https://doi.org/10.2352/EI.2026.38.16.AVM-106

The Influence of Monte Carlo Denoisers on the Quality of Real-Time Optical System Simulations

***Jan Halama*[1] *and Alexander Braun*[1]**

[1] ***University of Applied Sciences Düsseldorf, Düsseldorf, Germany***

Abstract

Raytracing in combination with Monte Carlo simulation is an accurate method to simulate optical systems in virtual 3D scenes. Since Monte Carlo simulation relies on random sampling, many samples per pixel need to be computed for a noise-free image, resulting in high computational effort. Even the fastest ray tracers can only trace a few samples per pixel in real-time. A common solution in computer graphics is to compute the image with a few samples per pixel and apply a Monte Carlo denoiser to remove the noise. Since the denoiser alters the image, the question arises to what extent this influences the quality of the simulation. Utilizing "Simulating tests to test simulation", we measure the SFR curve of a simulation denoised with the NVIDIA OptiX Denoiser and compare it with a highly sampled baseline simulation. Although the image is altered by denoising, using denoised ray tracing simulations yields more realistic results for real-time rendering than a Gaussian blur, but there is a significant texture loss.

Introduction

Autonomous vehicles are equipped with a variety of sensors to perceive their surroundings for safe navigation. Radar or LiDAR sensors alone are insufficient. The recognition of traffic signs is only possible with cameras, whose images are processed by a computer vision algorithm. Detecting the traffic signs in the image is a safety-critical task and therefore needs to be validated. Due to the large number of possible situations, validating the computer vision algorithms is only possible through simulations. The performance of the camera lens has a huge impact on the detection rate of the computer vision algorithm. Therefore, the camera lens must be included in the simulation.

For the simulation to be reliable, the simulation must be physically correct. Ray tracing in combination with Monte Carlo integration has evolved as the go-to algorithm for accurate light transport simulation [12]. It can deliver high quality results, but the random sampling of the Monte Carlo integration leads to noise in the image. Figure 1 shows the *Sponza* scene rendered with only 15 samples per pixel. The noise is clearly visible. Since Monte Carlo integration samples different domains, the noise in the image is a combination of the sampling noise of these different domains (integrating over the area of the light, the area of the lens, the spectrum of the light). To remove the noise, calculating many samples per pixels is necessary, resulting in high computational effort. In computer graphics, it is often argued that noise from Monte Carlo integration is acceptable, since an image sensor also causes noise in the image. But in contrast to noise introduced by Monte Carlo integration, image sensor noise is caused by a physical process. Monte Carlo noise, on the other hand, is a flaw of the used mathematical method. For a realistic simulation we must compute a noise free image using Monte Carlo integration and apply a realistic sensor noise model to the image.

