Real-time Realistic Volume Rendering of Consistently High Quality with Dynamic Illumination

Chunxiao Xu, Haojie Cheng, Zhenxin Chen, Jiajun Wang, Yibo Chen and Lingxiao Zhao

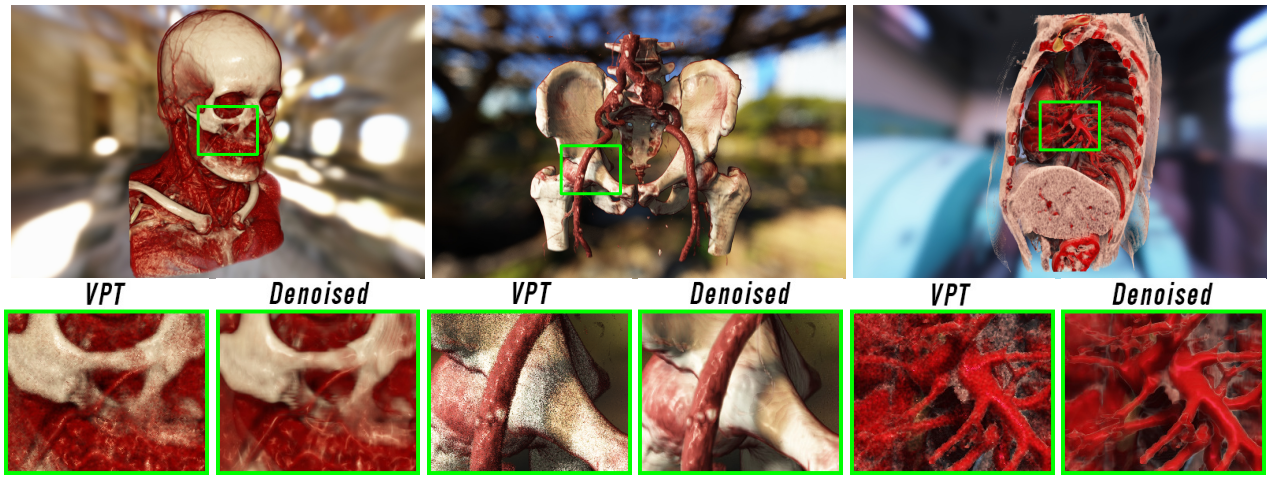


Fig. 1. The noisy shading results obtained directly using our optimized VPT-based DVR technique are labeled as *VPT*. To generate *Denoised* DVR results of higher rendering quality, these shading results were further processed using our spatiotemporal denoiser. All images were rendered using an Nvidia GTX 1080Ti graphics card with 25 samples per pixel. These results demonstrate a rendering frame rate of at least 40 frames per second.

Abstract—Direct Volume Rendering (DVR) plays an important role in scientific data visualization. To generate photo-realistic DVR results, the physical light transport throughout the volume is simulated by applying the Monte Carlo-based volumetric path tracing (VPT) approach. For real-time applications, due to the time constraint for rendering each frame, only a limited number of samples shall be taken for the computation per pixel. This can result in a significant amount of noise in the rendering results. This paper describes our optimized VPT sampling algorithm and a novel denoising technique to generate consistently high-quality realistic DVR results in real time. We develop a new shading model that can reduce estimation variance to enhance the quality of DVR results. Additionally, a hybrid acceleration structure is created by integrating both octree and macrocell to improve sampling efficiency. This allows the acquisition of sufficiently more shading samples while maintaining the desired interactive frame rate. To further eliminate remaining noise and improve temporal stability of DVR results, we develop a novel spatiotemporal denoising framework. Our denoiser decouples the estimated radiance into high-detail low-noise and low-detail high-noise components. Different denoising algorithms are separately applied to these components to reduce noise without introducing blurring artifacts. Our DVR system can consistently offer high rendering quality and good temporal stability across DVR result frames in real time. During fast user interactions and with rapid alterations of the illumination condition, our rendering method can still provide good visual comfort and representation accuracy without visible latency.

Index Terms—Realistic volume rendering, volumetric path tracing, dynamic global illumination, real-time denoising.

1 Introduction

Direct Volume Rendering (DVR) is normally a computationally intensive task for visualizing volumetric scientific data [1, 2, 3]. Instead of extracting and constructing explicit mesh representations, DVR techniques rely on determining the implicit representation of the 3D volumetric object on the fly during rendering. Based on the strategy of physically-based rendering (PBR) [4], volumetric path tracing (VPT) [5] can be used to simulate the light transport through the volume and generate photo-realistic DVR results. In recent years, VPT-based DVR methods have been attracting more and more research interests [6, 7, 8, 9].

