Accelerated Volume Rendering with Volume Guided Neural Denoising

S. Jabbireddy[1], S. Li[1], X. Meng[1], J. E. Terrill[2], A. Varshney[1]

[1]University of Maryland College Park, [2]National Institute of Standards and Technology, Maryland

Abstract

Monte Carlo path tracing techniques create stunning visualizations of volumetric data. However, a large number of computationally expensive light paths are required for each sample to produce a smooth and noise-free image, trading performance for quality. High-quality interactive volume rendering is valuable in various fields, especially education, communication, and clinical diagnosis. To accelerate the rendering process, we combine learning-based denoising techniques with direct volumetric rendering. Our approach uses additional volumetric features that improve the performance of the denoiser in the post-processing stage. We show that our method significantly improves the quality of Monte Carlo volume-rendered images for various datasets through qualitative and quantitative evaluation. Our results show that we can achieve volume rendering quality comparable to the state-of-the-art at a significantly faster rate using only one sample path per pixel.

CCS Concepts

• ***Computing methodologies*** → ***Ray tracing; Neural networks;***

1. Introduction

Monte Carlo (MC) path tracing is one of the most widely accepted methods to obtain physically-based volumetric visualizations. Capturing indirect global illumination is necessary to achieve photo realism. Cinematic [EDCNJ*17, DHF*16] or MC path tracing methods estimate light transport by sampling several paths per pixel and computing the light contribution at each point along the ray as it travels through the volume. Recent approaches [HMES20, IGMM22] have shown stunning visualization of medical volumetric content, such as CT scans. The perception and understanding of these volumes can be greatly improved with photo-realistic rendering, especially for inexperienced users. However, MC path tracing methods for volumes require nearly hundreds of rays per pixel to converge to a good quality solution [ZOM*22], which is computationally very expensive even with the latest graphics hardware.

Volume rendering involves solving the light transport integral [Kaj86]. The emission-absorption model is commonly used for scientific visualizations. Estimating the volume rendering integral involves casting a ray for each pixel and generating several samples along the ray till it exits the volume. However, these models ignore the scattering effects and lack realism. When scattering is also considered, the computations become even more expensive. Specifically, assume a single scattering model. A secondary ray is cast for each sample point on the view ray to compute the in-scattered light reaching the point. Samples are to be generated along this secondary ray to calculate the transmittance. As a result, many point locations must be sampled in a volume to compute the light integral along a single ray, limiting interactive and real-time performance.

Several efforts have been made to accelerate the volume rendering process. Techniques such as importance sampling [ZZ19, MMR*19, VG95, SA07], adaptive raymarching [Mn14], null-collision algorithms [KHLN17, NGHJ18], empty-space skipping [EHKRS06, HAAB*18] modify the path tracer in-process and effectively select samples in the volume avoiding the regions with less contribution to the final ray color. Alternatively, methods similar to irradiance caching [KVS*14, JDZJ08, RCB11] reuse the lighting calculations. In contrast to these methods, image-space denoising [ZJL*15] is a post-processing method and is gaining popularity in reducing noise in MC renderings. One can apply denoising in addition to any of the accelerating strategies. However, most existing work in denoising MC noise has focused on surface models. Its application to 3D volumes is challenging as there are no well-defined boundaries in volumes. The success of the surface-based MC denoising methods relies heavily on the smooth and noise-free auxiliary geometry buffer information to preserve details in the image. These buffers are generally either noisy or not well-defined for volumes. As a result, designing such equivalent auxiliary buffers for volumetric renderings is a challenging task.

This paper proposes a volumetric denoiser that efficiently reduces MC noise on images rendered at low sample rates. In particular, we use stochastic MC volumetric path tracing to render the volume data using one sample path per pixel (spp), which is later denoised using our volumetric denoising network. Our denoising framework is based on the neural bilateral grid [MZV*20]. We

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.2312/evs.20231042

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

propose additional volumetric features, obtained as a by-product of path tracing, that, when combined with the neural denoiser, achieve interactive frame rates while maintaining high image quality. We show the effectiveness of our proposed method through qualitative and quantitative results on image quality and performance.

2. Neural Volume Denoiser

2.1. Volumetric Path Tracer

Following Salama [Sal07], we render a given volume in two passes, from which the radiance is composited to output the final image.

