Computational Visual Media
https://doi.org/10.1007/s41095-023-0372-2 Vol. 10, No. 5, October 2024, 937–948

Research Article

Efficient participating media rendering with differentiable regularization

Wenshi Wu[1], **Beibei Wang**[1] (✉), **Miloš Hašan**[2], **Lei Zhang**[3], **Zhong Jin**[1], **and Ling-Qi Yan**[4]

© The Author(s) 2024.

Abstract Highly scattering media, such as milk, skin, and clouds, are common in the real world. Rendering participating media is challenging, especially for high-order scattering dominant media, because the light may undergo a large number of scattering events before leaving the surface. Monte Carlo-based methods typically require a long time to produce noise-free results. Based on the observation that low-albedo media contain less noise than high-albedo media, we propose reducing the variance of the rendered results using differentiable regularization. We first render an image with low-albedo participating media together with the gradient with respect to the albedo, and then predict the final rendered image with a low-albedo image and gradient image via a novel prediction function. To achieve high quality, we also consider the gradients of neighboring frames to provide a noise-free gradient image. Ultimately, our method can produce results with much less overall error than equal-time path tracing methods.

Keywords participating media; differentiable regularization; differentiable rendering; volumetric path tracing; temporal denoising

1 Introduction

Participating media, such as milk, wax, and skin, are common in daily life. In computer graphics, it is challenging to simulate light transport in participating media because low-order scattering-dominant participating media produce volumetric caustics, where the sharp features are difficult to capture; high-order scattering-dominant participating media produce smooth effects, but the light rays might undergo many scattering events before leaving the surface. In this paper, we focus on high-order scattering dominant homogeneous participating media without refractive boundaries.

Several approaches have been proposed to render participating media. Density estimation-based approaches [1–6] have been widely used to render participating media due to their high efficiency. However, most methods are complex and biased. In contrast, Monte Carlo-based approaches [7–9] are much simpler but require a long time to converge because of the long light paths within the media. Several strategies have been proposed to improve the convergence of Monte Carlo-based approaches for media rendering, including path guiding [10, 11], zero-variance-based approaches [12, 13], and precomputation-based approaches [14], which have successfully reduced the variance. Our method is also based on Monte Carlo-based path tracing, but it improves the rendering quality in a different manner. Moreover, our method can be combined with the above methods to further improve convergence.

In this paper, we propose a differentiable regularization-based approach based on the observation that the rendered result of low-albedo media has less noise than that of high-albedo media. First, we render the image with a small albedo while the other configurations are the same. During rendering, we compute the path radiance, along with the radiance

---

1 School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China. E-mail: W. Wu, wws19990401@njust.edu.cn; B. Wang, beibei.wang@njust.edu.cn (✉); Z. Jin, zhongjin@njust.edu.cn.
2 Adobe Research, San Jose, CA 95110-2704, USA. E-mail: milos.hasan@gmail.com.
3 Department of Computing, Hong Kong Polytechnic University, Hong Kong 999077, China. E-mail: john.zhang@polyu.edu.hk.
4 Department of Computer Science, University of California, Santa Barbara, CA 93106, USA. E-mail: lingqi@cs.ucsb.edu.
Manuscript received: 2023-02-08; accepted: 2023-08-15

gradient, with respect to the albedo, resulting in a rendered radiance image and gradient image. Next, we use the radiance and gradient images to predict the radiance image for the original albedo. Because low-albedo media have less noise than high-albedo media, our method can reduce the noise of the final image by using differentiable regularization [15]. Furthermore, we improve the quality of the gradient image using temporal denoising. Our main contributions can be summarized as follows:

- a differentiable regularization framework to improve quality in participating media rendering,
- a robust radiance prediction model to predict images for high-albedo media from a low-albedo rendered image and gradient image, and
- a temporal gradient denoising approach to further improve image quality.

In Section 2, we briefly review related works. We introduce some preliminaries of the proposed method in Section 3 and the three steps of our method in Section 4. We then validate and analyze our method in Section 5 and conclude our contributions in Section 6.

2 Related work

In this section, we briefly review some Monte Carlo-based participating media rendering methods (the reader can refer to Wu et al. [16] for more methods on homogeneous participating media rendering). We then introduce related studies on differentiable regularization, path space regularization, participating media rendering with radiance gradients, differentiable rendering, and scattering parameter exploitation.

Monte Carlo-based participating media rendering. Rushmeier [17] first introduced path tracing to volumetric rendering by solving the radiative transfer equation. Lafortune and Willems [8] first proposed bidirectional volumetric path tracing. Later, more methods [9] were extended to volumetric rendering. Sampling is an important problem in participating media rendering because there are many scattering events. Efficient sampling improves the convergence speed. Several lines of research have aimed to improve sampling in participating media, including path guiding [11], manifold next event estimation [18], zero-variance random walk [12, 13], and precomputation [14].

All of these methods target better sampling for media rendering. Unlike the above works, our method improves the rendering quality with differentiable regularization and can be combined with these methods to further improve the quality of the results.

Differentiable regularization. Differentiable regularization was first proposed by Fan et al. [15] for glints rendering. They rendered the scene with a changed configuration (larger light source size and larger surface roughness) to obtain a lower-noise rendered radiance image and gradient image and then predicted the rendered results under the desired configuration. Our method is inspired by this and introduces differentiable regularization for participating media rendering. However, there are some key differences in our work. First, we perform temporal denoising on the gradient image to improve its quality. Second, our prediction function is well-designed to fit the behavior of media rendering rather than using linear or log-linear functions directly.

Path space regularization. The differentiable regularization is performed on the screen space, while another group is performed on the path space. The path space regularization methods render scenes with complex light paths by manipulating the material parameters. Path space regularization was first proposed by Kaplanyan and Dachsbacher [19] for pure specular interactions and was improved by Bouchard et al. [20] by using a custom MIS weight to select between unbiased and biased samplers. This path space regularization idea was subsequently extended to microfacet models by Jendersie and Grosch [21].

The above methods cannot correct the error introduced by regularization, while both Fan et al. [15] and our method can extrapolate from the regularized result to predict the result of the original configurations by using gradient information.

Participating media rendering with radiance gradients. Radiance caching has been used for participating media rendering, where the interpolations consider the radiance gradients in terms of locations. Jarosz et al. [22] proposed analytic expressions for radiance gradient. Their method can handle various media, including isotropic, anisotropic, homogeneous, and heterogeneous media of arbitrary albedo. In their subsequent work, Jarosz et al. [23] considered volumetric shadow boundaries and occlusions in their occlusion-aware gradients. However, they ignored light transport within media when computing the gradients. To support both transmittance and occlusions, Marco et al. [24] proposed an efficient computation of radiance derivatives by separating the scene into a discrete set of geometries and media around points on the surface.