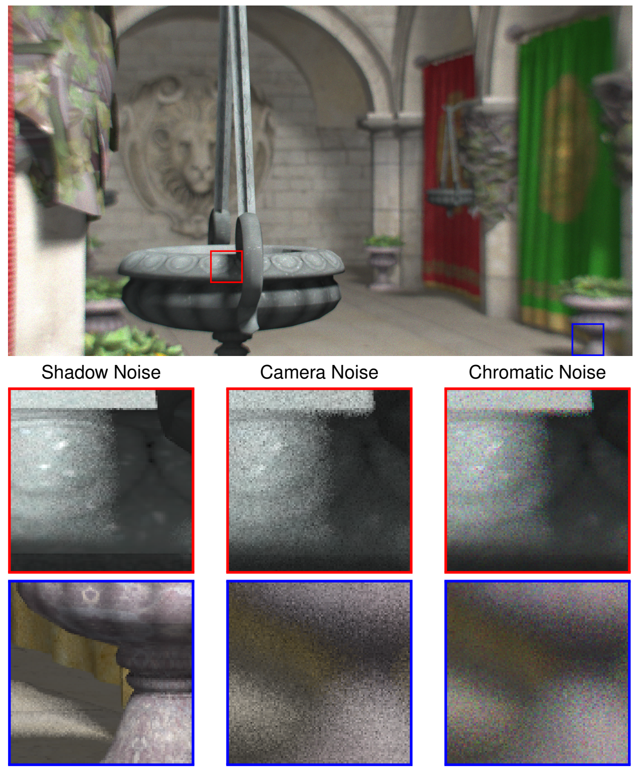

Figure 1: *Sponza* [14] scene rendered through the Canon Zoom Lens [17] with 15 samples per pixel. No denoiser was applied. The Monte Carlo Noise is clearly visible. Each integration domain (shadow, camera and light spectrum) contributes to the noise.

Real-time simulation is a hard requirement for a Hardware in the Loop (HiL) setup. Software in the Loop (SiL) does not need a real-time simulation but still benefits from a high-speed simulation due to the huge number of possible scenarios. The high computational effort for a noise free image makes it challenging for real-time simulations. Even the fastest ray tracers can compute only a small number of samples per pixel in real-time. This is due to two expensive phases for camera lens simulation in virtual 3d scenes: Ray generation, because the ray needs to be intersected and refracted with all lens surfaces, and light transport calculation. Different methods were developed to optimize the ray generation phase for real-time, either by caching the camera rays [8] or approximating the optical system by a polynomial [6, 7, 9, 20, 25]. To optimize the calculation of the light transport, one solution is to render the image with a few samples per pixel and apply a Monte Carlo denoiser to remove the noise.

State of the art Monte Carlo denoisers such as the NVIDIA OptiX Denoiser [2, 16] or Intel OpenImageDenoise [1] are based on neural networks and can deliver promising results from only a few samples per pixel. They are trained on combinations of noisy images and their corresponding ground truth rendered with many samples per pixel. Because Monte Carlo denoisers work on images from virtual scenes, it is possible to provide them with more information about the scene, such as albedo and surface normals. This yields better results and differentiates such Monte Carlo denoisers from denoisers created for real-world images, making them the go-to choice to denoise images rendered with Monte Carlo methods. Figure 2 shows a scene rendered using 15 samples per pixel with a denoiser applied and the different inputs to the denoiser. Despite the low sample count the denoiser can create an accurate noise-free reconstruction of the scene.

Since the Monte Carlo denoiser is a neural network, it is not possible to explain the way the image is altered. So far, denoisers are only evaluated numerically [2, 19, 22, 24]. Therefore, the question arises how this affects optical correctness of the camera lens simulation. To quantify the influence of the Monte Carlo denoiser, we follow *Simulating tests to test simulation* [15]. Existing metrics developed to characterize real cameras are applied to simulated images to quantify the simulation quality.

We contribute the evaluation of a Monte Carlo denoiser regarding the accuracy of camera lens simulation. Metrics developed to evaluate real cameras are used to quantify the simulation quality. Despite the lens blur accuracy not being influenced by the denoiser even for images rendered with low sample counts, the denoiser introduces significant texture loss.

Method

To quantify the delta introduced by the denoiser, we follow *Simulating tests to test simulation*, utilizing metrics developed to measure real cameras. We conduct measurements on denoised images rendered with a low sample count and noise-free reference images rendered using brute force and high sample counts. To assess how much the denoiser affects the realism of the simulation, we compare the ray tracing lens simulation with a Gaussian blur, which is easy to implement and therefore widely used, but it lacks any physical basis.

To quantify the accuracy of the lens blur, we measure the Spatial Frequency Response (SFR) using a slanted edge chart. The measurements are specified in ISO 12233 [5]. From experience we know that denoised images are a bit blurrier in in-focus areas than in the reference image. We quantify this texture loss using a dead leaves target and SFR measurements specified in ISO TS 19567-2 [11]. For slanted edge and dead leaves measurements, a test chart is rendered a virtual scene. For the slanted edge, we use ISO 12233 slanted edge patches, placed in image corners and center. For the lead leaves, we used the TE276 target by Image Engineering. The metrics are then calculated based on the image using the software iQ Analyzer X 1.11.1 by Image Engineering [10].