A VPT method involves the sampling along light paths that originate from the camera and traverse the volume space. A stochastic light scattering process is simulated based on the transfer function that maps volume density values to material attributes. Monte Carlo (MC) method is typically employed during the volume space sampling to estimate the scattering events and illumination effects in accordance with the rendering equation [6, 10]. A large number of samples are usually required to generate high-quality rendering outcomes with minimal noise.

To ensure a high rendering efficiency for real-time applications, only a limited number of samples per pixel (spp)

- *Chunxiao Xu, Haojie Cheng and Lingxiao Zhao are with University of Science and Technology of China, Hefei, China.*
- *Chunxiao Xu, Haojie Cheng, Zhenxin Chen, Jiajun Wang, Yibo Chen and Lingxiao Zhao are with Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, Suzhou, China. E-mail: hitic@sibet.ac.cn*

should be used per frame to reduce the computational expense. However, this can introduce remarkable noise. To resolve this conflict, real-time denoising techniques have been suggested to postprocess the MC-based rendering results [11, 12, 13, 14]. These techniques are mainly used for surface model rendering and noiseless geometry buffers (G-buffers) are used to preserve details. However, G-buffers obtained in VPT can be noisy due to varying free-flight distances for neighboring pixels [8, 15]. Noisy G-buffers make it more challenging to guide the VPT denoising process.

In recent years, some interactive denoising methods have been proposed for realistic rendering using VPT [8, 9, 15]. These techniques can help improve the rendering result quality of VPT methods, given only a few spp. However, overblurring artifacts can be generated in the denoised results, and annoying temporal flickering noise can be clearly visible during user interaction. These interferences can significantly disrupt human visual perception and cause visual discomfort for users during scientific data visualization.

Our research aims to generate noise-free and temporally stable results using VPT-based DVR in real time. We try to maintain high-quality rendering outcomes even in scenarios involving intricate interactions, such as swift virtual camera movements and dynamic lighting conditions. Through an analysis of the limitations present in current real-time VPT methods (Sec. 3), it can be speculated that optimizing both shading algorithms and denoising strategies [16] can lead to optimal rendering results of consistently high quality during user interaction. Our main contributions are outlined as follows:

- We develop a new VPT-based DVR framework that employs both an optimized path sampling protocol and a new denoising strategy. It can provide consistently high image quality of realistic DVR results during user interaction in real time (Fig. 1).
- We develop a *surface-medium coexistence* shading model, the shading estimation results of which align with the expected values of [6] while exhibiting a reduced estimation variance.
- A new spatiotemporal denoising framework is proposed to enhance the image quality. It decouples the radiance sample into high-detail-low-noise and low-detail-high-noise components. Different denoising algorithms are applied to process these two components respectively. Our denoiser can be used to effectively reduce the flickering DVR noise without introducing blurring artifacts.

In addition to the aforementioned contributions, our DVR framework employs an acceleration structure that combines octree [17] and macrocell [18], facilitating empty space skipping and volume traversal using the 3D-DDA algorithm [4, 19]. A demo video clip and an executable program with graphical user interface (GUI) are provided as supplementary materials to demonstrate the effectiveness of our prototyping volume visualization framework.

2 Related Work

In this section, related advanced illumination techniques for DVR and denoising approaches for MC-based path tracing are surveyed.

2.1 DVR methods

Light transfer equation (LTE) describes the process of light scattering, accumulation and absorption in the participating media [1, 4]. Many DVR techniques have been developed to simulate or approximate this process. Existing DVR techniques with advanced illumination are briefly surveyed in following paragraphs.

Volumetric illumination. A comprehensive survey [20] enumerates many significant works about advanced illumination techniques for DVR. We supplement with some more recent research. Based on volumetric photon mapping (VPM), Jönsson et al. [21] developed a method for identifying the unchanged photons during user interaction, thereby significantly enhancing photon reusability. Igouchkine et al. [22] introduced a material transfer function that facilitates high-quality rendering of multi-material volumes by generating surface-like behavior at structure boundaries and volume-like behavior within the structures. Magnus et al. [23] accomplished real-time refraction and caustics in DVR by interleaving the propagation of light and camera rays. Their method eliminates the need of memory-intensive storage of lighting information and precomputation. Bauer et al. [24] introduced photon field networks, a neural representation enabling interactive rendering with reduced noise.

MC based VPT. A comprehensive survey about MC-based volumetric lighting estimation can be found in [5]. Biased ray marching [25] and unbiased null-collision techniques [5, 26, 27, 28] were used to sample the light path and estimate transmittance. Physically-based VPT methods can generate highly realistic rendering results [6, 7]. To achieve optimal visual effects, HDR environment maps can be utilized to illuminate the volume. To effectively evaluate environment illumination, Radziewsky et al. [10] proposed an algorithm that can quickly build the approximate light visibility structure, which is then used for the purpose of importance sampling. Lesar et al. [29] have implemented VPT using WebGL, enabling cross-platform applications. VPT-based cinematic rendering [3, 30] plays an increasingly vital role in medical imaging visualization. Denisova et al. [31] proposed a DVR system that supports multiple scattering. They defined various materials and assigned different material types to voxels based on their densities. Wu et al. [32] introduced an efficient method for volumetric neural representation, achieving real-time VPT through a sampling stream algorithm.

