Efficient Participating Media Rendering with Differentiable Regularization

Wenshi Wu
Nanjing University of Science and Technology
China
wws19990401@njust.edu.cn

Beibei Wang*
Nanjing University of Science and Technology
China
https://wangningbei.github.io/

Miloš Hašan
Adobe Research
USA
http://www.miloshasan.net/

Lei Zhang
Hong Kong Polytechnic University
Hong Kong, China
john.zhang@polyu.edu.hk

Zhong Jin
Nanjing University of Science and Technology
China
zhongjin@njust.edu.cn

Ling-Qi Yan
University of California, Santa Barbara
USA
lingqi@cs.ucsb.edu

Abstract

Highly scattering media are common in the real world, *e.g.*, milk, skin, and cloud. Rendering participating media is challenging, especially for high-order scattering dominant media, since the light might have a large number of scattering events before leaving the surface. Monte Carlo-based methods usually require a long time to produce noise-free results. Based on the observation that lower-albedo media have less noise than higher-albedo media, we propose to reduce the variance of rendered results using differentiable regularization. We first render an image with the low-albedo participating media together with the gradient w.r.t. the albedo and then predict the final rendered image with a low-albedo image and a gradient image via a novel prediction function. To achieve higher quality, we also consider the gradient of the 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 results.

Keywords: participating media, differentiable regularization, differentiable rendering, volumetric path tracing, temporal denoising.

1. Introduction

Participating media are common in daily life, *e.g.*, milk, wax, and skin. In computer graphics, it is challenging to simulate the light transport in the participating media: 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 have a large number of scattering events before leaving the surface. In this paper, we focus on the high-order scattering dominant homogeneous participating media without refractive boundaries.

Several groups of approaches have been proposed to render participating media. Density estimation-based approaches [1, 2, 3, 4, 5, 6] are widely used to render participating media, due to their efficiency. However, these methods are mostly biased and complex. On the contrary, Monte Carlo-based approaches [7, 8, 9] are much simpler but require a long time to converge due to 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]. They have successfully reduced the variance. Our method is also based on the Monte Carlo-based path tracing, but improves the rendering quality in a different way. Moreover, our method can be combined with these methods to improve the convergence further.

In this paper, we propose a differentiable regularization-based approach, under the observation that the rendered result of low-albedo media has less noise than high-albedo media. We first 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 gradient, w.r.t. the albedo, resulting in a rendered radiance image

*Corresponding author.

and a gradient image. Next, we use the radiance and the gradient images to predict the radiance image for the original albedo. Since low-albedo media have less noise than high-albedo media, our method can decrease the noise of the final image by using differentiable regularization [15]. Furthermore, we improve the quality of the gradient image with temporal denoising. To summarize, our main contributions include the following:

- 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 a gradient image, and
- a temporal gradient denoising approach to further improve the image quality.

In the next section, we briefly review the related works. We introduce some preliminary knowledge of our method in Sec. 3 and the three steps of our method in Sec. 4. Then, we validate and analyze our method in Sec. 5 and conclude our contributions in Sec. 6.

2. Related Work

In this section, we first briefly review some Monte Carlo-based participating media rendering methods. Please refer to Wu *et al*. [16] for more methods on homogeneous participating media rendering. Then, we introduce some related works on differentiable regularization, path space regularization, 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] have been extended to volumetric rendering. Sampling is an important problem in participating media rendering since there are many scattering events within media. Efficient sampling will improve the convergence speed. Several lines of work have been proposed 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 these methods target better sampling for media rendering. Unlike their works, our method improves the rendering quality with differentiable regularization and can combine with their methods to further improve the quality of the results.

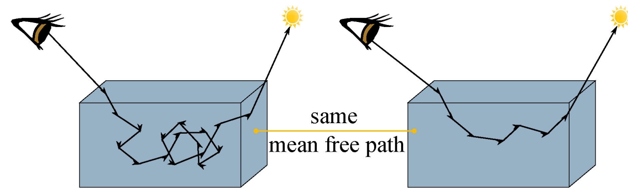


Figure 1. For high-order scattering dominant media, the light path is extremely long, especially when the albedo is high since the attenuation after each scattering is minor; when the albedo is low, the light path will be shortened, since the attenuation becomes quicker. Thus, low-albedo media will have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

Differentiable regularization. Differentiable regularization was first proposed by Fan *et al*. [15] for glints rendering. They render the scene with a changed configuration (larger light source size and larger surface roughness) to obtain a less-noise rendered radiance image and a gradient image and then predict the rendered results under the desired configuration. Our method is also inspired by this method and introduces the differentiable regularization for participating media rendering. However, there are some key differences compared to their 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. The path space regularization was first proposed by Kaplanyan and Dachsbacher [19] for pure specular interactions and improved by Bouchard *et al*. [20] by using a custom MIS weight to select between unbiased and biased samplers. This path space regularization idea is 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 original configurations using the gradient information.