The above methods all use the gradients at different locations for radiance interpolation, while our method exploits the radiance gradients with respect to albedo and uses them for rendered image prediction.

Differentiable rendering methods. Differentiable rendering methods compute the derivation of a rendered image with respect to arbitrary scene parameters, such as light sources, camera positions, and object position. Li et al. [25] proposed the first general-purpose differentiable path tracer, which samples the Dirac delta functions from the derivatives of the discontinuous integrand. Loubet et al. [26] proposed a reparameterization technique for differentiating path-traced images to improve performance. Zhang et al. [27] introduced a differential theory of radiative transfer for volume rendering, and Zhang et al. [28] proposed a path-space differentiable rendering formulation. Recently, Zhang et al. [29] formulated an analytical form of generalized differential path integrals that can capture light transport on surfaces and within the media. Nimier-David et al. [30] developed a versatile renderer, Mitsuba 2, which offers a GPU-based differentiable rendering framework, and Nimier-David et al. [31] introduced radiative back-propagation to this framework to improve the scalability and efficiency. More details can be found in Zhao et al. [32].

Recently, Zhang et al. [33] introduced the antithetic sampling of BSDFs and light-transport paths to Monte Carlo differentiable rendering, which can accelerate the convergence and is easily integrated into the existing differentiable rendering pipeline.

Our method uses differentiable rendering to render an image with respect to the media albedo. We use automatic differentiation tools from the Eigen library to compute the gradient due to its efficiency.

Scattering parameter exploitation. Hašan and Ramamoorthi [34] used the intensities of homogeneous media with different albedo and the corresponding derivatives of each image pixel to approximate the heterogeneous target media. Zhao et al. [35] used similarity theory to find low-order scattering dominant media to accelerate Monte Carlo rendering. All of these methods exploit scattering parameters to reduce rendering difficulties. In contrast, our method uses differentiable rendering.

3 Background

In this section, we describe the properties of participating media in Section 3.1 and then review the volume rendering equation in Section 3.2.

3.1 Participating media properties

The main parameters of homogeneous participating media include phase function $p(\omega, \omega')$, absorption coefficient σ_{a}, and scattering coefficient σ_{s}. Another approach is to use the extinction coefficient $\sigma_{\mathrm{t}} = \sigma_{\mathrm{a}} + \sigma_{\mathrm{s}}$ and albedo $\alpha = \dfrac{\sigma_{\mathrm{s}}}{\sigma_{\mathrm{t}}}$. The phase function represents the probability density of a ray with an incident direction ω and outgoing direction ω'.

When a medium has a small mean free path (mfp, $l = \dfrac{1}{\sigma_{\mathrm{t}}}$) and high albedo, the light ray might experience thousands of scattering events before leaving the surface, resulting in high variance without sufficient samples. These media are called high-order scattering-dominant media. Examples include skin, wax, marble, and milk. Our method targets rendering of high-order scattering media.

We observed that the rendered results of low-albedo media have less noise than those of high-albedo media with an equal sampling rate because the light path will be shorter with lower albedo, leading to less variance.

3.2 Radiative transfer equation

The light transport in participating media is modeled with Radiative Transfer Equation [36]:

$$\begin{aligned} L(x,\omega) = & T_{\mathrm{r}}(x \leftrightarrow x_{\mathrm{s}}) L(x_{\mathrm{s}}, \omega) \\ & + \int_0^s T_{\mathrm{r}}(x \leftrightarrow x_{\mathrm{t}}) \sigma_{\mathrm{s}}(x_{\mathrm{t}}) L_{\mathrm{i}}(x_{\mathrm{t}}, \omega) \mathrm{d}t \end{aligned} \quad (1)$$

T_{r} is the transmittance, defined as

$$T_{\mathrm{r}}(x \leftrightarrow x_{\mathrm{s}}) = \exp\left(-\int_0^s \sigma_{\mathrm{t}}(x_{\mathrm{t}}) \mathrm{d}t\right) \quad (2)$$

where s is the distance along a ray through the medium to the nearest surface x_{s}, x_{t} is a point, and its distance to the surface x_{s} is between 0 and s. $\sigma_{\mathrm{t}}(x_{\mathrm{t}})$ is the extinction coefficient at point x_{t}. When point x_{t} is outside the media, $\sigma_{\mathrm{t}}(x_{\mathrm{t}}) = 0$. $L(x_{\mathrm{s}}, \omega)$ is computed using the Rendering Equation [37]. $L_{\mathrm{i}}(x_{\mathrm{t}}, \omega)$ is the in-scattering radiance at x_{t} , which collects the incident radiance from all directions over the unit sphere $\Omega_{4\pi}$ according to the phase function p, defined as

$$L_{\mathrm{i}}(x_{\mathrm{t}},\omega)=\int_{\Omega_{4\pi}} p(\omega,\omega_{\mathrm{t}})L(x_{\mathrm{t}},\omega_{\mathrm{t}})\mathrm{d}\omega_{\mathrm{t}} \tag{3}$$

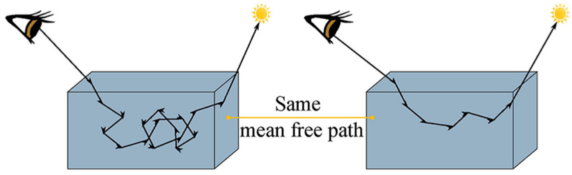
4 Method

We observed that low-albedo media have less noise than high-albedo media when rendering with Monte Carlo-based path tracing. Therefore, we render the images using modified lower-albedo media and predict the results for the actual media configuration, as shown in Fig. 1.

Our method consists of three steps: First, we perform a differentiable volumetric path tracing on a modified media configuration (see Section 4.1) and record the radiance longside the gradient with respect to the albedo. Second, we perform temporal denoising of the gradient image (see Section 4.2). Third, we predict the results for the original media using the modified radiance and denoised gradient images (see Section 4.3). A radiance image and denoised gradient image are shown in Fig. 2.

4.1 Differentiable volumetric path tracing

Rendering participating media with high albedo is challenging because the light path has slow attenuation, resulting in long light paths and high variance. By contrast, media with low albedo show faster attenuation along the light path, resulting in less noise. Therefore, we propose manipulating the albedo parameter and computing the gradient with respect to the albedo via simple differentiable volumetric path tracing.



(a) Difficult path (b) Easy path

Fig. 1 For high-order scattering-dominant media, the light path is extremely long, especially when the albedo is high because the attenuation after each scattering is minor; when the albedo is low, the light path will be shortened as the attenuation becomes quicker. Thus, low-albedo media will have less noise.