2.2 Denoising approaches

Denoising methods have been suggested as a helpful addition for improving the quality of raw rendering results, while a preferable rendering speed can be maintained. For an in-depth review of reconstruction techniques for MC rendering, please refer to [33, 34]. An effective denoiser should not only filter noise out from the rendering results, but also guarantee the temporal stability of consecutive frames [12, 13].

Denoising approaches for surface model rendering. There are mainly three classes of denoising approaches for MC-based path tracing: guided kernel filtering approaches, fitting-based approaches and deep learning-based approaches. Most of these techniques rely on the use of

noiseless auxiliary buffers (G-buffers). Zimmer et al. [35] decoupled the rendering outcome into distinct components and estimated the temporal motion of these components separately. Subsequently, individual denoising processes were applied to each component, aiming to enhance the robustness of temporal denoising. Schied et al. [11, 12] employed a variance-guided extended kernel filter, which enables the real-time attainment of high-quality denoised results. The temporal gradients of samples are calculated to better reject outdated previous DVR result. Moon et al. [36] proposed a fitting-based method that used a linear model to estimate pixel values. Multiple adjacent pixels share the same linear model that performs the filtering. Bitterli et al. [37] used nonlinearly weighted first-order regression with auxiliary buffers to denoise samples. Koskela et al. [38] proposed a block-wise feature regression method, employing linear models to fit neighborhood features, aiming to achieve the denoised outcome.

Deep learning denoisers can be classified into two categories. Some methods, e.g. the recurrent denoising autoencoder (RAE) [13], make use of networks that take noisy images as the input and then output denoised images. Other approaches [14, 39, 40, 41] replace a part of a traditional denoiser with a network. For example, methods proposed in [14, 39, 40] apply convolutional neural networks (CNN) to train parameters of filtering kernels. Denoisers [13, 14, 42, 43] based on lightweight neural network architectures can achieve real-time denoising. Real-time deep learning denoisers often struggle to generalize effectively, especially in applications that involve complex interactions [34].

Denoising approaches for VPT. Martschinke et al. [44] proposed an approach to enable interactive VPT-based DVR by leveraging temporal antialiasing principles, allowing for high-quality rendering results with minimal noise and applicability in virtual reality. Iglesias et al. [8] recently developed a linear fitting-based VPT denoising method that used the weighted recursive least square (WRLS) to update model parameters. Hofmann et al. [9] utilized the noisy G-buffer obtained from sampling as auxiliary features for neural denoising. It is challenging to preserve details in their method, as described in Sec. 3. Hofmann et al. [15] used a deep learning denoiser that has a U-net architecture to denoise rendering results of participating media. Additionally, they used a separate small neural network to predict the sample density for adaptive sampling. Their denoising framework can be adopted in sparse volume rendering, while scientific visualization requires a higher fidelity of details. Bauer et al. [45] proposed FoVolNet, a foveated rendering technique that sparsely samples the volume around a focal point and reconstructs the full frame using a neural network. Taibo et al. [46] proposed an immersive 3D medical visualization system that uses extended linear regression denoising to enable real-time cinematic rendering in AR/VR.

3 Problem Analysis

Rendering results generated using VPT-based DVR typically exhibit a significantly higher noise level compared to surface model rendering. There are two main reasons of this. First, in VPT sampling, MC estimation is required for sampling both free path length and the scattering distribution function (SDF). Second, the calculation of light visibility in VPT includes an estimation of transmittance [27], which leads to increased noise levels in shading results. These two factors can introduce variance into the rendered results. Achieving high-quality denoised outcome becomes challenging when denoising algorithms are applied to raw renderings with high levels of noise. For spatial denoising, it is noteworthy that attempts to remove noise from highly noisy shaded results may generate over-blurring artifacts. Finding the temporal correlation between adjacent noisy rendered frames is also challenging. The temporal flickering noise is therefore hard to appropriately eliminate.

There are two main aspects that should be considered for the noise reduction of raw DVR results. The first one is to use optimized shading strategy that provides smaller DVR result variance. The second one is to enhance VPT sampling efficiency for maximizing the sampling density per frame without compromising the rendering speed, such as using the acceleration structure. Our VPT-based DVR method takes both of these aspects into account to improve the quality of the DVR results.