Figure 2: *Sponza* scene with 15 samples per pixel, denoised with OptiX Denoiser. Providing albedo and normal beside the RGB input lowers the denoiser error.

The result of the denoiser and the difference to the reference image is heavily influenced by the noise level of the input image. Since the noise level is directly related to the number of samples per pixel used, we render the test chart with different sample counts and compare the results to the brute force baseline.

The simulation environment used is the same as in our Ray-LUT Paper [8]. We use the OptiX Sample application by Wald and Parker [23] with a *Realistic Camera* [13] implementation based on Pharr, Jakob, and Humphreys [18]. To speed up the camera lens simulation for real-time performance, we apply the Ray-LUT and cache the camera rays, utilizing axis symmetry to reduce memory usage. We conduct our measurements on a NVIDIA RTX PRO 6000 Blackwell with 96gb of VRAM.

As a camera lens we use a complex Canon Zoom lens [17] with 34 surfaces. The Ray-LUT enables us to simulate such complex lenses in real-time. All simulations are rendered with a resolution of 1928x1088 pixels. For denoised simulations, we render 15 samples pixel, because this is the maximum amount of samples the A6000 can compute in real-time. For the brute force reference images we compute 4096 samples per pixel. To simulate chromatic aberrations, we sample three wavelengths for R, G and B and compute the wavelength-dependent index of refraction using the Sellmeier equation [21]. For the simulations with Gaussian blur, we render the test chart using a pinhole camera and apply an isoplanar blur using Pillow Python Package [3]. We use a blur radius of three pixels, because in the image corners the resulting SFR curve matches the curve of the Canon lens most.

Results

Figure 3 shows the simulated slanted edge patches and highlights two different image regions in different states. The patch in the image center and the bottom left patch are selected, because aberrations increase in the image periphery. Since the Canon lens

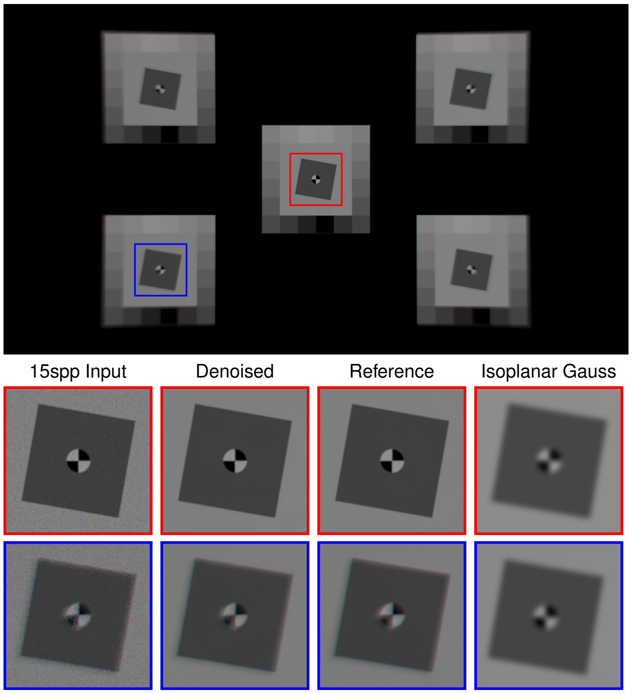

Figure 3: Slanted edge charts rendered with 15 samples per pixel and OptiX denoiser applied. Lens: Canon Zoom Lens [17], f/4.