Rendered radiance image Gradient image

Fig. 2 Source image (left) and denoised gradient image (right) for the Lucy scene. In this example, the extinction coefficient σ_{t} is 10, target albedo α_{t} is 0.97, source albedo is $0.93\times\alpha_{\mathrm{t}}$, and sample rate is 512.

Equation (1) can be rewritten as a path integral as Eq. (4):

$$L(x,\omega)=\int_{\mathbb{S}_{\mathrm{p}}} f(\tilde{p})\mathrm{d}\tilde{p} \tag{4}$$

where $\mathbb{S}_{\mathrm{p}}$ is a path space that contains all paths starting from x in the direction ω and ending at the light source.

Using Monte Carlo sampling on Eq. (4), Eq. (5) is obtained:

$$L(x,\omega)=\frac{1}{M}\sum_{i=1}^{M}\frac{f(\tilde{p}_i)}{\mathrm{pdf}(\tilde{p}_i)} \tag{5}$$

where M is the sampling count, and pdf is the probability density function (pdf) to sample path $\tilde{p}_i$, whose contribution is defined as

$$\begin{aligned} f(\tilde{p}_i)=&L_{\mathrm{e}}(x_K,-\omega_{K-1})\\ &T_{\mathrm{r}}(x_{K-1}\leftrightarrow x_K)\prod_{k=1}^{K-1} w_k g(\omega_{k-1},\omega_k)\end{aligned} \tag{6}$$

where K is the maximum number of bounces, L_{e} represents the radiance emitted by the light source, $x_0, x_1, \cdots, x_K$ is a sequence of bounce points along the path, x_0 and x_K define the start and end of the path, and w_k is the weight of path $x_{k-1}\leftrightarrow x_k$:

$$w_i=\begin{cases}\alpha\sigma_{\mathrm{t}}T_{\mathrm{r}}(x_{i-1}\leftrightarrow x_i), & x_i\in\text{medium}\\ T_{\mathrm{r}}(x_{i-1}\leftrightarrow x_i), & x_i\in\text{surface}\end{cases} \tag{7}$$

and g:

$$g(\omega_{k-1},\omega_k)=\begin{cases}p(-\omega_{k-1},\omega_k), & x_k\in\text{medium}\\ \rho(-\omega_{k-1},\omega_k)\cos(\theta_k), & x_k\in\text{surface}\end{cases} \tag{8}$$

where $p(\omega_{k-1},\omega_k)$ is the phase function, $\rho(\omega_{k-1},\omega_k)$ is the bidirectional scattering distribution function (BSDF), and θ_k is the angle between the surface normal at x_k and direction ω_k. These terms are all independent of albedo α. We only consider the albedo in $f(\tilde{p}_i)$, and its gradient can be derived analytically.

In practice, we use automatic differentiation to simultaneously obtain the radiance and gradient with respect to albedo at the same time. In this paper, we used the autodiff tool in the Eigen C++ library to perform automatic differentiation. The radiance and gradient with respect to albedo were computed along a sampled path and recorded in a pixel. We accumulate the radiance and gradient from all sampled paths within a pixel and average them to obtain the radiance and gradient with respect to the albedo for a pixel.

4.2 Temporal gradient denoising

With the rendered color and gradient images for the source albedo, we found that the gradient image suffers from a large amount of noise, as shown in Fig. 3. Thus, we propose performing temporal denoising on the gradient images because it is common to render a sequence rather than a single image. We render the image sequence with a moving camera, where the albedo is set as a source parameter. The motion vector d and velocity v_d in the screen space were computed by combining the camera's moving direction and velocity of the camera parameters.

The temporal denoising includes two steps: reprojection and blending. First, we reproject every pixel of the current frame's neighboring frames to its matching pixel with the motion vector and then blend them to obtain the denoised value. The reprojection function for pixel i returns the pixel coordinate in the target frame and is defined as

$$f_{\text{proj}}(i) = i + d \cdot v_d \cdot t \tag{9}$$

where t is the time.

$$G_{\text{denoised}}(i) = \frac{w_{\text{a}}}{1-(1-w_{\text{a}})^N}\left[\sum_{k=0}^{N-1}(1-w_{\text{a}})^k G_k\left(f_{\text{proj}}^{-1}(i)\right)\right] \tag{10}$$

where k is the distance (number of frames) between the image in the sequence and target image ($k=0$), and G_k is the gradient image with distance k. w_{a} is a smoothing factor set to $w_{\text{a}} = 0.1$ in all test scenes. N is the number of frames used for denoising, which was set to 11 in our implementation. f_{proj}^{-1} is the reciprocal function of reprojection function f_{proj}. Any denoising (not only temporal) can be applied to a gradient image.

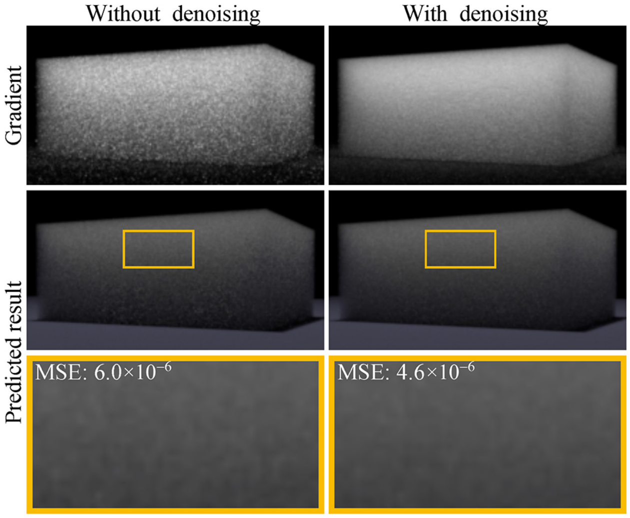


Fig. 3 Comparison of gradient images without (top right) and with (top left) denoising and their predicted results (bottom).

Denoising the gradient is better than denoising the original image because the bias/artifacts from denoising are not directly visible if applied only to the gradient.

4.3 Participating media regularization

With the source pixel value R_{s} and gradient G_{s} at the source albedo α_{s}, we must predict the pixel value R_{t} by using the target albedo α_{t}. The two typical prediction functions are linear and log-linear, as shown by Fan et al. [15]. However, they cannot be used for albedo prediction because the behavior of the media's albedo is complex.