First pass The first pass deals with the first intersection of the ray with the volumetric region based on the transfer function [LKG*16, VPG11]. We compute diffuse lighting due to a point light source placed above the volume. For each pixel p in the screen space, we generate several rays R_p that originate from the camera. The term *samples per pixel (spp)* corresponds to the number of rays generated for each pixel. Along each ray R_{p_i} ($i \in [1, spp]$), we take samples at multiple locations x_j, where $j = 1...m$, considered at equal interval spacing. The interval size is set to the size of one voxel. At each sample location, we evaluate the opacity of the voxel $\alpha(x_j)$ using the transfer function. On encountering a non-transparent voxel along the path, we compute the light contribution at that location x_j. We obtain the visibility at x_j by casting a ray R_{L_k} from the point light source to x_j and sample along the ray at equal intervals. If an opaque sample is encountered, the point is considered to be occluded. We model ambient occlusion by assuming a spherical uniform light source at a large enough radius surrounding the volume. To approximate the integral over the sphere, we randomly sample a direction and calculate the light contribution from a virtual point light source positioned in that direction.

Second pass The second pass in the volume rendering pipeline models the scattering effects. We assume single scattering. Once the sampled ray makes its first intersection with the volume, it is then transmitted into the volume. We trace this ray and accumulate the attenuation factor till the composited opacity exceeds a threshold. The ray, at this point, scatters in a random direction over the hemisphere centered around the gradient direction. The scattered ray is traced till it leaves the volume, and the environment lighting is composited. The final color is obtained by multiplying the voxel color with the composited light, considering attenuation.

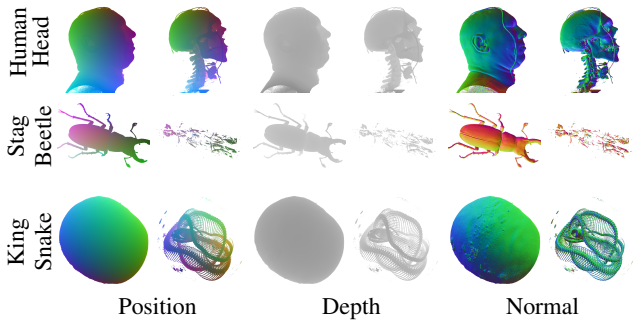

Figure 1: *The figure shows the auxiliary features representing the color-coded position, depth in grayscale, and color-coded normals corresponding to the two events on three datasets.*

2.2. Auxiliary features

We propose using a set of volumetric features obtained as a by-product in our rendering pipeline to guide our denoising network. These features include the depths, positions, and normals corresponding to the first surface-like interaction with the volume and the first scattering event within the volume (Figure 1). The normals are estimated from the gradients of the scalar values. Unlike surface-based models, where such auxiliary features can be easily computed at the locations where the light intersects the scene, these features are not well-defined for 3D volumes. We believe these features provide additional information to the denoiser and preserve the high-frequency details in the image.

2.3. Neural Denoiser

Our volumetric denoising network (VDN) is based on the Neural Bilateral Grid proposed by Meng *et al.* [MZV*20]. The noisy image $r \in \mathbb{R}^{H \times W}$ and the auxiliary features $\mathbf{f}$ from our volumetric path tracer pass through a convolutional neural network [AG17] (GuideNet) to predict a guide image g as shown in Figure 2. A three-dimensional bilateral grid B with resolution $(\lfloor \frac{H}{n_h} \rfloor, \lfloor \frac{W}{n_w} \rfloor, \lfloor \frac{256}{n_d} \rfloor)$ is created by projecting the noisy radiance values onto the grid based on the values from the guide image g. Here, (n_h, n_w, n_d) denotes the sampling factors across each dimension. Multiple levels of bilateral grids can be created using different sampling factors. The bilateral grid is then subjected to spatial filtering to remove the MC noise. Similar to Meng *et al.* [MZV*20], we perform filtering by convolving the bilateral grid with a tent filter $T(x, y, z)$. The filtered bilateral grid $\hat{B}$ can be obtained using Equation 1, where $j = (j_x, j_y)$ represent a pixel in the image, $p \in \mathbb{R}^3$ denotes a spatial location in B, and $\hat{g}(j) = (j_x, j_y, g(j))$.