Differentiable rendering methods. Differentiable rendering methods compute the derivation of a rendered image w.r.t. arbitrary scene parameters, such as light sources, camera positions, the position of the object, etc. Li *et al*. [22] proposed the first general-purpose differentiable path tracer, which samples the Dirac delta functions from the derivatives of the discontinuous integrand. Loubet *et al*. [23]

proposed a reparameterization technique for differentiating path-traced images to improve performance. Zhang *et al*. [24] introduced a differential theory of radiative transfer for volume rendering, and Zhang *et al*. [25] proposed a path-space differentiable rendering formulation. Recently, Zhang *et al*. [26] formulated an analytical form of generalized differential path integrals that can capture light transport on the surfaces and within the media. Nimier-David *et al*. [27] developed a versatile renderer Mitsuba 2, which offers a GPU-based differentiable rendering framework, and Nimier-David *et al*. [28] introduced radiative back-propagation in this framework to improve the scalability and the efficiency. More details can be found in Zhao *et al*. [29].

Recently, Zhang *et al*. [30] 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 w.r.t. 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 [31] used the intensities of the homogeneous media with different albedo and the corresponding derivatives of each image pixel to approximate the heterogeneous target media. Zhao *et al*. [32] used the similarity theory to find low-order scattering dominant media to accelerate Monte Carlo rendering. Besides our method, all these mentioned methods exploit scattering parameters to reduce the difficulties of rendering. Compared to these works, our method especially uses differentiable rendering.

3. Background

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

3.1. Participating media properties

The main parameters in homogeneous participating media include phase function $p\,(\omega, \omega')$, absorption coefficient σ_a, and scattering coefficient σ_s. Another way is using extinction coefficient $\sigma_t = \sigma_a + \sigma_s$ and albedo $\alpha = \frac{\sigma_s}{\sigma_t}$. The phase function represents the probability density of a ray with an incident direction ω and an outgoing direction ω'.

When a medium has a small mean free path (mfp, $l = \frac{1}{\sigma_t}$) and a high albedo, the light ray might have thousands of scattering events before leaving the surface, resulting in a 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 high-order scattering media.

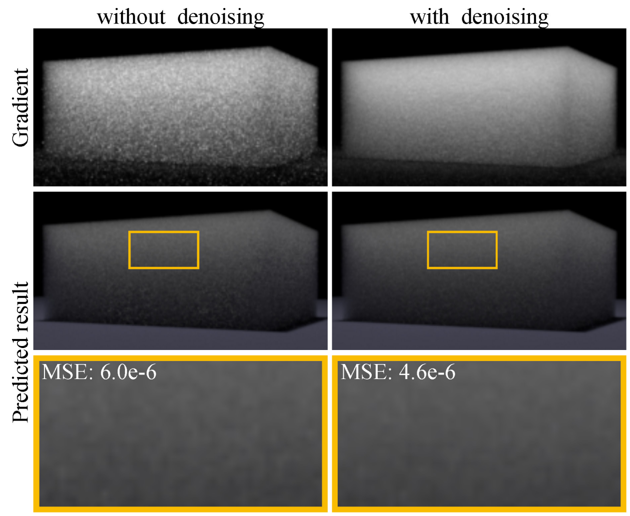


Figure 2. Comparison of gradient images (top) without denoising and with denoising (top) and their predicted results (bottom).

We observed that the rendered results of low-albedo media have less noise than high-albedo media with an equal sampling rate since 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 [33]:

$$\begin{aligned} L(x,\omega) &= T_r(x \leftrightarrow x_s) L(x_s,\omega) \\ &+ \int_0^s T_r(x \leftrightarrow x_t)\sigma_s(x_t) L_i(x_t,\omega)\mathrm{d}t. \end{aligned} \tag{1}$$

T_r is the transmittance, defined as

$$T_r(x \leftrightarrow x_s) = \exp(-\sigma_t \|x - x_s\|), \tag{2}$$

where s is the distance along a ray through the medium to the nearest surface x_s, and x_t is a point, and its distance to surface x_s is between 0 and s. $L(x_s,\omega)$ is computed from Rendering Equation [34]. $L_i(x_t,\omega)$ is the in-scattering radiance at x_t , which collects incident radiance from all directions over the unit sphere $\Omega_{4\pi}$ according to the phase function p, defined as

$$L_i(x_t,\omega) = \int_{\Omega_{4\pi}} p(\omega,\omega_t) L(x_t,\omega_t)\mathrm{d}\omega_t. \tag{3}$$