According to the behaviors of the radiance and gradient with changing albedo as shown in Fig. 4, for different pixels and under different extinction coefficients, we found the following. First, the pixel value increases with increasing albedo, and the growth speed increases simultaneously. Second, the growth speed of a pixel's radiance depends on the media's mfp and the outgoing positions obtained in the path tracing process. After the traced paths are shot from the camera and pass through this pixel, they arrive at the boundary of the medium, and several outgoing positions are obtained. These outgoing positions are close to each other. For a pixel whose outgoing positions are close to the light source (such as pixel A in Fig. 4), the growth speed of the radiance is faster in the small mfp than in the large mfp case. For pixels whose outgoing positions are far from the light source (such as pixel C in Fig. 4), the growth speed of the radiance is faster in the large mfp than in the small mfp case. The former types of pixels always have high values, while the latter are close to 0 when the mfp is small.

Therefore, we propose a novel prediction function that includes two exponential functions for extrapolation:

$$R_{\text{t}} = R_{\text{s}} + G_{\text{s}}\beta\left(\mathrm{e}^{(\alpha_{\text{t}}-\alpha_{\text{s}})/\beta} - 1\right)\mathrm{e}^{-\gamma R_{\text{s}}(\alpha_{\text{t}}-\alpha_{\text{s}})^2} \tag{11}$$

The definitions of β and γ are as Eq. (12):

$$\beta = \frac{c_\beta}{\sigma_{\text{t}}\alpha_{\text{s}}}, \quad \gamma = \frac{\ln(\sigma_{\text{t}}+1)^5}{1-\alpha_{\text{s}}} \tag{12}$$

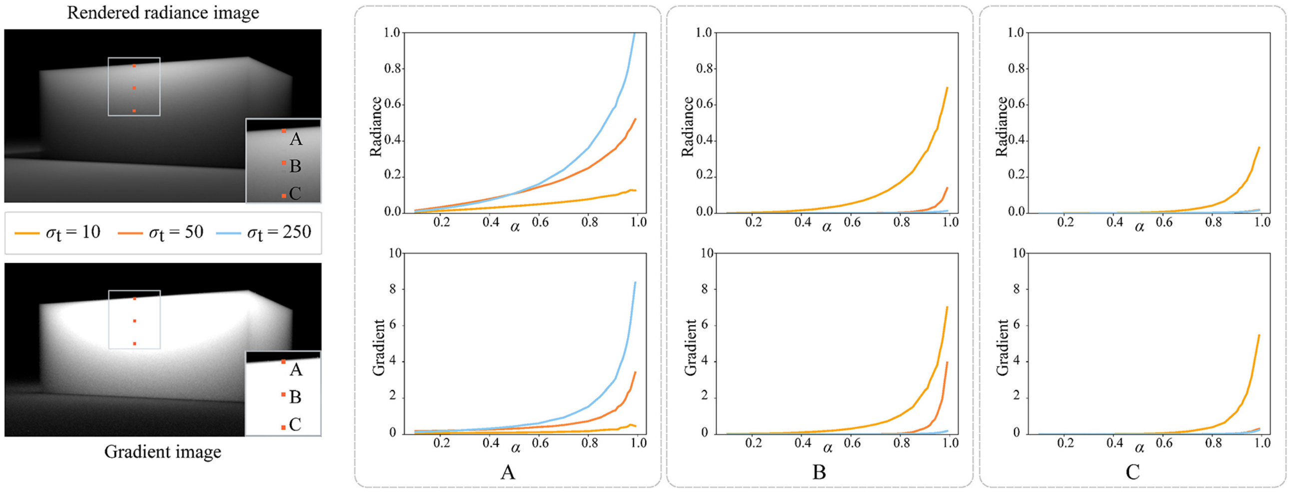


Fig. 4 Radiance curves (top) and gradient curves (bottom) as functions of the albedo for three pixels with different extinction coefficients (shown in different colors). The shapes of these curves inspired us the design of our prediction function (Eq. (11)).

where c_β $(0 < c_\beta \leqslant \sigma_t\alpha_s)$ is a constant such that $\beta \leqslant 1$. In practice, we always search for the best c_β around 0.5.

Our prediction function works for a wide range of materials, from large mfp to small mfp, as shown in Fig. 6. We also compare our prediction function with the linear and log-linear prediction functions, as shown in Fig. 7, and our prediction function better fits the ground truth.

4.4 Design details of the prediction function

According to the observations, we use two exponential terms to form our prediction function. The first exponential term in Eq. (11) ensure that our prediction function increases with increasing albedo and increasing extinction coefficient. The second exponential term in Eq. (11) fine-tunes the shape of the prediction function. When the source albedo is close to 1, mfp is small, G_s is high, and γ is required to control the high value of the first exponential term. Then, we set $1-\alpha_s$ as the denominator of the γ term. For pixels whose values are always high in different mfp cases, the growth speed at this pixel increases as the extinction coefficient increases. Meanwhile, the speed of alleviation should be slower at the same time. Then, we set γ term as a logarithmic function with respect to the extinction coefficient. When R_s is close to 0, G_s is also close to 0. Therefore, the attenuation effect of the second exponential term must be restrained. We then introduce R_s into the second exponential term.

5 Results

We have implemented our algorithm in Mitsuba 0.5 [38]. We compare our method with volumetric path tracing (PT) with equal time and use converged PT as the ground truth. All timings in this section are measured on an Intel i7-10700 @2.90 GHz (16 cores) with 16 GB of main memory. We use mean squared error (MSE) to measure the difference from the reference. In this paper, we focus on participating media with high albedo; the media configurations are shown in Table 1, and we use Henyey–CGreenstein phase function in our scenes. For convenience, if all channels of the albedo or extinction coefficient have the same value, we show only one value.

In the following, by *source*, we mean the rendered

Table 1 Media configurations, scene settings, computation time, and mean squared error (MSE) for our test scenes. g is the mean cosine of the phase function. Spp. represents samples per pixel for path tracing

Scene	Media			Resolution	Ours				Pt.			Reference	
	g	σ_t	α_t		Factor	Spp.	Time (s)	MSE	Spp.	Time (s)	MSE	Spp.	Time (min)
Bunny	0.7	10	0.95	500 × 500	0.95	256	29.67	$\mathbf{1.1}\times10^{-4}$	200	29.76	2.3×10^{-4}	16×10^3	45.61
Candle	0.8	10	[0.98, 0.94, 0.88]	256 × 256	0.95	2048	72.60	$\mathbf{6.1}\times10^{-5}$	1300	70.20	1.3×10^{-4}	16×10^3	15.89
Lucy	0.8	10	0.97	512 × 512	0.93	512	68.40	$\mathbf{1.5}\ \times10^{-5}$	350	69.60	4.4×10^{-5}	16×10^3	60.00

result with low albedo and without prediction, while *target* represents final predicted result with the desired albedo.