In VPT-based DVR, path sampling usually has considerable uncertainty and there is no noise-free G-buffers for guiding denoising. Existing denoising methods are primarily designed for surface model rendering and have challenges when applied to scenes that involve participating media. For rendering surface models, the radiance obtained during shading estimation can be split into albedo and illumination [11]. Since the albedo shows no MC noise, only the illumination needs to be denoised, and the denoising output shall be obtained by multipling albedo with denoised illumination. Even if some blurring artifacts are introduced to illumination component, the impact on the overall quality of the denoised output will be very limited, especially in a scene with rich surface texture information. For VPT-based DVR, it is crucial to devise more efficient denoising methods to prevent the introduction of blurring artifacts.

Additionally, several other issues should be addressed. First, in DVR, volumes are often considered to contain translucent structures. Temporal ghosting artifacts can easily be introduced into the denoised results during the temporal denoising. Second, while adjusting lighting conditions to highlight volumetric structures, the resulting variations in shadows and illuminations from user-modified light sources facilitate the spatial and depth perception of the user. Enhancing the rendering quality, particularly with dynamic lighting, can contribute to user decision-making during interactions. In more challenging applications, such as observing rendered results using head-mounted display (HMD) devices, the camera viewpoint constantly changes, resulting in the continuous variation of shading. The technique proposed in this paper presents promising capabilities to address these issues.

4 VPT Sampling with Less Variance

This section begins by outlining the formulas related to LTE. Subsequently, our *surface-medium coexistence* model that effectively minimizes variance in shading results is introduced. In addition, we delineate the acceleration structures

utilized in our renderer to facilitate rapid volumetric space traversal and empty space skipping. The symbols utilized to denote the LTE are detailed in Table 1. The Bidirectional Reflection Distribution Function is abbreviated as BRDF.

TABLE 1: Notations of the LTE

Symbol	Description
$\mathbf{x}$	a point in volume space
ω	direction vector of light propagation
$\mu_a(\mathbf{x})$	absorption coefficient at point $\mathbf{x}$
$\mu_s(\mathbf{x})$	scattering coefficient at point $\mathbf{x}$
$\mu_t(\mathbf{x})$	extinction coefficient at point $\mathbf{x}$
$L(\mathbf{x},\omega)$	radiance at point $\mathbf{x}$ with the direction ω
$Li(\mathbf{x},\omega)$	in-scattering radiance at point $\mathbf{x}$ with direction ω
$T_r(\mathbf{x},\mathbf{y})$	transmittance between $\mathbf{x}$ and $\mathbf{y}$
$g(\mathbf{x})$	gradient at point $\mathbf{x}$
$\rho(\omega,\mathbf{x},\omega')$	phase function from ω' to ω at point $\mathbf{x}$
$f(\omega,\mathbf{x},\omega')$	BRDF from ω' to ω at point $\mathbf{x}$
$\mathcal{S}_{4\pi}$	the set of all directions
rand()	random number in the range of 0 to 1

4.1 LTE in participating media

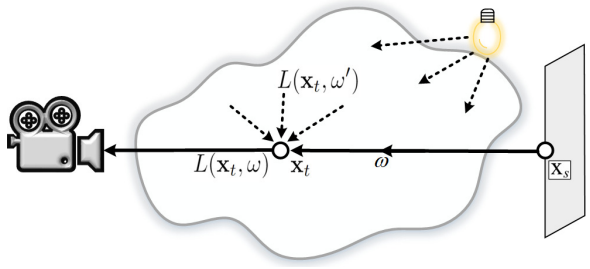


Fig. 2. In non-luminescent participating media, the standard light transport procedure involves scattering and absorption effects.

When considering solely light absorption and scattering, the LTE can be represented by Equ. 1 and depicted in Fig. 2, where $\mathbf{x}_t = \mathbf{x} - t\omega$. The upper limit of the integral, denoted as s, represents the distance from point $\mathbf{x}$ to the surface of the volume bounding box along the direction of $-\omega$.

$$L(\mathbf{x},\omega) = T_r(\mathbf{x},\mathbf{x}_s)L(\mathbf{x}_s,\omega) + \int_0^s T_r(\mathbf{x},\mathbf{x}_t)\mu_s(\mathbf{x}_t)Li(\mathbf{x}_t,\omega)\mathrm{d}t \quad (1)$$

In-scattering radiance $Li(\mathbf{x}_t,\omega)$ and light transmittance $T_r(\mathbf{x},\mathbf{x}_t)$ can be described as :

$$Li(\mathbf{x},\omega) = \int_{\mathcal{S}_{4\pi}} L(\mathbf{x},\omega')\rho(\omega,\mathbf{x},\omega')\mathrm{d}\omega' \quad (2)$$

$$T_r(\mathbf{x},\mathbf{x}_t) = \exp\big(-\int_0^t \mu_t(\mathbf{x}+s\omega)\mathrm{d}s\big) \quad (3)$$

where $\mu_t(\mathbf{x}) = \mu_a(\mathbf{x}) + \mu_s(\mathbf{x})$.