4. Method

We observe that low-albedo media have less noise than high-albedo media when rendering with Monte Carlo-based

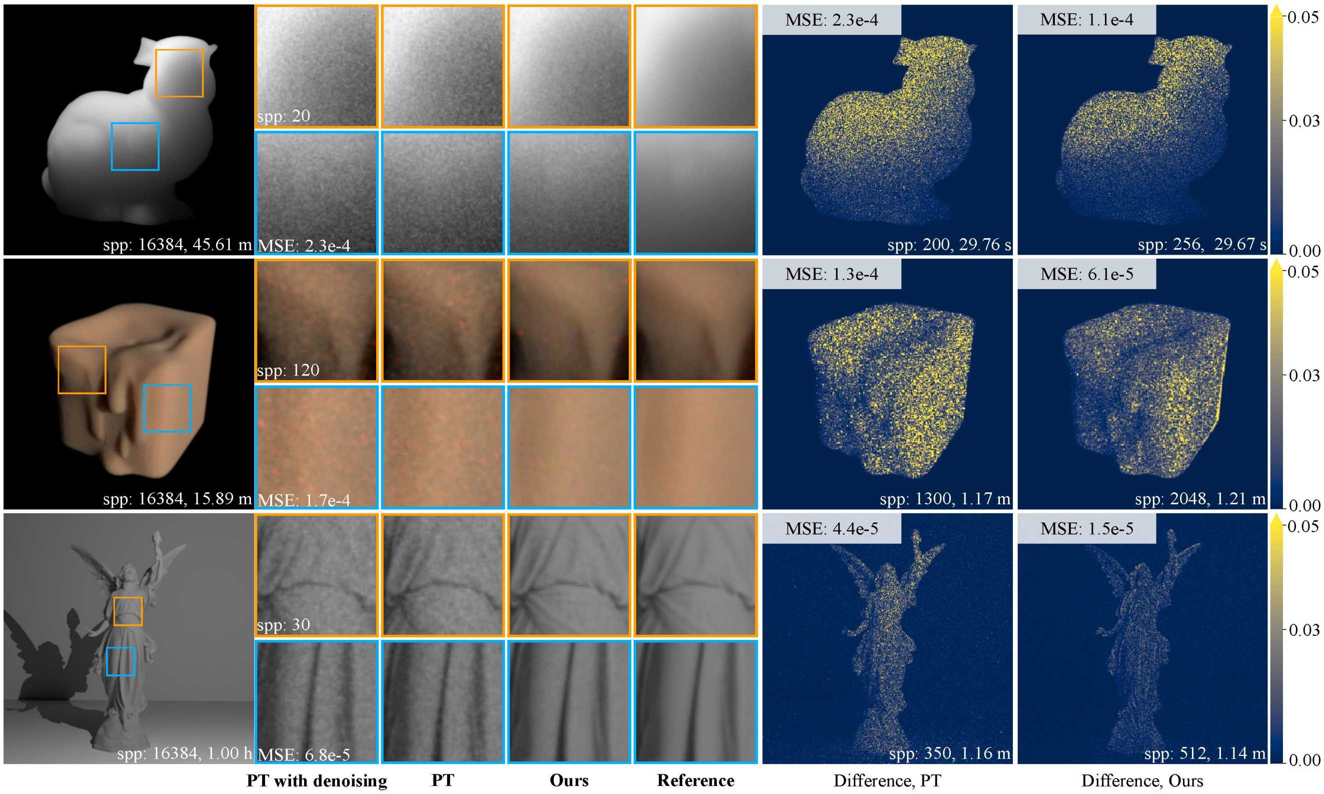


Figure 3. Comparison among our method, temporal denoised path tracing with equal time, path tracing with equal time, and the reference. The difference between path tracing and the reference and the difference between our method and reference are shown in the last two difference maps.

path tracing. Therefore, we render the images with the modified lower-albedo media and predict the result for the actual media configuration, as shown in Figure 1.

Our method includes three steps. First, we perform a differentiable volumetric path tracing on a modified media configuration (see Sec. 4.1) and record the radiance together with the gradient w.r.t. the albedo. Second, we perform temporal denoising for the gradient image (see Sec. 4.2). Third, we predict the results for the original media with the modified radiance image and the denoised gradient image (see Sec. 4.3).

4.1. Differentiable volumetric path tracing

Rendering participating media with high albedo is challenging since the light path has a slow attenuation, resulting in long light paths and the high variance. On the contrary, media with low albedo have a faster attenuation along the light path, resulting in less noise. Therefore, we propose to manipulate the albedo parameter and compute the gradient w.r.t. the albedo via a simple differentiable volumetric path tracing.

Eq. 1 could be rewritten as a path integral:

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

where $\mathbb{S}_p$ is a path space, which contains all paths starting from x with direction ω and ending at the light source.

