. 2014. Path Space Similarity Determined by Fourier Histogram Descriptors. In *SIGGRAPH Talks*. ACM, 39:1–39:1.

L. Goddard. 2014. Silencing the Noise on Elysium. In *SIGGRAPH Talks*.

Nima Khademi Kalantari, Steve Bako, and Pradeep Sen. 2015. A Machine Learning Approach for Filtering Monte Carlo Noise. *ACM Trans. Graph.* 34, 4 (July 2015).

Janne Kontkanen, Jussi Räsänen, and Alexander Keller. 2004. Irradiance Filtering for Monte Carlo Ray Tracing. In *Proc. of MC2QMC*. Springer, 259–272.

M. Mara, M. McGuire, D. Nowrouzezahrai, and D. Luebke. 2016. Deep G-Buffers for Stable Global Illumination Approximation. In *Proc. of HPG*, 11.

M. McGuire, M. Mara, D. Nowrouzezahrai, and D. Luebke. 2017. Real-Time Global Illumination using Precomputed Light Field Probes. In *Proc. of I3D*, 11.

B. Moon, N. Carr, and S. Yoon. 2014. Adaptive Rendering Based on Weighted Local Regression. *ACM Trans. Graph.* 33, 5, Article 170 (Sept. 2014), 14 pages.

Diego Nehab, Pedro V. Sander, Jason Lawrence, Natalya Tatarchuk, and John R. Isidor. 2007. Accelerating Real-time Shading with Reverse Reprojection Caching. In *Graphics Hardware*. 25–35.

Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. 2004. Digital Photography with Flash and No-flash Image Pairs. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 664–672.

F. Rousselle, C. Knaus, and M. Zwicker. 2012. Adaptive Rendering with Non-local Means Filtering. *ACM Trans. Graph.* 31, 6 (Nov. 2012), 195:1–195:11.

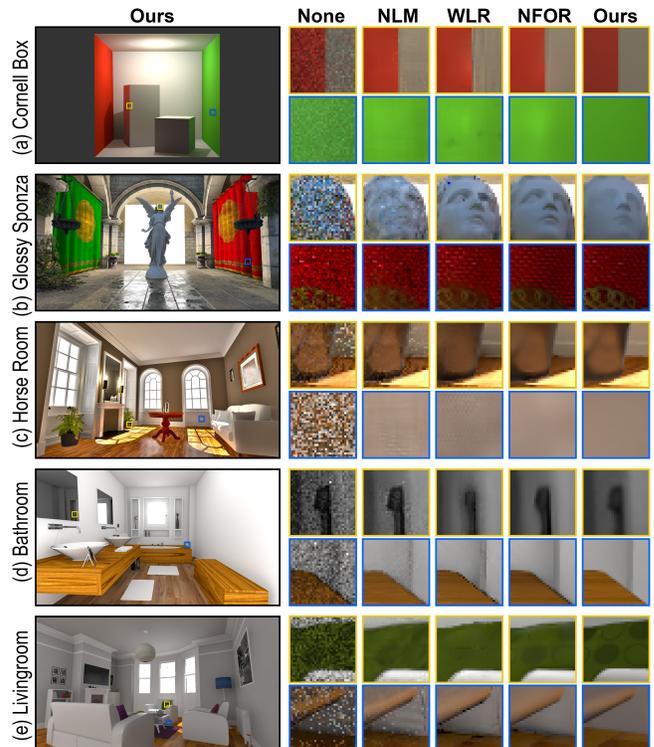


Figure 7: Comparison against leading offline methods at 1280x720 with 16 samples per pixel.

- Marco Salvi. 2016. An Excursion in Temporal Supersampling. (2 March 2016). <https://developer.nvidia.com/gdc-2016> Talk at GDC.
- Christoph Schied, Anton Kaplanyan, Anjul Patney, Chris Wyman, Chakravarty Reddy 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 *Proc. of HPG*. 12.
- Cyril Soler, Kartie Subr, Frédo Durand, Nicolas Holzschuch, and François Sillion. 2009. Fourier Depth of Field. *ACM Trans. Graph.* 28, 2 (May 2009), 18:1–18:12.
- Y. Tokuyoshi. 2015. Specular Lobe-Aware Filtering and Upsampling for Interactive Indirect Illumination. *Comp. Graph. Forum* 34, 6 (2015), 135–147.
- L. Yang, D. Nehab, P. V. Sander, P. Sitthi-Amorn, J. Lawrence, and H. Hoppe. 2009. Amortized Supersampling. In *SIGGRAPH Asia*. 135:1–135:12.
- Henning Zimmer, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. 2015. Path-space Motion Estimation and Decomposition for Robust Animation Filtering. In *Proc. of EGSR*. 131–142.
- M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon. 2015. Recent Advances in Adaptive Sampling and Reconstruction for Monte Carlo Rendering. *Comput. Graph. Forum* 34, 2 (May 2015), 667–681.