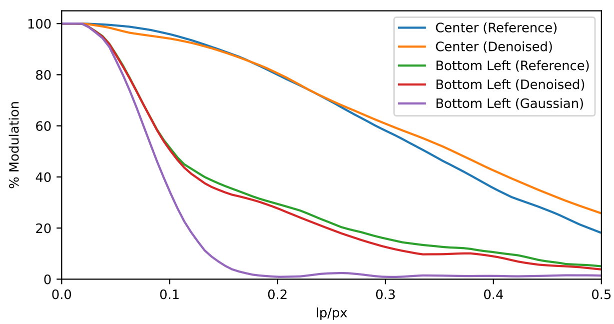

Figure 4: Slanted Edge SFR curves of raytraced lens simulations (reference and denoised) and Gaussian blur simulation, for different target locations.

is rotational symmetric, investigating one patch is sufficient. For both image regions, the denoiser input clearly show noise. Applying the denoiser leaves no noise remaining and visually matches the reference image quite well. Since the Gaussian blur is isoplanar, both regions yield the same result. The blur radius of three pixels matches the blur on the slanted edge.

For each patch the four slanted edges are measured using iQ-Analyzer and averaged, yielding one SFR curve per patch. Figure 4 shows the resulting curves for the two images regions, for denoised and reference. Since the applied Gaussian blur is isoplanar, only the curve for the bottom left patch is presented. The curves of the denoised simulation match the reference curves quite well. For the image center, the denoised curve is slightly sharper from around 0.2 line pairs per pixel. For the bottom left, the curve shape of the denoised and the reference curve are also similar. Until 0.1 line pairs per pixel, the curves overlap, for higher frequencies the denoised simulation is slightly more blurry. In contrast, the curve of the Gaussian blur is completely different and already drops way

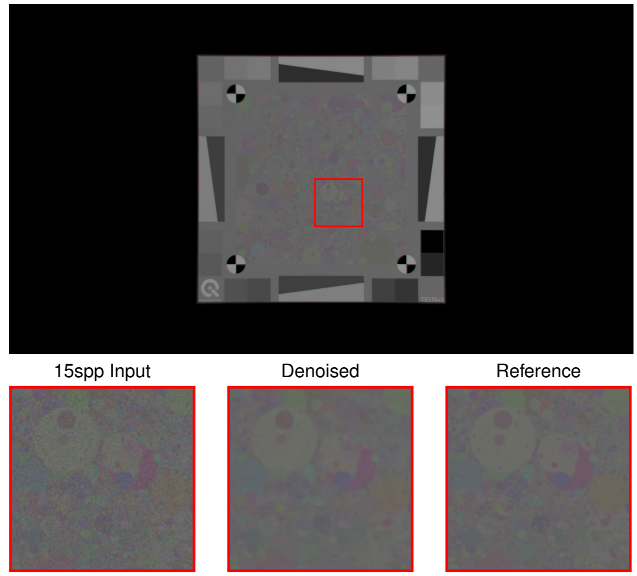

Figure 5: Dead leaves chart rendered with 15 samples per pixel and OptiX Denoiser applied. Lens: Canon Zoom Lens [17], f/4.

below the reference curve around 0.075 line pairs per pixel. The difference to the reference curve is much larger than for the denoised simulation.

We compare the SFR curve for different SPPs with the reference curve. Often scalar values like SFR10 or SFR50 are used to compare the sharpness of a lens. Because a single value cannot capture the entire curve, we compare the area of the curve instead. Figure 7 shows a plot of the relative SFR curve area over different amounts of samples per pixel. Overall, the results don't differ between the two slanted edge patches. For just one sample per pixel, the area of the curve already matches the area of the reference curve by 60%. The Gaussian blur only matches the reference curve area by 57.61%. With increasing samples per pixel, the accuracy increases, but starts to converge between 16 and 32 samples per pixel. There is no significant improvement by using up to 128 samples per pixel.

Figure 5 shows the simulated dead leaves charts and highlights one image region in different states. The denoiser input clearly shows noise, which is removed by the denoiser. Texture loss is already visible compared to the reference simulation. The SFR curve of the dead leaves chart quantifies this significant texture loss. In contrast to the slanted edge chart the dead leaves chart is way more sensitive to the amount of samples per pixel. Figure 7 shows a plot of the relative SFR curve area. The curve area compared to the reference curve does grow with higher sample counts and only starts to slowly converge around 64 samples per pixel. Even for 128 samples per pixel, which is not suitable for real-time simulations in any case, the curve area does not match as good as for the slanted edge chart.