Using Monte Carlo sampling on Eq. 4 results in the following equation:

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

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:

$$f(\tilde{p_i}) = \prod_{k=1}^{K} w_k g(\omega_{k-1}, \omega_k), \tag{6}$$

where K is the max number of bounces, $x_0, x_1, \cdots, x_K$ is a sequence bounce points along the path, x_0 and x_K define the start and the end of path, w_k is the weight of path

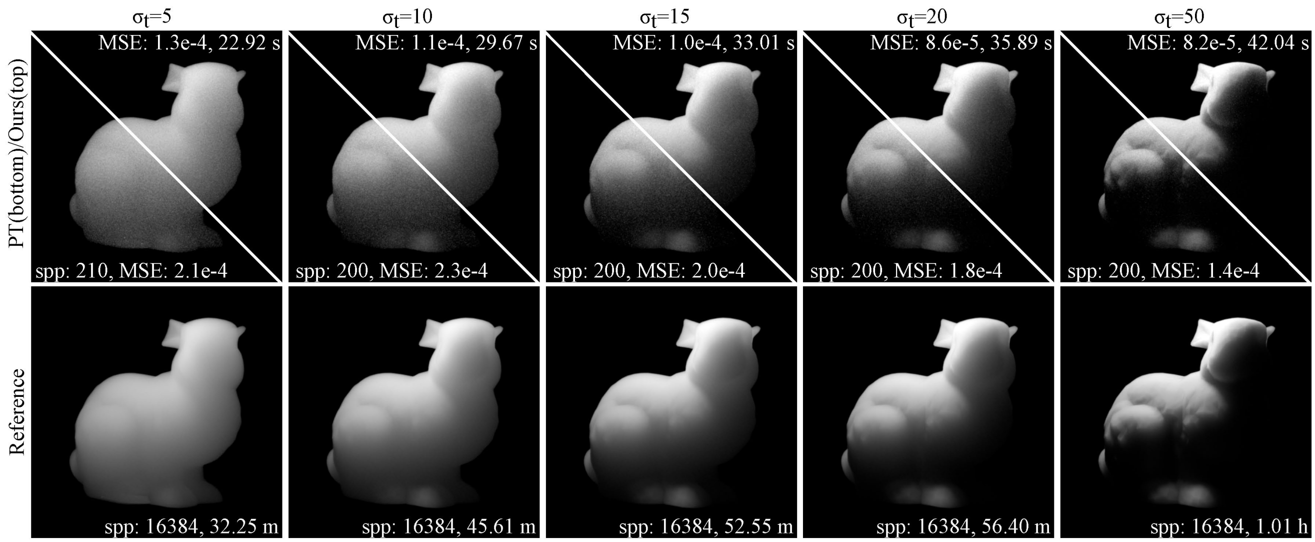


Figure 4. Comparison between our method and volumetric path tracing (equal-time) on media with varying extinction coefficients $\sigma_t \in \{5, 10, 15, 20, 50\}$. Our method produces better results than path tracing consistently. The sample rate of the source scene is set as 256.

$x_{k-1} \leftrightarrow x_k$:

$$w_i = \begin{cases} \alpha\sigma_t T_r(x_{i-1} \leftrightarrow x_i) & \text{if } \ x_i \in \text{medium}, \\ T_r(x_{i-1} \leftrightarrow x_i) & \text{if } \ x_i \in \text{surface} \end{cases} \tag{7}$$

and g:

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

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

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

4.2. Temporal gradient denoising

With the rendered color and the gradient image for the source albedo, we find that the gradient image suffers from a large amount of noise, as shown in Figure 2. Thus, we propose to perform temporal denoising on the gradient images since it is common to render a sequence rather than a single image.

First, we reproject every pixel of the current frame's neighboring frames to its matching pixel with the motion vector, and then we blend them to get the denoised value. The reprojection function for pixel i is defined:

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

where d is the motion vector, v_d is the velocity and t is the time.

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

where k is the distance (number of frames) between the image in the sequence and the target image ($k = 0$), and G_k is the gradient image with distance k. w_a is a smooth factor, which is set as $w_a = 0.1$ in all the test scenes. N is the number of frames used for denoising and is set as 11 in our implementation. Note that any denoising (not only temporal) could be used for the gradient image.