5.1 Quality validation

In Fig. 5, we compare our method with temporal denoised volumetric path tracing and volumetric path tracing with equal rendering time. This temporal denoising method is the same as that used for our gradient denoising method. We also show the error maps between our method and reference and between path tracing and the reference. By comparison, our method produces fewer errors than denoised path tracing and path tracing without denoising both visually and quantitatively. Moreover, we found that applying temporal denoising to path tracing does not perform better than path tracing with equal time. These three scenes include both simple shapes (Bunny and Candle scenes) and complex shapes (Lucy scene). In both cases, our method outperforms path tracing.

In Fig. 6, we compare our method with path tracing with equal time on the Bunny scene with varying

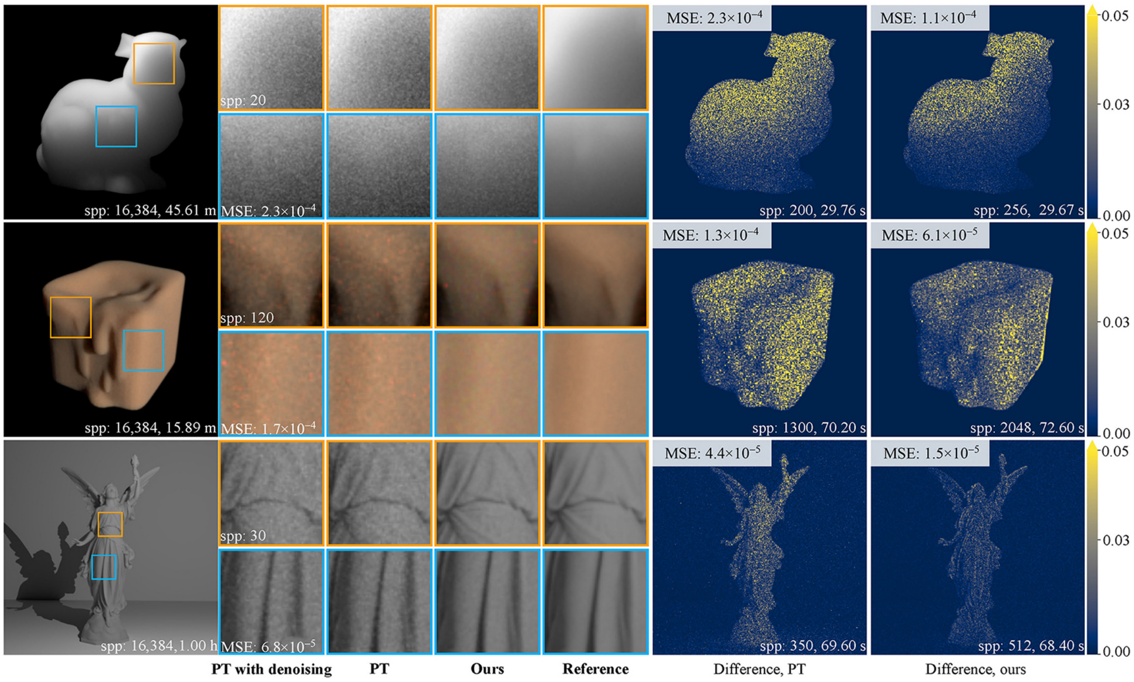


Fig. 5 Comparison among our method, temporal denoised path tracing with equal time, path tracing with equal time, and the reference. The difference between path tracing and the reference and that between our method and the reference are shown in the last two difference maps.

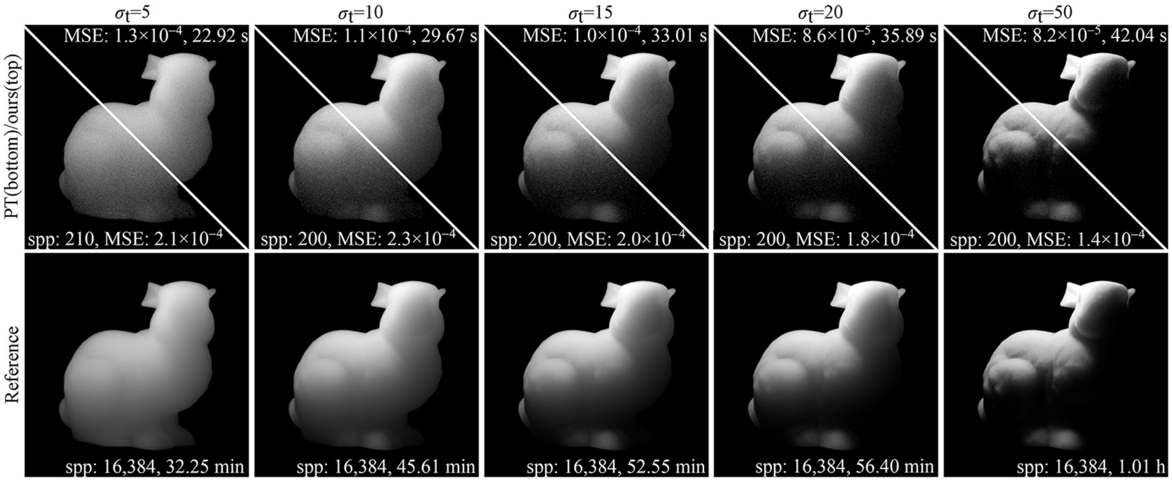


Fig. 6 Comparison between our method and volumetric path tracing (equal-time) on media with varying extinction coefficients $\sigma_t \in \{5, 10, 15, 20, 50\}$. Our method consistently produces better results than path tracing consistently. The sample rate of the source scene is set as 256.

extinction coefficient σ_t. Our method consistently produces less noise in the range of σ_t, from thin to dense media. This demonstrates that our differentiable regularization framework and prediction function are both robust.

5.2 Parameter analysis

Prediction function is a key component in our method. To validate its benefits, we compare it with the linear and log-linear models of the Cube scene shown in Fig. 7. We visualize the radiance curves for a slice of pixels along the red line in the source image, images predicted by three different prediction functions, and the reference image. The results demonstrate that our prediction function produces the closest result to that of the reference. To validate the second exponential term's benefits, we compare the shapes of our prediction function both with and without the second exponential term with the reference radiance curves at the pixel with high R_s, as shown in Fig. 8. By comparison, we prove that the second exponential term can efficiently control the growth of the first exponential term and ensure that the shape of our prediction function is similar to that of the GT.

Gradient denoising also significantly impacts the results. Figure 3 compares the results with and without gradient denoising. We found that temporal denoising reduces the noise in the gradient image and further improves the rendered quality.

Choice of source albedo. The choice of source albedo significantly affects the final image quality. Figure 9 shows the error curve as a function of the source albedo for the Lucy scene. The curve shows that the best source albedo is around $0.9 - 0.95 \times \alpha_t$. We also show the time cost and error curves in this range ($0.9 - 0.95 \times \alpha_t$).