The solution to the LTE involves two crucial techniques: the estimation of transmittance and the sampling of scattering positions. Since Galtier et al. [26] re-derived Monte Carlo-based volume rendering methods from the perspective of radiative transfer, a plethora of related techniques, collectively termed the null-scattering methodology [5, 27, 28, 47], has emerged. Within this methodology, ratio tracking is a commonly employed and effective method for estimating transmittance [4, 27], and is adopted in our DVR approach for transmittance estimation. Additionally, we employ the null-collision technique [26] to sample scattering positions.

4.2 *Surface-medium coexistence* model

In scientific visualization, numerous methods enhance realisticness and expressiveness by revealing implicit surface structures within volumetric data [20]. Kroes et al. [6, 48] demonstrated that incorporating shading with BRDFs and phase functions can significantly enhance the realisticness of rendering results. Their method has been widely adopted in subsequent research [3, 8, 10, 49]. Recognizing the opportunity for improving their shading computation approach, we develop our shading model, the *surface-medium coexistence* model exhibits reduced estimation variance and maintains the same expectations as their method. In the following paragraphs, we present an overview of their shading scheme, and introduce our shading model.

In ExposureRenderer (ER) [48], two kinds of particles are defined in the volume space. One exhibits surface reflection characteristics, and the other exhibits volume scattering characteristics. Specifically, particles of the surface reflection type are used to simulate the effects of reflected light from implicit surfaces within the volumetric data, and particles of the volumetric scattering type are used to model the subsurface scattering effects. The proportions of particles contributing to surface reflection and volume scattering at a point $\mathbf{x}$ are denoted by $P_{brdf}(\mathbf{x})$ and $P_{phase}(\mathbf{x})$, respectively. The calculation of these proportions is elaborated in [48].

The volumetric path tracing process can be summarized as follows: (1) sampling free path length, (2) calculating direct illumination at the real scattering position, and (3) sampling new scattering directions and updating path throughput [4]. In ER, when a real scattering event occurs at $\mathbf{x}$, the type of the particle with which the ray has collided is determined as follows: (1) sampling a uniform random number r_i between 0 and 1, and (2) if r_i is less than $P_{\text{brdf}}(\mathbf{x})$, the collision is considered to have occurred with a surface scattering particle, denoted as event E_{brdf}. Conversely, if r_i exceeds $P_{\text{brdf}}(\mathbf{x})$, the collision is assumed to be with a volumetric scattering particle, denoted as event E_{phase}. The technique for estimating direct illumination through importance sampling in ER is defined by Equ. 4:

$$M_{ER}\big(Li(\mathbf{x},\omega)\big) = \begin{cases} T_r(\mathbf{x},\mathbf{y}_i)\cdot\frac{f(\omega,\mathbf{x},\omega')L(\mathbf{x},\omega')}{P_{brdf}(\mathbf{x})\cdot p(\mathbf{y}_i)} & \text{for } E_{brdf} \\ T_r(\mathbf{x},\mathbf{y}_i)\cdot\frac{\rho(\omega,\mathbf{x},\omega')L(\mathbf{x},\omega')}{P_{phase}(\mathbf{x})\cdot p(\mathbf{y}_i)} & \text{for } E_{phase} \end{cases} \quad (4)$$

where $\mathbf{y}_i$ is the sampled point of the light source, $p(\mathbf{y}_i)$ represents the corresponding sampling probability density and ω' denotes the direction from point $\mathbf{x}$ to $\mathbf{y}_i$. The sampling of $\mathbf{y}_i$ commonly employs the technique of multiple importance sampling (MIS) [6, 50].

The ER method still needs to be improved by addressing its inefficiency problem. To do this, we develop the *surface-medium coexistence* shading model proposed in this paper. Our model no longer differentiates between particle types. Instead, it assumes the presence of a single particle type, which exhibits both volumetric and surface scattering properties. The SDF describing the material attributes of this particle can be defined as Equ. 5.

$$h(\omega,\mathbf{x},\omega') = f(\omega,\mathbf{x},\omega')\cdot P_{brdf}(\mathbf{x}) + p(\omega,\mathbf{x},\omega')\cdot P_{phase}(\mathbf{x}) \quad (5)$$

Since there is only one type of particle present, when a real scattering event occurs, it is unnecessary to randomly select the type of particle that collides. The direct illumination estimation based on our model can be described as follows:

$$M_h\big(Li(\mathbf{x},\omega)\big) = h(\omega,\mathbf{x},\omega')\cdot T_r(\mathbf{x},\mathbf{y}_i)\frac{L(\mathbf{x},\omega')}{p(\mathbf{y}_i)} \quad (6)$$

Essentially, our model enhances the utilization rate of transmittance estimation.