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

4.3. Participating media regularization

With the source pixel value R_s and the gradient G_s at the source albedo α_s, we need to predict the pixel value R_t with the target albedo α_t. Two typical prediction functions

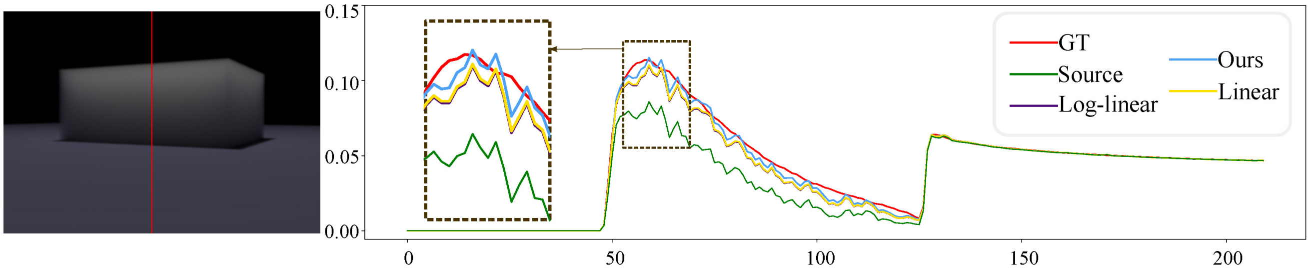


Figure 5. Radiance variation curves on a slice of pixels (the locations are shown as the red line at the left image) of the ground truth(GT), predicted results of linear, log-linear, our prediction functions, and the source image. The source $\alpha_s = 0.95 \times \alpha_t$ and target $\alpha_t = 0.95$. Our prediction function is the closest to the ground truth.

are the linear and the log-linear functions, as shown in Fan *et al*. [15]. However, they cannot be used in the albedo prediction, as the behavior of the media's albedo is complex. Therefore, we propose a novel prediction function, which includes two exponential functions for extrapolation.

Our prediction function is defined by multiplying two exponential functions:

$$R_t = R_s + G_s\beta \left(e^{(\alpha_t - \alpha_s)/\beta} - 1\right) e^{-\gamma R_s (\alpha_t - \alpha_s)^2}. \quad (11)$$

The definitions of β and γ are as follows:

$$\beta = \frac{c_\beta}{\sigma_t \alpha_s}, \;\; \gamma = \frac{\ln{(\sigma_t + 1)}^5}{1 - \alpha_s}, \quad (12)$$

where c_β ($0 < c_\beta \leq \sigma_t \alpha_s$) is a constant. This constant is used to scale the mfp, i.e. σ_t^{-1}. Thus, our prediction function can handle various models with different scales and shapes. We always set c_β as 0.5 in practice.

This definition is based on the following observations: first, the pixel value grows with the increasing albedo, and the growth speed increases at the same time; second, the growth speed is faster in the high-albedo and the small mfp cases than in the low-albedo and the large mfp cases. Therefore, we introduce the β (Eq. 12) to the first exponential term in Eq. 11 to ensure these two characteristics. However, using this exponential term alone causes some problems. In the high-albedo and small mfp case, the β is close to 0, and then the value multiplies to G_s will be significantly high. The small differences of G_s among neighboring pixels will be significantly amplified, leading to discontinuity among neighboring pixels in the predicted result. To solve this issue, we include another exponential term in Eq. 11. This term can alleviate the amplification of the first exponential term, depending on the albedo and the extinction coefficient in the γ term (Eq. 12). When the albedo is close to 1, and the mfp is small, we need the value of the γ term to be high enough to control the high value caused by the first exponential term. Then, we set $1 - \alpha_s$ as the denominator in the γ term. The growth speed becomes faster as the extinction coefficient increases. The speed of alleviation should be slower at the same time. Then, we set the γ term as a logarithmic function w.r.t. the extinction coefficient. We also introduce R_s in Eq. 11 to keep the smoothness among the neighboring pixels.

Our prediction function works for a wide range of materials, from large mfp to small mfp, as shown in Figure 4. We also compare our prediction function against the linear and the log-linear prediction function in Figure 5, and our prediction function better fits the ground truth.

5. Results

We have implemented our algorithm inside Mitsuba renderer [35]. We compare our method against volumetric path tracing (PT) with equal time and use converged PT as ground truth. All timings in this section are measured on an Intel i7-10700@2.90GHz (16 cores) with 16GB of main memory. We use Mean Squared Error (MSE) to measure the difference with the reference. In this paper, we focus on participating media with the high albedo, and the media configurations are shown in Table 1, and we use Henyey–Greenstein phase function in our scenes. For convenience, if all channels of the albedo or the extinction coefficient have the same value, we will only show one value.

In the following, by *source* we mean the rendered result with the low albedo, without prediction, while by *target* we mean the final predicted result with the desired albedo.