$$\hat{B}(p) = \frac{\sum_j T(\hat{g}(j) - p) \cdot r(j)}{\sum_j T(\hat{g}(j) - p)} \tag{1}$$

In the last stage, the denoised image $\tilde{R}$ is reconstructed by slicing

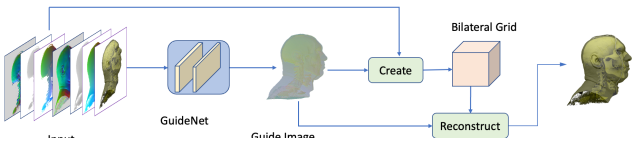

Figure 2: *The figure shows our volumetric denoising pipeline using a single-level neural bilateral grid adapted from Meng* et al. *[MZV*20]. The GuideNet takes the noisy image and the auxiliary features as input and predicts the guide image, which guides bilateral filtering to produce the denoised image.*

the filtered bilateral grid using the same guide image g. The denoised image value at a pixel j is given by Equation 2. $\{p|B(p) \neq 0\}$ denotes the set of non-zero integer grid cells.

$$\tilde{R}(j) = \frac{\sum_{\{p|\hat{B}(p)\neq 0\}} \hat{B}(p) \cdot T(\hat{g}(j) - p)}{\sum_{\{p|\hat{B}(p)\neq 0\}} T(\hat{g}(j) - p)} \tag{2}$$

3. Experimental Setup

Our volume renderer is implemented in CUDA on a workstation with Intel i5-6600 and NVIDIA Quadro RTX 5000 GPU. The de-

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

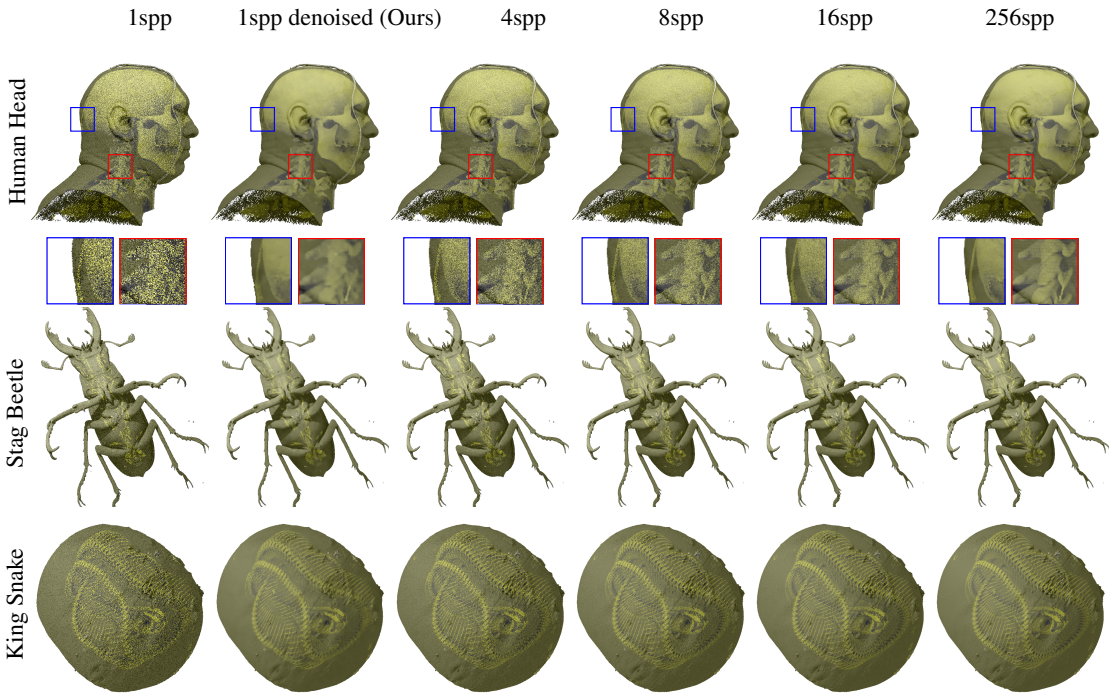

Figure 3: *The figure visually compares images rendered with different sample paths per pixel using our volumetric path tracer. The second column (1spp denoised) shows our result of using VDN to denoise the one sample per pixel Monte Carlo rendered images.*

noising framework is implemented in Tensorflow using an NVIDIA GeForce RTX 2080 Ti GPU. The noisy input images are rendered using one spp, and the ground truth with 256 spp.