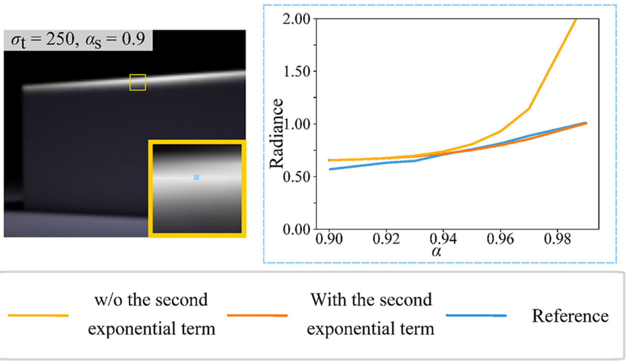


Fig. 8 Impact of the second exponential term in our prediction function. Here, we compare our prediction function with the second exponential term (orange curve), our prediction function without the second exponential term (yellow curve), and the reference radiance curve (blue curve).

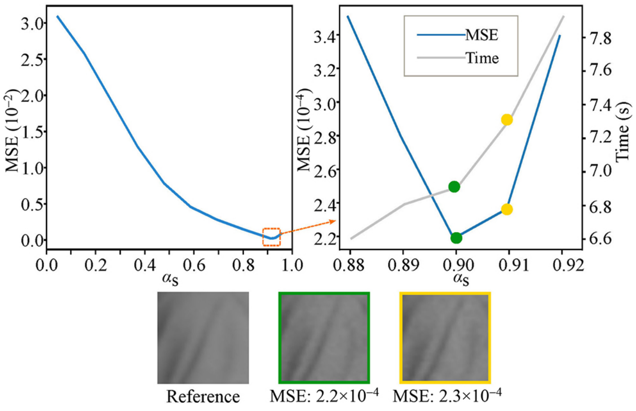


Fig. 9 Error (MSE) curve as a function of source albedo α_s on the Lucy scene. Here, α_s ranges from 0.1 to 0.94 ($0.1 - 0.97 \times \alpha_t$). The sample rate of the source scene is set as 512. The errors are measured on cropped images (30×30). The error and time cost curve (rightmost) are on a small range of the source albedo, $0.9 - 0.95 \times \alpha_t$. The images corresponding to the orange and green dots are shown at the bottom.

In Fig. 10, we show both the source image and target image with different source albedo. Using the source albedo far from the target albedo (e.g., with 0.5 or 0.7 as a factor) results in a large bias. As the source albedo increases, the bias decreases, while the variance increases. Thus, the best source albedo is

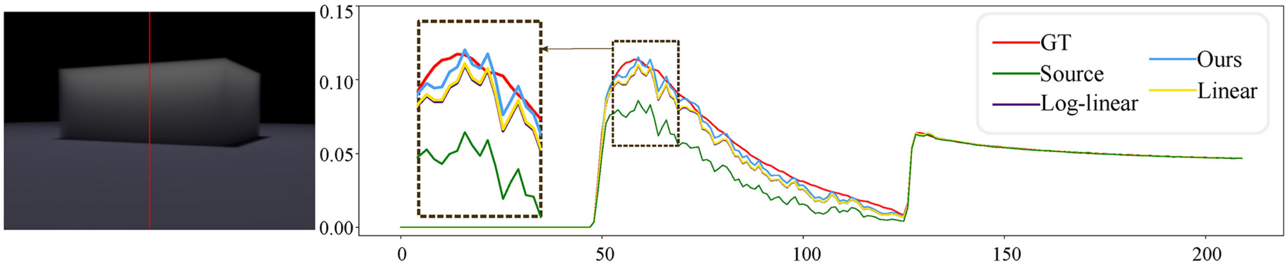


Fig. 7 Radiance variation curves on a slice of pixels (locations shown as the red line in the left image) of the ground truth (GT), predicted results of linear, log-linear, our prediction functions, and source image. The source $\alpha_s = 0.95 \times \alpha_t$ and target $\alpha_t = 0.95$. Our prediction function is the closest to the GT.

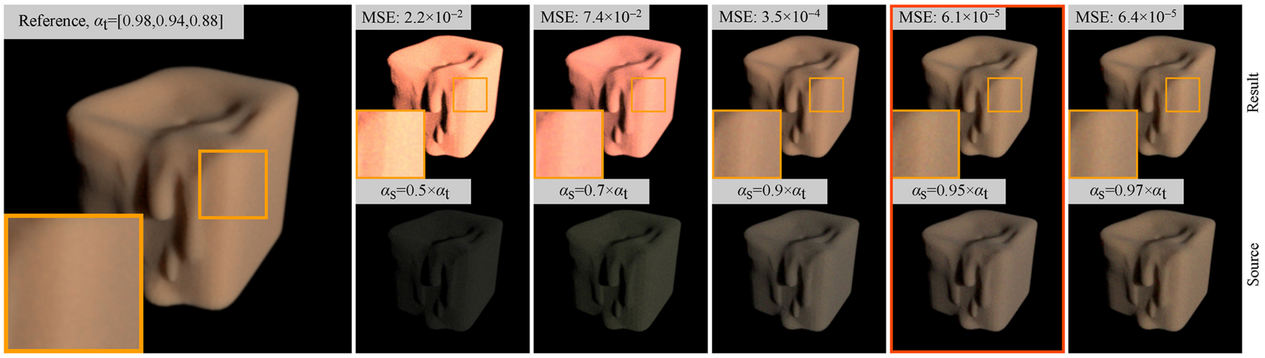


Fig. 10 Predicted results (top) from different sources (bottom) on the Candle scene. When the source albedo is far from the target albedo, there is obvious color bias after the prediction.

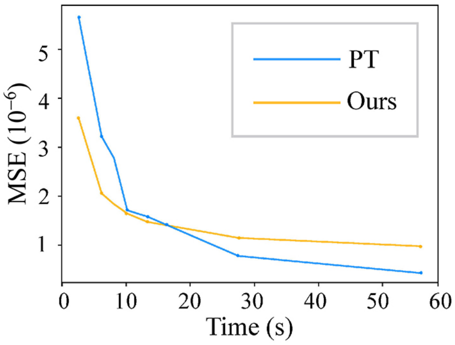


Fig. 11 MSE between our method or path tracing with the reference (rendered with 33,768 spp) over varying rendering time on the Cube scene. The error of our method decreases until approximately 16 s and then remains almost constant. The error before 16 s comes from both variance and bias. After the result is converged, the error comes from the bias. Therefore, our method is suitable for rendering with a low time budget.