VPT-based DVR commonly uses single-channel μ_t and multi-channel albedo to approximate color media [8, 48]. Let A_{brdf} represent the albedo in the BRDF, and A_{phase} denote the albedo acting on the phase function. The luminance of A_{brdf} is denoted by l_{brdf}, and the luminance of A_{phase} is denoted by l_{phase}. For sampling the new bounce direction of the camera ray in our model, the probability of sampling the BRDF $P_b(\mathbf{x})$ can be denoted as:

$$P_b(\mathbf{x}) = \frac{l_{brdf}(\mathbf{x})\cdot P_{brdf}(\mathbf{x})}{l_{brdf}(\mathbf{x})\cdot P_{brdf}(\mathbf{x}) + l_{phase}(\mathbf{x})\cdot P_{phase}(\mathbf{x})} \quad (7)$$

and the probability of sampling the phase function is $1 - P_b(\mathbf{x})$. Employing this sampling method can result in a probability distribution of sampled bounce directions that closely approximates the SDF distribution.

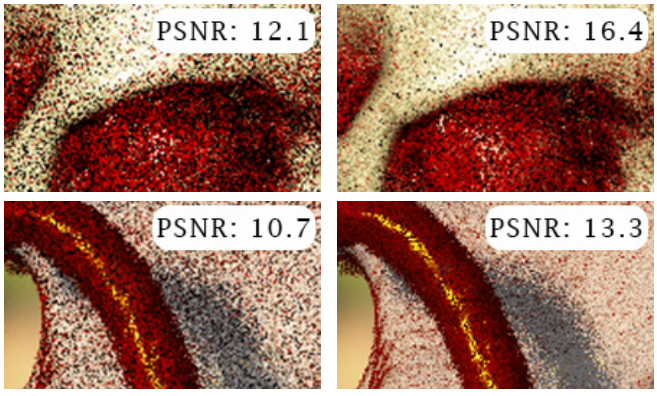


Fig. 3. The images are all rendered with 3 spp. *ER* represent the shading results using the approach of [6], and *Ours* is the results using our *surface-medium coexistence* model. The PSNR values quantify the error between these images and reference images rendered at 1024 spp. For further comparative results, please refer to supplementary material #1.

The shading estimation results obtained using our model has an equivalent expectation to ER but with reduced variance. The proof is provided in supplementary material #1. As shown in Fig. 3, with the same spp, images rendered using our model have less noise. Moreover, the additional time cost incurred by calculating multiple SDF values can be disregarded compared to the time cost required for estimating transmittance. Therefore, for estimating one shading sample, our model does not have a significant increase in time compared with ExposureRenderer.

4.3 VPT acceleration using octree and macrocell

Sampling efficiency can be improved via acceleration structures. Examples of popular accelerators include the octree [17] and Volume Dynamic B+Tree (VDB) [51]. We implement a hybrid acceleration structure based on macrocell and octree. The octree proves advantageous for efficiently skipping empty regions, while the 3D-DDA sampling technique [19] based on macrocell facilitates swift traversal in the volumetric space.

The spatial partitioning at different levels of octree used in our renderer is illustrated in Fig. 4. Each partition block represents an octree node. All nodes are sequentially stored in a linear structure in a spatial order. Each node encompasses indices for the parent node and all child nodes, along with the voxel value range within its bounding box. In response to changes in the transfer function, updates to the maximum extinction coefficient μ_{max} in each node are necessary based on the voxel value range stored in the node. The μ_{max} is utilized for VPT sampling [4, 5, 26] and if this value is zero within a node, the node can be skipped.

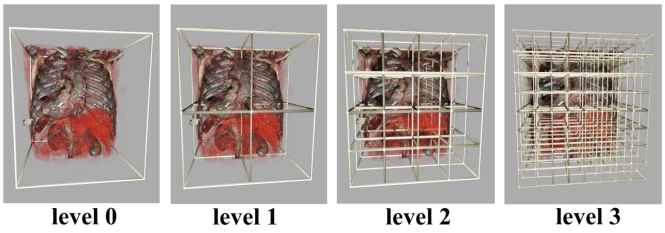


Fig. 4. The figure depicts the spatial partitioning for the volume at different levels. For the ease of localization within each level, the bounding boxes of all nodes in a level are of uniform size.

The acceleration structure is not the primary focus of our paper, and various optimizations constitute low-level implementation details. We provide a thorough account of our accelerator in supplementary material #2 to maintain the comprehensiveness of the system description.

5 Image-space DVR Denoising

A denoising operation is necessary to further reduce noise and enhance the temporal stability of the VPT-based DVR results. An overview of our denoising framework is outlined in Fig. 5. In our denoising method, a radiance sample obtained in the shading estimation is decoupled into two components. These two components are denoised separately using our spatiotemporal denoiser.