Datasets and transfer functions: We validate the performance of our denoising method on a set of 3D volumes: CT scans of *human head*, *stag beetle*, and *king snake*. We define a custom transfer function to map the 3D voxel scalar field $V(x)$ to the color $C(x) \in [0,1]^3$ and the opacity $\alpha(x) \in [0,1]$ of the voxels $x \in [0,1]$,

$$C(x),\ \alpha(x) = \left\{ \begin{array}{ll} (1,1,0), 0.7 & t^l_{outer} < V(x) < t^h_{outer} \\ (1,1,1), 1.0 & t^l_{inner} < V(x) < t^h_{inner} \end{array} \right\} \quad (3)$$

where $0 < t^l_{outer} < t^h_{outer} < t^l_{inner} < t^h_{inner} < 1$ define the two events where we evaluate the effect of light interactions. The voxels with scalar values outside of these ranges are considered transparent.

Metric	1spp	**1spp denoise**	4spp	8spp	16spp	64spp	256spp
			Human Head				
PSNR↑	20.19	**30.52**	25.97	29.00	32.11	39.05	+inf
SSIM↑	0.796	**0.966**	0.873	0.910	0.942	0.984	1.0
Time (ms)↓	55.14	**72.74**	210.64	416.86	833.04	3.3k	13.2k
			Stag Beetle				
PSNR↑	27.80	**35.91**	33.55	36.62	39.69	46.52	+inf
SSIM↑	0.951	**0.992**	0.977	0.987	0.993	0.998	1.0
Time (ms)↓	46.34	**63.84**	198.98	422.06	892.25	3.8k	15.5k
			King Snake				
PSNR↑	22.77	**35.30**	28.75	31.81	34.95	41.89	+inf
SSIM↑	0.832	**0.974**	0.909	0.943	0.967	0.992	1.0
Time (ms)↓	111.00	**128.80**	482.36	969.70	1.9k	7.7k	31.0k

Table 1: *Quantitative metrics on image quality and the overall rendering time per frame, compared to 256-spp images. Our one spp three-level neural bilateral grid denoiser (1spp denoise) achieves high-quality results at interactive frame rates.*

4. Results

Quality comparison Our results are shown in Figure 3. The first column shows the noisy input images, and the second shows the corresponding denoised images. Columns three to six represent images rendered with 4-spp, 8-spp, 16-spp, and 256-spp for reference. The second row shows the image quality of zoomed-in versions of the two regions in the human head. We also validate the quality of the denoised images using two image quality metrics - Structural Similarity Index (SSIM) [WBSS04] and Peak Signal-to-Noise Ratio (PSNR) in Table 1. We observe both qualitatively and quantitatively that the denoised one-spp images have a quality higher than the 8-spp images and comparable to 16-spp images.

Rendering time Our method significantly reduces the end-to-end rendering time while maintaining superior image quality. The overall time depends on the time to render the volumetric path traced image and the time to denoise the generated noisy image. Moreover, the rendering time depends on the volume and the transfer function. In contrast, the denoising time depends only on the resolution of the image once the network is trained. Our three-level neural bilateral grid takes 17 milliseconds to denoise a 1024×1024 resolution image while maintaining a quality higher than the image rendered at 8spp. This improves the rendering time by nearly seven times. When a bilateral grid at a single resolution is used

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

in our denoising network, the run-time further reduces to 13 milliseconds with a slight drop in image quality. We provide a scatter plot visualization of the run times in Figure 4. In comparison, the single autoencoder volumetric denoising network by Hoffman *et al.* [HMES20] takes nearly 63 milliseconds on the same GPU to denoise a 1024×1024 image. We use our volumetric auxiliary features described in Section 2.2 to train the autoencoder. Our experiments show that we can obtain comparable image quality at a $3.5\times$ faster run-time compared to autoencoder-based volumetric denoiser [HMES20] as shown in table 2.