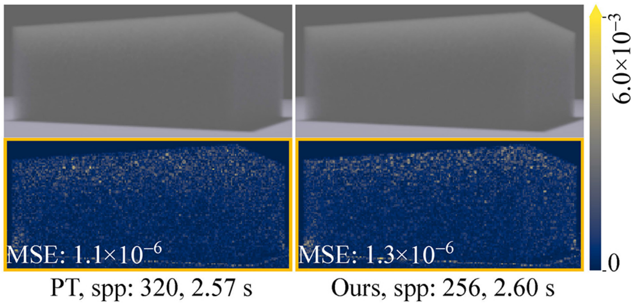


Fig. 12 Failure case: comparison between our method and path tracing on rendering the low-albedo media. The source $\alpha_s = 0.9 \times \alpha_t$ and target $\alpha_t = 0.5$. The extinction coefficient is 10.

around $0.9 - 0.95 \times \alpha_t$.

Considering the accuracy and time cost, the source albedo should be chosen as $0.9 - 0.95 \times \alpha_t$ in practice.

5.3 Performance measurement

In Table 1, we report all the scene-setting, computation time, and their MSEs with the reference rendering results of our test scenes. The cost of prediction and denoising is negligible compared with the rendering time 0.2 s for a 512×512 image. Thus, we ignored this in our computation time. Our method produces higher-quality results than volumetric path tracing, with equal time for all scenes. We use 11 frames for gradient denoising for all our scenes. As can be seen from the table, our method has a higher sample rate than equal-time path tracing, as a lower albedo leads to shorter paths and less rendering time.

5.4 Limitations and discussion

We have identified several limitations of our method. Our method introduces bias into the rendered results as a trade-off between bias and variance. However, the overall error (bias and variance) still decreases down. Our method targets renderings with a low time budget. When the sample rate is high, our results will show a larger error than path tracing, as shown in Fig. 11. Our method is designed for high-albedo media and does not benefit media with low albedo, as shown in Fig. 12. For this type of media, the light path is short, making path tracing less challenging.

6 Conclusions

We presented a novel differentiable regularization framework to improve the quality of participating media rendering. Our method is simple and suitable for any high-albedo homogeneous participating media. Our method produces fewer noise results than path tracing, although it does introduce bias. Moreover, we can apply other approaches (e.g., path guiding and advanced sampling) in the rendering step of our method, which can improve the rendering quality. Subsequently, with our method's denoising and prediction steps, the noise can be further reduced.

In the future, we intend to improve our differentiable regularization by introducing novel prediction functions (e.g., neural networks) or differentiating more parameters.

Acknowledgements

We thank the reviewers for their valuable comments. This work has been partially supported by the National Natural Science Foundation of China under Grant No. 62172220.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Ling-Qi Yan is the Associate Editor of this journal.

References

[1] Jensen, H. W.; Christensen, P. H. Efficient simulation of light transport in scenes with participating media using photon maps. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, 311–320, 1998.

[2] Jarosz, W.; Nowrouzezahrai, D.; Thomas, R.; Sloan, P. P.; Zwicker, M. Progressive photon beams. *ACM Transactions on Graphics* Vol. 30, No. 6, 1–12, 2011.

[3] Jarosz, W.; Zwicker, M.; Jensen, H. W. The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum* Vol. 27, No. 2, 557–566, 2008.

[4] Křivánek, J.; Georgiev, I.; Hachisuka, T.; Vévoda, P.; Šik, M.; Nowrouzezahrai, D.; Jarosz, W. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics* Vol. 33, No. 4, Article No. 103, 2014.

[5] Bitterli, B.; Jarosz, W. Beyond points and beams: Higher-dimensional photon samples for volumetric light transport. *ACM Transactions on Graphics* Vol. 36 No. 4, Article No. 112, 2017.

[6] Deng, X.; Jiao, S.; Bitterli, B.; Jarosz, W. Photon surfaces for robust, unbiased volumetric density estimation. *ACM Transactions on Graphics* Vol. 38 No. 4, Article No. 46, 2019.

[7] Kajiya, J. T.; Herzen, B. P. V. Ray tracing volume densities. *ACM SIGGRAPH Computer Graphics* Vol. 18, No. 3, 165–174, 1984.

[8] Lafortune, E. P.; Willems, Y. D. Rendering participating media with bidirectional path tracing. In: *Rendering Techniques 1996. Eurographics Workshop on Rendering Techniques*. Pueyo, X.; Schröder, P. Eds. Springer Cham, 91–100, 1996.

[9] Pauly, M.; Kollig, T.; Keller, A. Metropolis light transport for participating media. In: *Rendering Techniques 2000. Eurographics*. Péroche, B.; Rushmeier, H. Eds. Springer Vienna, 11–22, 2000.

[10] Deng, H.; Wang, B.; Wang, R.; Holzschuch, N. A practical path guiding method for participating media. *Computational Visual Media* Vol. 6, No. 1, 37–51, 2020.

[11] Herholz, S.; Zhao, Y.; Elek, O.; Nowrouzezahrai, D.; Lensch, H. P. A.; Křivánek, J. Volume path guiding based on zero-variance random walk theory. *ACM Transactions on Graphics* Vol. 38, No. 3, Article No. 25, 2019.

[12] Křivánek, J.; d'Eon, E. A zero-variance-based sampling scheme for Monte Carlo subsurface scattering. In: Proceedings of the ACM SIGGRAPH Talks, Article No. 66, 2014.

[13] Meng, J.; Hanika, J.; Dachsbacher, C. Improving the dwivedi sampling scheme. *Computer Graphics Forum* Vol. 35, No. 4, 37–44, 2016.

[14] Wang, B.; Ge, L.; Holzschuch, N. Precomputed multiple scattering for light simulation in participating medium. In: Proceedings of the ACM SIGGRAPH Talks, Article No. 35, 2017.

[15] Fan, J.; Wang, B.; Wu, W.; Hašan, M.; Yang, J.; Yan, L. Q. Efficient specular glints rendering with differentiable regularization. *IEEE Transactions on Visualization and Computer Graphics* Vol. 29, No. 6, 2940–2949, 2023.

[16] Wu, W.; Wang, B.; Yan, L. Q. A survey on rendering homogeneous participating media. *Computational Visual Media* Vol. 8, No. 2, 177–198, 2022.

[17] Rushmeier, H. E. Realistic image synthesis for scenes with radiatively participating media. Ph.D. Dissertation. Cornell University, 1988.

[18] Weber, P.; Hanika, J.; Dachsbacher, C. Multiple vertex next event estimation for lighting in dense, forward-scattering media. *Computer Graphics Forum* Vol. 36, No. 2, 21–30, 2017.

[19] Kaplanyan, A. S.; Dachsbacher, C. Path space regularization for holistic and robust light transport. *Computer Graphics Forum* Vol. 32, No. 2pt1, 63–72, 2013.