5.1. Quality validation

In Figure 3, we compare our method with the temporal denoised volumetric path tracing and the volumetric path tracing with equal rendering time. This temporal denoising method is the same as our gradient denoising method. We also show the error maps between our method and the reference and between path tracing and the reference. By comparison, our method produces less error than denoised path tracing and path tracing without denoising both visually and quantitatively. Moreover, we also find that applying temporal denoising to path tracing cannot perform better than path tracing with equal time. These three scenes include

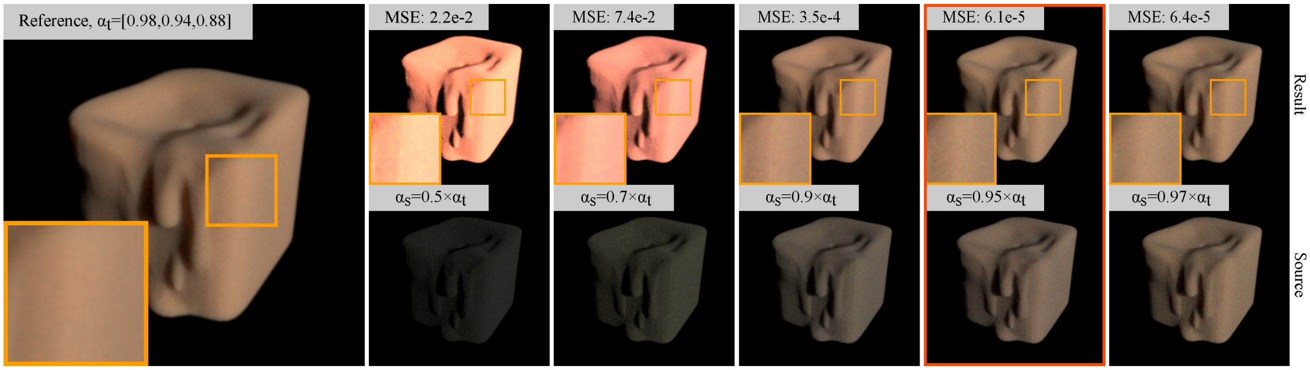


Figure 6. Predicted results (top) from different sources (bottom) on the Candle scene. When the source albedo is far from the target albedo, there is obvious color bias after the prediction.

Table 1. Media configurations, scene settings, computation time, and Mean Squared Errors (MSE) for our test scenes. g is the mean cosine of the phase function. Spp. represents sample per pixel for path tracing. All timings are in minutes.

Scene	Media			Resolution	Ours				Pt.			Reference	
	g	σ_t	α_t		Factor	Spp.	Time	MSE	Spp.	Time	MSE	Spp.	Time
Bunny	0.7	10	0.95	500 × 500	0.95	256	0.49	**1.1e-4**	200	0.50	2.3e-4	16K	45.61
Candle	0.8	10	[0.98, 0.94, 0.88]	256 × 256	0.95	2048	1.21	**6.1e-5**	1300	1.17	1.3e-4	16K	15.89
Lucy	0.8	10	0.97	512 × 512	0.93	512	1.14	**1.5e-5**	350	1.16	4.4e-5	16K	60.00

both simple shapes (Bunny scene and Candle scene) and complex shapes (Lucy scene). In both cases, our method outperforms path tracing.

In Figure 4, we compare our method against path tracing with equal time on the Bunny scene with varying 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 our prediction function are both robust.

The source images and the denoised gradient images in the prediction step used in the above scenes are shown in the supplementary material.

5.2. Parameter analysis

Prediction function is a key component in our method. To validate its benefits, we compare it against the linear and the log-linear models on the Cube scene in Figure 5. 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. By comparison, our prediction function produces the closest result to the reference.

Gradient denoising also significantly impacts the results. In Figure 2, we compare the results with and without gradient denoising. By comparison, we find that the temporal denoising reduces the noise on the gradient image and further improves the rendered quality result.

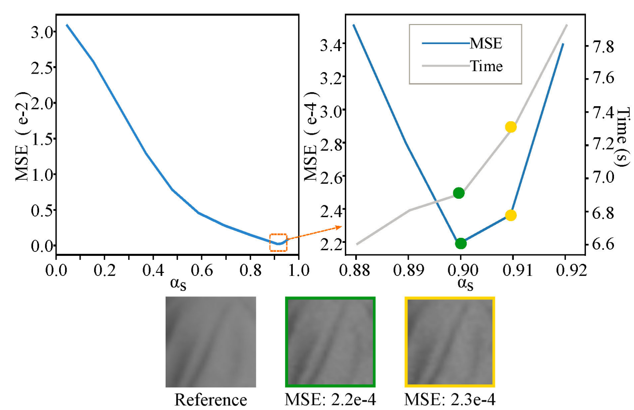


Figure 7. The error (MSE) curve as a function of source albedo α_s on the Lucy scene. The α_s ranges from 0.1 to 0.94 ($0.1 - 0.97\times$ α_t). The sample rate of the source scene is set as 512. The errors are measured on cropped images (30×30). The error curve and the time cost curve (rightmost) are 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.