Discussion

The influence of the denoiser on the slanted edge SFR is not strong. With 15 samples per pixel as input to the denoiser, the SFR curve matches the reference simulation closely. While there is an increase in accuracy for higher sample counts, it is not significant. 15 samples per pixel already seems like the sweet spot in terms of

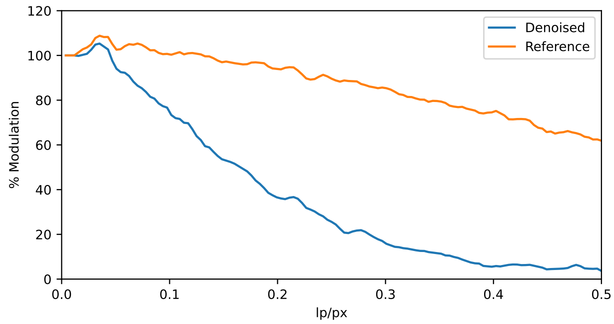

Figure 6: Dead leave SFR curve for denoised simulation with 15 samples per pixel and reference simulation.

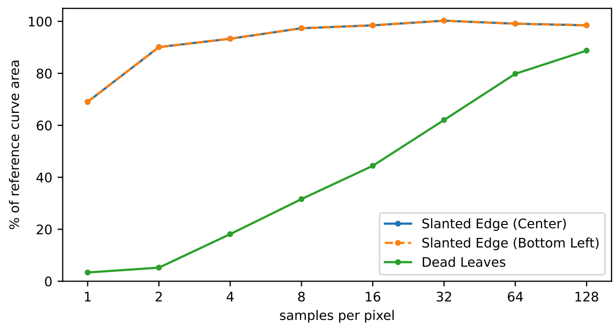

Figure 7: Relative area of SFR of denoised images compared to SFR curve area of reference simulation, over different amounts of samples per pixel.

computational effort and accuracy. Even with just one sample per pixel, the SFR curve of the denoised image matches the reference more closely than a Gaussian Blur. For the dead leaves target, the texture loss is significant. In contrast to the slanted edge SFR, more samples per pixel increase the accuracy. But even with 128 samples per pixel, the dead leaves SFR does not match the reference curve as closely as it does for the slanted edge SFR with only 15 samples per pixel.

The main goal of simulations is to study computer vision algorithms in different situations. So far, it is too early to state that using a denoiser has no influence on the simulation. One would have to investigate how the denoiser affects computer vision algorithms, i.e., whether the algorithms react differently to a denoised simulation and to a noise-free brute force simulation.

Such an assessment is particularly important for real-time simulations. While graphics hardware is getting faster every year, this increase is used to add more complexity to simulations. In the foreseeable future, real-time simulations will only be able to handle a few samples per pixel [2], using a denoiser will therefore remain unavoidable. For offline simulations, a denoiser is not strictly necessary. However, even here, a denoiser offers the potential to save computing time, for example, to generate large datasets. In the film industry, the use of denoisers for offline rendering is already widespread [4].

Conclusion and Future Work

We investigated the influence of the NVIDIA OptiX Monte Carlo denoiser on the quality of a camera lens simulation using *Simulating tests to test simulation*. Test charts developed for real cameras were rendered with a low number of samples per pixel and denoised using the NVIDIA OptiX denoiser. We compared those renderings to brute force simulations of the same chart with a high number of samples per pixel. To quantify the influence of the denoiser on the lens blur, we investigated the spatial frequency response using a slanted edge target. As the denoiser creates a slightly blurry image, we utilized a dead leaves target to quantify the texture loss introduced by the denoiser. For the slanted edge target, the denoised images show nearly the same SFR curve as the reference simulations. Using more samples per pixel does not improve the simulation quality significantly. The SFR curve remains realistic in any case and clearly differs from the SFR curve of a Gaussian blur, even for low sample counts. In contrast, the SFR curve derived from the dead leaves target shows a significant texture loss introduced by the denoiser. Increasing the number of samples per pixel reduces the texture loss but requires way more samples than for the slanted edge SFR.