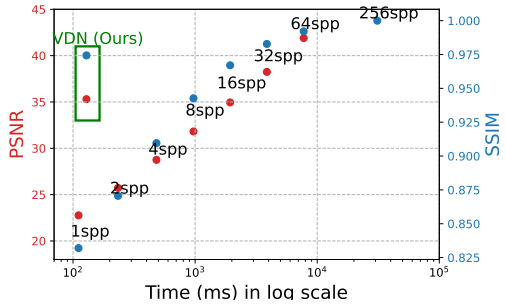

Figure 4: *Our method (VDN) obtains an image quality close to the 16-spp image while maintaining a run-time between one-spp and two-spp images.*

Method	Time	PSNR		
		Human Head	Stag Beetle	King Snake
NDPT [HMES20]	63 msec	31.51	37.16	36.23
VDN (Ours)	**17 msec**	30.52	35.91	35.30

Table 2: *The table shows that VDN produces images with a quality comparable to NDPT [HMES20] but at* $3.5\times$ *faster rates.*

Generalization To analyze the generalizability of our denoiser on datasets not seen during training, we train three different networks that take training images from two volumes and test on the third one. Our results in Table 3 show that the denoiser is generalizable and produces images with a quality close to the independent models where the training and test images come from the same volume. Since the denoiser is trained on cropped patches, the network is not limited to the training dataset alone and can generalize well to remove MC noise from similarly rendered images.

Training data	Testing data	PSNR	SSIM
King Snake and Stag Beetle	Human Head	30.06	0.96
Human Head and King Snake	Stag Beetle	35.26	0.99
Human Head and Stag Beetle	King Snake	34.50	0.97

Table 3: *The table shows the performance of our denoiser when the training and the test images come from different 3D volumes. The high image quality depicts the generalizability of our method.*

Effect of Auxiliary features We compare our approach to a denoiser trained purely on noisy color images to analyze the effect of using the proposed volumetric auxiliary features. Table 4 shows the image quality obtained with and without these volumetric auxiliary features. We report the mean image quality over 50 test views for each dataset from a network trained on 128×128 sized image patches. The results show that the denoiser guided by defined additional features improves the quality of the rendered images.

Data	Only Color		*x*3 Only Color		Color+Aux	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Human Head	29.90	0.956	30.09	0.957	30.31	0.966
Stag Beetle	34.97	0.990	35.56	0.993	35.69	0.992
King Snake	33.72	0.961	34.06	0.964	35.11	0.973

Table 4: *The effect of using volumetric auxiliary features in our denoiser.* ***Only Color*** *does not use any auxiliary features in the denoiser. x*3 ***Only color*** *has network parameters comparable to the network with auxiliary features. We observe that additional feature information consistently improves the image quality.*

5. Discussion and Conclusion

Discussion While several image-space denoisers [BVM*17, GLA*19, CKS*17, IMF*21] have been proposed for surface-based models, we choose the neural bilateral grid in our work. The neural bilateral grid has fast run-time performance along with excellent edge-preserving characteristics, which is essential when visualizing medical data such as CT scans. Another recent approach for denoising [FWHB21] based on kernel prediction architecture also shows an improved run-time and reconstruction quality. It would be interesting to study the effect of the proposed auxiliary features with such denoisers.

Geometric features such as depth and surface normals have shown improved performance in denoising surface-based data using neural networks [BVM*17, KBS15]. In this work, we extract similar features from our volume rendering framework at various light interaction events. Our experiments show that coupling these volumetric features with the denoiser improves the quality of the reconstructions. However, our method is limited to medical volumetric content, where discernible layers are encountered, and a single scattering is sufficient [Sal07]. For phenomena like smoke, fire, and other translucent materials like wax and human skin that need more complex transfer functions, additional volumetric features might be required to effectively guide the denoiser. Integrating advanced automatic feature selection methods [ZOM*22] can steer further studies in this direction. We believe our work will inspire future research to experiment with different features that effectively guide volume denoisers in real time.