Choice of source albedo. The choice of the source albedo affects the final image quality significantly. In Figure 7, we show 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 the error curves at this range ($0.9 - 0.95 \times \alpha_t$).

In Figure 6, we show both the source image and the tar-

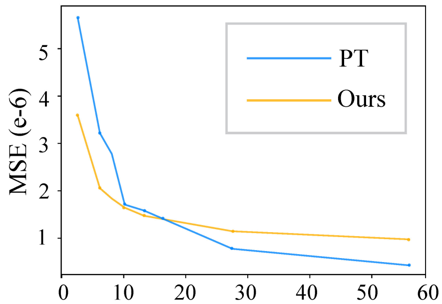


Figure 8. MSE between our method or path tracing with the reference (rendered with 33,768 spp) over varying rendering time on the Cube scene. The error of our method decreases until approximately 16 s and then keeps almost constant. The error before 16 s comes from both variance and bias. After the result is converged, the error comes from bias. Therefore, our method is suitable for rendering with a low time budget.

get 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. When increasing the source albedo, the bias decreases while the variance increases. Thus, the best source albedo is around $0.9 - 0.95 \times \alpha_t$.

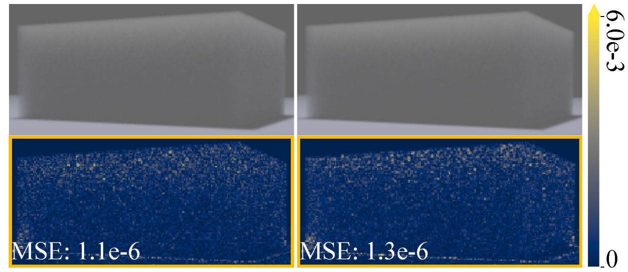
Considering the accuracy and the time cost, the source albedo should be chosen in $0.9 - 0.95 \times \alpha_t$ in practice.

5.3. Performance measurement

In Table 1, we report all the scene-setting, computation time, and their error (MSE) with the reference rendering results of our test scenes. The cost of prediction and denoising is negligible compared to rendering time (0.2s for a 512×512 image). Thus we ignore it in our computation time. Our method produces higher-quality results than volumetric path tracing with equal time in all scenes quantitatively. We use 11 frames for gradient denoising for all our scenes. As we can see from the table, our method has a higher sample rate than the equal-time path tracing, as 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, our overall error (bias and variance) still goes down. Our method targets renderings with a low time budget. When the sample rate is high, our result will have a larger error than path tracing, as shown in Figure 8. Our method is designed for high-albedo media and does not benefit the media with low albedo, as shown in Figure 9.



PT, spp: 320, 2.57 s   Ours, spp: 256, 2.60 s

Figure 9. 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 the target $\alpha_t = 0.5$.

6. Conclusion

We have presented a novel differentiable regularization framework to improve rendering quality in participating media rendering. Our method is simple and is suitable for any high-albedo homogeneous participating media. Our method produces fewer noise results than path tracing, although it will 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. Then, with our method's denoised and prediction steps, the noise can be further reduced.

In the future, we are interested in improving 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.

References

[1] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Computer Graphics (ACM SIGGRAPH '98 Proceedings)*, pages 311–320, 1998. r. 1

[2] W. Jarosz, D. Nowrouzezahrai, R. Thomas, P.-P. Sloan, and M. Zwicker. Progressive photon beams. *ACM Transactions on Graphics*, 30(6):181:1–181:12, December 2011. 1

[3] W. Jarosz, M. Zwicker, and H. W. Jensen. The beam radiance estimate for volumetric photon mapping. *Comput. Graph. Forum (proc. Eurographics)*, 27(2):557–566, Apr. 2008. 1

[4] J. Krivanek, I. Georgiev, T. Hachsuki, P. Vevoda, M. Sik, D. Nowrouzezahrai, and W. Jaroz. Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics*, 33(4), July 2014. 1

[5] B. Bitterli and W. Jarosz. Beyond points and beams: Higher-dimensional photon samples for volumetric light

transport. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 36(4), July 2017. 1

[6] X. Deng, S. Jiao, B. Bitterli, and W. Jarosz. Photon surfaces for robust, unbiased volumetric density estimation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 38(4), 2019. 1

[7] J. T. Kajiya and B. P. V. Herzen. Ray tracing volume densities. In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 165–174, July 1984. 1

[8] E. P. Lafortune and Y. D. Willems. Rendering participating media with bidirectional path tracing. In *Rendering Techniques '96 (Proceedings of the Seventh Eurographics Workshop on Rendering)*, pages 91–100. Springer-Verlag/Wien, 1996. 1, 2