5.1 Decoupling of radiance

The radiance estimated using MC method in surface model rendering can be described as the product of albedo and illumination. Since the albedo obtained in surface model rendering is noise-free, only the illumination component needs to be denoised [11]. However, due to the absence of explicit surface intersections and noise-free albedo in VPT sampling, the radiance estimated using MC usually needs to be denoised as a whole. Noisy auxiliary features [8, 9] are usually used to guide the denoising. Therefore, blurring effects are possibly introduced. Our work tries to address this issue via decoupling the radiance sample.

A similar methodology is from the work of Zimmer et al. [35], who employed a decoupling strategy in handling radiance samples by decomposing them into distinct components. Each component underwent individual temporal motion estimation and denoising procedures. Their decoupling approach primarily involves decomposing radiance into different levels of specular radiance and diffuse radiance, followed by decomposing various radiance components into albedo and irradiance. Different from their method, our decoupling approach involves decomposing radiance into components of different noise levels and detail levels. In

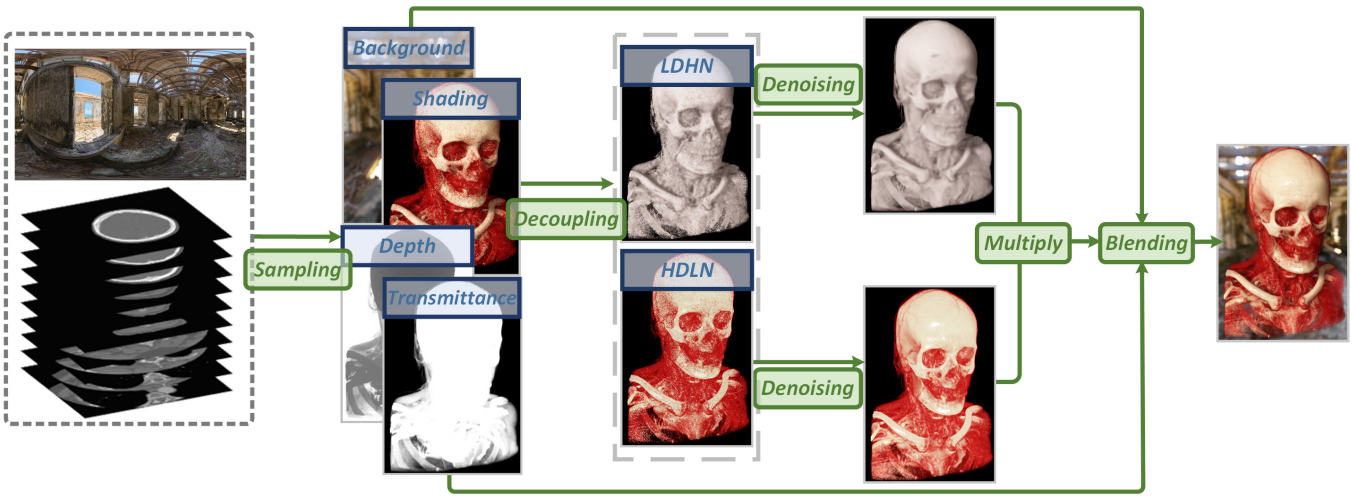


Fig. 5. An overview of our denoising process. In VPT sampling, the shading value (*Shading*), background radiance (*Background*), volume transmittance (*Transmittance*), and depth (*Depth*) are obtained. The shading outcome is decomposed into high-detail components (HDLN) and low-detail components (LDHN). Subsequently, both components undergo denoising separately using different spatiotemporal denoisers, and the denoised components are then multiplied to obtain the denoised shading result. The denoised shading result is blended with *Background* using *Alpha*, producing the display output.

our method, a radiance sample is decomposed into low-detail-high-noise (LDHN) component $F_{\mathbf{L}}$ and high-detail-low-noise (HDLN) component $F_{\mathbf{x}}$:

$$F_{\mathbf{L}} = \int_{\mathcal{S}_{4\pi}} L(\mathbf{x}, \omega')\mathrm{d}\omega' \qquad F_{\mathbf{x}} = \frac{L(\mathbf{x}, \omega)}{F_{\mathbf{L}}}$$

$F_{\mathbf{L}}$ includes radiance of light sources $L(\mathbf{x}, \omega')$ and light transmittance $T_r(\mathbf{x}, \mathbf{y}_i)$. $F_{\mathbf{x}}$ contains information on scattering properties. When using MC method, the estimation for $F_{\mathbf{L}}$ is:

$$F_{\mathbf{L}} \approx \frac{T_r(\mathbf{x}, \mathbf{y}_i) \cdot L(\mathbf{x}, \omega')}{p(\mathbf{y}_i)}$$

$F_{\mathbf{L}}$ contains the transmittance estimation and it has severe noise. In contrast, $F_{\mathbf{x}}$ encompasses scattering characteristics and has a lower degree of noise.