[20] Bouchard, G.; Iehl, J. C.; Ostromoukhov, V.; Poulin, P. Improving robustness of Monte-Carlo global illumination with directional regularization. In: Proceedings of the SIGGRAPH Asia Technical Briefs, Article No. 22, 2013.

[21] Jendersie, J.; Grosch, T. Microfacet model regularization for robust light transport. *Computer Graphics Forum* Vol. 38, No. 4, 39–47, 2019.

[22] Jarosz, W.; Donner, C.; Zwicker, M.; Jensen, H. W. Radiance caching for participating media, *ACM Transactions on Graphics* Vol. 27, No. 1, Article No. 7, 2008.

[23] Jarosz, W.; Zwicker, M.; Jensen, H. W. Irradiance gradients in the presence of participating media and occlusions. *Computer Graphics Forum* Vol. 27, No. 4, 1087–1096, 2008.

[24] Marco, J.; Jarabo, A.; Jarosz, W.; Gutierrez, D. Second-order occlusion-aware volumetric radiance caching. *ACM Transactions on Graphics* Vol. 37, No. 2, Article No. 20, 2018.

[25] Li, T. M.; Aittala, M.; Durand, F.; Lehtinen, J. Differentiable Monte Carlo ray tracing through edge sampling. *ACM Transactions on Graphics* Vol. 37, No. 6, Article No. 222, 2018.

[26] Loubet, G.; Holzschuch, N.; Jakob, W. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Transactions on Graphics* Vol. 38, No. 6, Article No. 228, 2019.

[27] Zhang, C.; Wu, L.; Zheng, C.; Gkioulekas, I.; Ramamoorthi, R.; Zhao, S. A differential theory of radiative transfer. *ACM Transactions on Graphics* Vol. 38, No. 6, Article No. 227, 2019.

[28] Zhang, C.; Miller, B.; Yan, K.; Gkioulekas, I.; Zhao, S. Path-space differentiable rendering. *ACM Transactions on Graphics* Vol. 39, No. 4, Article No. 143, 2020.

[29] Zhang, C.; Yu, Z.; Zhao, S. Path-space differentiable rendering of participating media. *ACM Transactions on Graphics* Vol. 40, No. 4, Article No. 76, 2021.

[30] Nimier-David, M.; Vicini, D.; Zeltner, T.; Jakob, W. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics* Vol. 38, No. 6, Article No. 227, 2019.

[31] Nimier-David, M.; Speierer, S.; Ruiz, B.; Jakob, W. Radiative backpropagation: An adjoint method for lightning-fast differentiable rendering. *ACM Transactions on Graphics* Vol. 39, No. 4, Article No. 146, 2020.

[32] Zhao, S.; Jakob, W.; Li, T. M. Physics-based differentiable rendering: From theory to implementation. In: Proceedings of the ACM SIGGRAPH Courses, Article No. 14, 2020.

[33] Zhang, C.; Dong, Z.; Doggett, M.; Zhao, S. Antithetic sampling for Monte Carlo differentiable rendering. *ACM Transactions on Graphics* Vol. 40, No. 4, Article No. 77, 2021.

[34] Hašan, M.; Ramamoorthi, R. Interactive albedo editing in path-traced volumetric materials. *ACM Transactions on Graphics* Vol. 32, No. 2, Article No. 11, 2013.

[35] Zhao, S.; Ramamoorthi, R.; Bala, K. High-order similarity relations in radiative transfer. *ACM Transactions on Graphics* Vol. 33, No. 4, Article No. 104, 2014.

[36] Chandrasekhar, S. *Radiative Transfer*. New York, USA: Dover Publications, 1960.

[37] Kajiya, J. T. The rendering equation. In: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, 143–150, 1986.

[38] Jakob, W. Mitsuba renderer. 2010. Available at https://www.mitsuba-renderer.org/

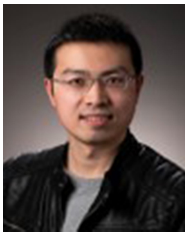
Wenshi Wu is a master student at Nanjing University of Science and Technology (NJUST). She received her bachelor degree from the same university in 2021. Her research interest mainly includes participating media rendering.

Beibei Wang received her Ph.D. degree from Shandong University, in 2014 and visited Telecom ParisTech from 2012 to 2014. She is an associate professor with Nanjing University of Science and Technology. She worked as a postdoc with Inria from 2015 to 2017. She joined NJUST in March 2017. Her research interests include rendering and game development.

Miloš Hašan is a senior research scientist at Adobe Research in San Jose. His research focuses on several areas of computer graphics and rendering, including light transport simulation, material reflectance models, and inverse rendering problems including material capture; he also contributes to production rendering systems. He received his Ph.D. degree in computer science from Cornell University in 2009.

Lei Zhang received his B.Sc. degree in 1995 from Shenyang Institute of Aeronautical Engineering, Shenyang, China, his M.Sc. and Ph.D degrees in control theory and engineering from Northwestern Polytechnical University, Xi'an, China, respectively in 1998 and 2001. From 2001 to 2002, he was a research associate in the Department of Computing, The Hong Kong Polytechnic University. From Jan. 2003 to Jan. 2006 he worked as a postdoctoral fellow in the Department of Electrical and Computer Engineering, McMaster University, Canada. In 2006, he joined the Department of Computing, The Hong Kong Polytechnic University, as an assistant professor. He was promoted to associate professor in Sept. 2010 and to professor in Jul. 2015, and has been a chair professor in the same department since Jul. 2017. His research interests include image and video processing, computer vision, pattern recognition and biometrics, etc.

Zhong Jin is a faculty member of School of Computer Science and Engineering, NJUST since July, 2000. He was promoted to be a professor by NJUST in April, 2002. He is the author of over 150 scientific papers in pattern recognition, image processing, face recognition, applied mathematics. His current interests are in the areas of pattern recognition, image processing, face recognition, facial expression analysis and recognition, content-based image retrieval, environmental perception and understanding.



Ling-Qi Yan is an assistant professor of computer science at UC Santa Barbara, co-director of the MIRAGE Lab, and affiliated faculty in the Four Eyes Lab. Before that, he received his doctoral degree from the Department of Electrical Engineering and Computer Sciences at UC Berkeley and obtained his bachelor degree in computer science from Tsinghua University. His research interests include physically-based rendering, realtime ray tracing, and realistic appearance modeling.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit `http://creativecommons.org/licenses/by/4.0/`.

Other papers from this open access journal are available free of charge from http://www.springer.com/journal/41095. To submit a manuscript, please go to https://www.editorialmanager.com/cvmj.