[9] M. Pauly, T. Kollig, and A. Keller. Metropolis light transport for participating media. In B. Peroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*, pages 11–22. Springer Wien, 2000. 1, 2

[10] H. Deng, B. Wang, R. Wang, and N. Holzschuch. A practical path guiding method for participating media. *Computational Visual Media*, 6, 02 2020. 1

[11] S. Herholz, Y. Zhao, O. Elek, D. Nowrouzezahrai, H. P. A. Lensch, and J. Krivanek. Volume Path Guiding Based on Zero-Variance Random Walk Theory. *ACM Transactions on Graphics*, Mar. 2019. 1, 2

[12] J. Křivánek and E. d'Eon. A zero-variance-based sampling scheme for monte carlo subsurface scattering. In *ACM SIGGRAPH 2014 Talks*, SIGGRAPH '14, pages 66:1–66:1, 2014. 1, 2

[13] J. Meng, J. Hanika, and C. Dachsbacher. Improving the dwivedi sampling scheme. *Computer Graphics Forum*, 35(4):37–44, 2016. 1, 2

[14] B. Wang, L. Ge, and N. Holzschuch. Precomputed multiple scattering for light simulation in participating medium. *IEEE Transactions on Visualization and Computer Graphics*, 26(7), 2020. 1, 2

[15] J. Fan, B. Wang, W. Wu, M. Hasan, J. Yang, and L.-Q. Yan. Efficient specular glints rendering with differentiable regularization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2022. 2, 6

[16] W. Wu, B. Wang, and L.-Q. Yan. A survey on rendering homogeneous participating media. *Computational Visual Media*, 8:177–198, 06 2022. 2

[17] H. E. Rushmeier. Realistic image synthesis for scenes with radiatively participating media. 1988. 2

[18] P. Weber, J. Hanika, and C. Dachsbacher. Multiple vertex next event estimation for lighting in dense, forward-scattering media. *Computer Graphics Forum*, 36(2):21–30, 2017. 2

[19] A. Kaplanyan and C. Dachsbacher. Path space regularization for holistic and robust light transport. *Computer Graphics Forum*, 32, 05 2013. 2

[20] G. Bouchard, J.-C. Iehl, V. Ostromoukhov, and P. Poulin. Improving robustness of monte-carlo global illumination with directional regularization. In *SIGGRAPH Asia 2013 Technical Briefs*, New York, NY, USA, 2013. Association for Computing Machinery. 2

[21] J. Jendersie and T. Grosch. Microfacet Model Regularization for Robust Light Transport. *Computer Graphics Forum*, 2019. 2

[22] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 2

[23] G. Loubet, N. Holzschuch, and W. Jakob. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Trans. Graph.*, 38(6), nov 2019. 2

[24] C. Zhang, L. Wu, C. Zheng, I. Gkioulekas, R. Ramamoorthi, and S. Zhao. A differential theory of radiative transfer. *ACM Trans. Graph.*, 38(6), nov 2019. 3

[25] C. Zhang, B. Miller, K. Yan, I. Gkioulekas, and S. Zhao. Path-space differentiable rendering. *ACM Trans. Graph.*, 39(4), jul 2020. 3

[26] C. Zhang, Z. Yu, and S. Zhao. Path-space differentiable rendering of participating media. *ACM Trans. Graph.*, 40(4), jul 2021. 3

[27] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Trans. Graph.*, 38(6), nov 2019. 3

[28] M. Nimier-David, S. Speierer, B. Ruiz, and W. Jakob. Radiative backpropagation: An adjoint method for lightning-fast differentiable rendering. *ACM Trans. Graph.*, 39(4), jul 2020. 3

[29] S. Zhao, W. Jakob, and T.-M. Li. Physics-based differentiable rendering: From theory to implementation. In *ACM SIGGRAPH 2020 Courses*, SIGGRAPH '20, 2020. 3

[30] C. Zhang, Z. Dong, M. Doggett, and S. Zhao. Antithetic sampling for monte carlo differentiable rendering. 40(4), jul 2021. 3

[31] M. Hašan and R. Ramamoorthi. Interactive albedo editing in path-traced volumetric materials. *ACM Trans. Graph.*, 32(2), apr 2013. 3

[32] S. Zhao, R. Ramamoorthi, and K. Bala. High-order similarity relations in radiative transfer. *ACM Trans. Graph.*, 33(4), jul 2014. 3

[33] S. Chandrasekhar. *Radiative transfer*. Dover publications, New York, 1960. 3

[34] J. T. Kajiya. The rendering equation. *SIGGRAPH Comput. Graph.*, 20(4):143–150, 1986. 3

[35] W. Jakob. Mitsuba renderer, 2010. `http://www.mitsuba-renderer.org/`. 6