Conclusion In this work, we integrate a deep-learning-based denoising method with volume rendering that replaces the heavy pipeline of MC volume rendering to achieve high-quality renderings at interactive rates. Our method first renders a noisy image using only one spp through volumetric path tracing. The noisy image and the proposed volumetric auxiliary features are then passed through the neural bilateral grid denoiser to obtain a high-quality result. Our results show that an image quality higher than 8-spp renderings can be obtained by denoising a one-spp rendered image, achieving a speed-up factor of nearly seven times, thus enabling interactive high-quality volumetric rendering.

Acknowledgements: We thank the anonymous reviewers for their insightful feedback to improve the paper. This work has been supported by NIST award 70NANB16H206. Any opinions, findings, conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the research sponsors.

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.

References

[AG17] ALOYSIUS N., GEETHA M.: A review on deep convolutional neural networks. In *2017 International Conference on Communication and Signal Processing (ICCSP)* (2017), pp. 0588–0592. doi:10.1109/ICCSP.2017.8286426. 2

[BVM*17] BAKO S., VOGELS T., MCWILLIAMS B., MEYER M., NOVÁK J., HARVILL A., SEN P., DEROSE T., ROUSSELLE F.: Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph. 36*, 4 (jul 2017). doi:10.1145/3072959.3073708. 4

[CKS*17] CHAITANYA C. R. A., KAPLANYAN A. S., SCHIED C., SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–12. 4

[DHF*16] DAPPA E., HIGASHIGAITO K., FORNARO J., LESCHKA S., WILDERMUTH S., ALKADHI H.: Cinematic rendering–an alternative to volume rendering for 3D computed tomography imaging. *Insights into imaging 7*, 6 (2016), 849–856. 1

[EDCNJ*17] EID M., DE CECCO C. N., NANCE JR J. W., CARUSO D., ALBRECHT M. H., SPANDORFER A. J., DE SANTIS D., VARGA-SZEMES A., SCHOEPF U. J.: Cinematic rendering in CT: a novel, lifelike 3D visualization technique. *American Journal of Roentgenology 209*, 2 (2017), 370–379. 1

[EHKRS06] ENGEL K., HADWIGER M., KNISS J. M., REZK-SALAMA C.: Real-Time Volume Graphics. In *Eurographics 2006: Tutorials* (2006), Magnenat-Thalmann N., Bühler K., (Eds.), The Eurographics Association. doi:10.2312/egt.20061064. 1

[FWHB21] FAN H., WANG R., HUO Y., BAO H.: Real-time monte carlo denoising with weight sharing kernel prediction network. *Computer Graphics Forum 40*, 4 (2021), 15–27. doi:https://doi.org/10.1111/cgf.14338. 4

[GLA*19] GHARBI M., LI T.-M., AITTALA M., LEHTINEN J., DURAND F.: Sample-based Monte Carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–12. 4

[HAAB*18] HADWIGER M., AL-AWAMI A. K., BEYER J., AGUS M., PFISTER H.: Sparseleap: Efficient empty space skipping for large-scale volume rendering. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 974–983. doi:10.1109/TVCG.2017.2744238. 1

[HMES20] HOFMANN N., MARTSCHINKE J., ENGEL K., STAMMINGER M.: Neural denoising for path tracing of medical volumetric data. *Proc. ACM Comput. Graph. Interact. Tech. 3*, 2 (aug 2020). doi:10.1145/3406181. 1, 4

[IGMM22] IGLESIAS-GUITIAN J. A., MANE P., MOON B.: Real-time denoising of volumetric path tracing for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 28*, 7 (2022), 2734–2747. doi:10.1109/TVCG.2020.3037680. 1

[IMF*21] IŞIK M., MULLIA K., FISHER M., EISENMANN J., GHARBI M.: Interactive monte carlo denoising using affinity of neural features. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459793. 4

[JDZJ08] JAROSZ W., DONNER C., ZWICKER M., JENSEN H. W.: Radiance caching for participating media. *ACM Trans. Graph. 27*, 1 (mar 2008). doi:10.1145/1330511.1330518. 1

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (1986), pp. 143–150. 1

[KBS15] KALANTARI N. K., BAKO S., SEN P.: A machine learning approach for filtering monte carlo noise. *ACM Trans. Graph. 34*, 4 (jul 2015). doi:10.1145/2766977. 4