So far, we only investigated the NVIDIA OptiX denoiser, since it is a popular choice for real-time simulations. Other denoisers like OpenImageDenoise [1] could be explored and compared to the NVIDIA OptiX denoiser.

Using a denoiser seems like a promising solution to achieve real-time simulations or reduce the computational effort required to generate large datasets using offline rendering. The purpose of many simulations in the context of autonomous driving is to study the detection rate of computer vision algorithms. A denoiser can only be implemented into those kinds of simulations if it does not change the results, i.e. does not affect the computer vision algorithm. So far, we have not touched this area of research.

Acknowledgments

We would like to thank the NVIDIA Corporation for the donation of the RTX PRO 6000 Blackwell GPU, which we used to conduct all experiments in this paper. We would like to thank Image Engineering GmbH & Co. KG, Kerpen-Horrem, Germany for the access to their software.

References

[1] Attila T. Áfra. *Intel® Open Image Denoise*. 2025. URL: `https://www.openimagedenoise.org`.

[2] Chakravarty R. Alla Chaitanya et al. "Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder". en. In: *ACM Transactions on Graphics* 36.4 (Aug. 2017), pp. 1–12. ISSN: 0730-0301, 1557-7368. DOI: `10.1145/3072959.3073601`. (Visited on 09/15/2025).

[3] Jeffrey A. Clark. *Pillow*. 2026. URL: `https://pillow.readthedocs.io/en/stable/` (visited on 03/16/2026).

[4] Henrik Dahlberg, David Adler, and Jeremy Newlin. "Machine-learning denoising in feature film production". en. In: *ACM SIGGRAPH 2019 Talks*. Los Angeles California: ACM, July 2019, pp. 1–2. ISBN: 978-1-4503-6317-4. DOI: `10.1145/3306307.3328150`. (Visited on 02/20/2026).

[5] *DS/ISO 12233 Digital cameras - Resolution and spatial frequency responses*. 2024.

[6] Thomas Goossens et al. "Ray-transfer functions for camera simulation of 3D scenes with hidden lens design". en. In: *Optics Express* 30.13 (June 2022), p. 24031. ISSN: 1094-4087. DOI: 10.1364/OE.457496.

[7] Johannes Hanika and Carsten Dachsbacher. "Efficient Monte Carlo rendering with realistic lenses". In: *Computer Graphics Forum* 33.2 (2014), pp. 323–332. DOI: 10.1111/cgf.12301.

[8] Jan Honsbrok et al. "Ray-LUT: A Lookup-Based Method for Camera Lens Simulation in Real-Time Using Ray Tracing:" in: *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2025, pp. 177–184. ISBN: 978-989-758-728-3. DOI: 10.5220/0013105900003912.

[9] Matthias B. Hullin, Johannes Hanika, and Wolfgang Heidrich. "Polynomial Optics: A Construction Kit for Efficient Ray-Tracing of Lens Systems". In: *Computer Graphics Forum* 31.4 (2012), pp. 1375–1383. DOI: 10.1111/j.1467-8659.2012.03132.x.

[10] *iQ-Analyzer-X*. Kerpen, Germany, 2025. URL: https://www.image-engineering.de/products/software/iq-analyzer-x.

[11] *ISO 19567-2 Photography — Photography — Digital camerasPart 2: Texture analysis using stochastic pattern*. 2019.

[12] A. Keller et al. "The path tracing revolution in the movie industry". en. In: *ACM SIGGRAPH 2015 Courses*. Los Angeles California: ACM, July 2015, pp. 1–7. ISBN: 978-1-4503-3634-5. DOI: 10.1145/2776880.2792699.

[13] Craig Kolb, Don Mitchell, and Pat Hanrahan. "A Realistic Camera Model for Computer Graphics". In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: Association for Computing Machinery, 1995, pp. 317–324. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218463.

[14] Frank Meinl. *Crytek Sponza*. 2010. URL: https://casual-effects.com/g3d/data10/index.html#mesh8 (visited on 11/13/2023).

[15] Patrick Mueller, Matthias Lehmann, and Alexander Braun. "Simulating tests to test simulation". In: *Electronic Imaging* 32.16 (Jan. 2020), pp. 149–1–149–8. ISSN: 2470-1173. DOI: 10.2352/ISSN.2470-1173.2020.16.AVM-149.

[16] *NVIDIA OptiX™ AI-Accelerated Denoiser*. URL: https://developer.nvidia.com/optix-denoiser (visited on 09/15/2025).

[17] Hideki Ogawa and Canon Kabushiki Kaisha. "Zoom lens". US Patent 5537259. Issue: US Patent 5537259. Apr. 1995.

[18] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: from theory to implementation*. eng. Fourth edition. Cambridge London: The MIT Press, 2023. ISBN: 978-0-262-04802-6.

[19] Robin Rouphael et al. "Neural Denoising for Spectral Monte Carlo Rendering". In: *Eurographics 2022 - Posters* (2022), pp. 25–26. DOI: 10.2312/EGP.20221011.

[20] Emanuel Schrade, Johannes Hanika, and Carsten Dachsbacher. "Sparse high-degree polynomials for wide-angle lenses". In: *Computer Graphics Forum* 35.4 (Jan. 2016), pp. 89–97. DOI: 10.1111/cgf.12952.

[21] Wolfgang Sellmeier. "Zur Erklärung der abnormen Farbenfolge im Spectrum einiger Substanzen". In: *Annalen der Physik* 219.6 (Jan. 1871), pp. 272–282. DOI: 10.1002/andp.18712190612.

[22] Thijs Vogels et al. "Denoising with kernel prediction and asymmetric loss functions". en. In: *ACM Transactions on Graphics* 37.4 (Aug. 2018), pp. 1–15. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3197517.3201388.

[23] Ingo Wald and Steven G. Parker. "RTX Accelerated Ray Tracing with OptiX". In: *ACM SIGGRAPH 2019 Courses*. SIGGRAPH '19. New York, NY, USA: Association for Computing Machinery, 2019. ISBN: 978-1-4503-6307-5. DOI: 10.1145/3305366.3340297. URL: https://github.com/ingowald/optix7course (visited on 08/23/2023).

[24] Xianyao Zhang et al. "Neural Denoising for Deep-Z Monte Carlo Renderings". en. In: *Computer Graphics Forum* 43.2 (May 2024), e15050. ISSN: 0167-7055, 1467-8659. DOI: 10.1111/cgf.15050.

[25] Quan Zheng and Changwen Zheng. "Adaptive sparse polynomial regression for camera lens simulation". In: *The Visual Computer* 33.6 (Jan. 2017), pp. 715–724. ISSN: 1432-2315. DOI: 10.1007/s00371-017-1402-9.

Author Biography

Jan Halama received his B.Sc. in 2020 and his M.Sc. in 2024 from the University of Applied Sciences Düsseldorf, Germany. He is currently pursuing a PhD. with a focus on physical-realistic simulation of optical systems in real-time.

Alexander Braun received his diploma in physics and laser fluorescence spectroscopy from the University of Göttingen in 2001. His PhD research in quantum optics and quantum computers was carried out at the Universities of Hamburg and Siegen in 2007. He started working as an optical designer for camera-based ADAS with the company Kostal, and later became responsible for the optical quality of the series mass production. He is now researching optical metrology, optical models for simulation and the link between optical quality and computer vision performance at the University of Applied Sciences Düsseldorf.