[KHLN17] KUTZ P., HABEL R., LI Y. K., NOVÁK J.: Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Trans. Graph. 36*, 4 (jul 2017). URL: https://doi.org/10.1145/3072959.3073665, doi:10.1145/3072959.3073665. 1

[KVS*14] KHLEBNIKOV R., VOGLREITER P., STEINBERGER M., KAINZ B., SCHMALSTIEG D.: Parallel irradiance caching for interactive Monte-Carlo direct volume rendering. *Computer Graphics Forum 33*, 3 (2014), 61–70. doi:10.1111/cgf.12362. 1

[LKG*16] LJUNG P., KRÜGER J., GROLLER E., HADWIGER M., HANSEN C. D., YNNERMAN A.: State of the art in transfer functions for direct volume rendering. *Computer Graphics Forum 35*, 3 (2016), 669–691. doi:https://doi.org/10.1111/cgf.12934. 2

[MMR*19] MÜLLER T., MCWILLIAMS B., ROUSSELLE F., GROSS M., NOVÁK J.: Neural importance sampling. *ACM Trans. Graph. 38*, 5 (oct 2019). doi:10.1145/3341156. 1

[Mn14] MUÑOZ A.: Higher order ray marching. *Computer Graphics Forum 33*, 8 (2014), 167–176. URL: http://dx.doi.org/10.1111/cgf.12424, doi:10.1111/cgf.12424. 1

[MZV*20] MENG X., ZHENG Q., VARSHNEY A., SINGH G., ZWICKER M.: Real-time Monte Carlo Denoising with the Neural Bilateral Grid. In *Eurographics Symposium on Rendering - DL-only Track* (2020), Dachsbacher C., Pharr M., (Eds.), The Eurographics Association. doi:10.2312/sr.20201133. 1, 2

[NGHJ18] NOVÁK J., GEORGIEV I., HANIKA J., JAROSZ W.: Monte Carlo Methods for Volumetric Light Transport Simulation. *Computer Graphics Forum 37*, 2 (2018), 551–576. doi:https://doi.org/10.1111/cgf.13383. 1

[RCB11] RIBARDIÈRE M., CARRÉ S., BOUATOUCH K.: Adaptive records for volume irradiance caching. *The visual computer 27*, 6 (2011), 655–664. 1

[SA07] SUBR K., ARVO J.: Steerable importance sampling. In *2007 IEEE Symposium on Interactive Ray Tracing* (2007), pp. 133–140. doi:10.1109/RT.2007.4342601. 1

[Sal07] SALAMA C. R.: GPU-based Monte-Carlo volume raycasting. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)* (2007), IEEE, pp. 411–414. 2, 4

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), pp. 419–428. 1

[VPG11] VUÇINI E., PATEL D., GRÖLLER M. E.: Enhancing visualization with real-time frequency-based transfer functions. In *Visualization and Data Analysis 2011* (2011), Wong P. C., Park J., Hao M. C., Chen C., Börner K., Kao D. L., Roberts J. C., (Eds.), vol. 7868, International Society for Optics and Photonics, SPIE, p. 78680L. doi:10.1117/12.872342. 2

[WBSS04] WANG Z., BOVIK A., SHEIKH H., SIMONCELLI E.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4 (2004), 600–612. doi:10.1109/TIP.2003.819861. 3

[ZJL*15] ZWICKER M., JAROSZ W., LEHTINEN J., MOON B., RAMAMOORTHI R., ROUSSELLE F., SEN P., SOLER C., YOON S.-E.: Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum 34*, 2 (2015), 667–681. doi:https://doi.org/10.1111/cgf.12592. 1

[ZOM*22] ZHANG X., OTT M., MANZI M., GROSS M., PAPAS M.: Automatic feature selection for denoising volumetric renderings. *Computer Graphics Forum 41*, 4 (2022), 63–77. doi:https://doi.org/10.1111/cgf.14587. 1, 4

[ZZ19] ZHENG Q., ZWICKER M.: Learning to importance sample in primary sample space. *Computer Graphics Forum 38*, 2 (2019), 169–179. doi:https://doi.org/10.1111/cgf.13628. 1

© 2023 The Authors.
Proceedings published by Eurographics - The European Association for Computer Graphics.