In practical computations, it is not viable to obtain $F_{\mathbf{x}}$ by dividing the estimated $Li(\mathbf{x}, \omega)$ by $F_{\mathbf{L}}$ due to the potential diminutive or zero values of $F_{\mathbf{L}}$, leading to inaccurate results. Hence, acquiring $F_{\mathbf{x}}$ necessitates the dissection of the rendering equation. A detailed example of decoupling is provided in the supplementary material #1. Complex realistic materials typically consist of diffuse and smooth specular reflection components. In denoising, these components are usually processed separately; our implementation adheres to this approach. Since the phase function used in our renderer is low-frequency, it shall be categorized under the diffuse component. Both HDLN and LDHN can be divided into diffuse and specular parts, and a radiance sample can be decoupled into:

$$Li(\mathbf{x}, \omega) \approx F_{\mathbf{x}}(d) \cdot F_{\mathbf{L}}(d) + F_{\mathbf{x}}(s) \cdot F_{\mathbf{L}}(s) + F_m \qquad (8)$$

where F_m denotes the shading values for the second and subsequent scatterings in multiple scattering. As F_m also has high noise, its denoising process aligns entirely with that of F_L. Therefore, we omit the description of F_m in the subsequent exposition. The distinct components generated by decoupling are shown in Fig. 6 and supplementary material #1.

Fig. 6. The appearances of distinct components obtained after decoupling. $F_{\mathbf{x}}(d)$ and $F_{\mathbf{x}}(s)$ represent the diffuse and specular part of HDLN component, respectively. $F_{\mathbf{L}}(d)$ and $F_{\mathbf{L}}(s)$ correspond to the diffuse and specular part of LDHN component, respectively. The presented appearances in the figure were obtained after individually sampling each component with 1024 spp. For additional examples, please refer to supplementary material #1.

The denoising method for the diffuse part can also be applied to the specular part. Therefore, we will only focus on the diffuse part for simplicity. Indeed, employing the same denoising procedure for both the diffuse part and specular part implies that denoising parameters need to be computed only once, resulting in time-saving benefits for the denoising process.

Efficient denoising of multiple samples is crucial. In the case that n radiance samples are obtained per pixel, we can describe the diffuse part using the following equation:

$$L_{dif}(\mathbf{x}, \omega) \approx \frac{1}{n} \sum_{i=1}^{n} \Big(F_{\mathbf{x}}(d_i) \cdot F_{\mathbf{L}}(d_i) \Big) \qquad (9)$$

If $F_{\mathbf{x}}(d_1) \approx F_{\mathbf{x}}(d_2) \approx \cdots \approx F_{\mathbf{x}}(d_l)$ and $F_{\mathbf{x}}(d_{l+1}) \approx F_{\mathbf{x}}(d_{l+2}) \approx \cdots \approx F_{\mathbf{x}}(d_{l+m})$ where l and m satisfy $l+m=n$, $L_{dif}(\mathbf{x}, \omega)$ can be approximated as:

$$\begin{aligned} L_{dif}(\mathbf{x}, \omega) \approx \frac{l}{n} \Big(\frac{1}{l} \sum_{i=1}^{l} F_{\mathbf{x}}(d_i) \Big) \cdot \Big(\frac{1}{l} \sum_{i=1}^{l} F_{\mathbf{L}}(d_i) \Big) \\ + \frac{m}{n} \Big(\frac{1}{n} \sum_{i=l+1}^{l+m} F_{\mathbf{x}}(d_i) \Big) \cdot \Big(\frac{1}{m} \sum_{i=l+1}^{l+m} F_{\mathbf{L}}(d_i) \Big) \end{aligned} \qquad (10)$$

This involves categorizing samples into two categories based on the similarity of their HDLN values. Increasing the number of categories in classification is expected to improve the result accuracy, as shown in Fig. 7. The *Ref* is rendered using Equ. 9 and it can be noticed that clustering the data into two categories does not yield any discernible visible difference when compared to the reference. Categorizing the data into three classes not only fails to yield a notable improvement in result accuracy, but also introduces additional computational overhead. Therefore, we choose to cluster samples within each pixel into two categories. Although traditional clustering algorithms [52] can be used, we found a more effective way to cluster samples in our method. First, the luminance values of $F_{\mathbf{x}}$ samples are calculated and sorted. The maximum difference between adjacent elements is found in the ordered sequence of luminance values. The preceding elements are classified into one group and the subsequent elements are classified into another. Additionally, if the absolute difference between the minimum and maximum values is less than 0.1, all samples in a pixel are grouped into a single category.

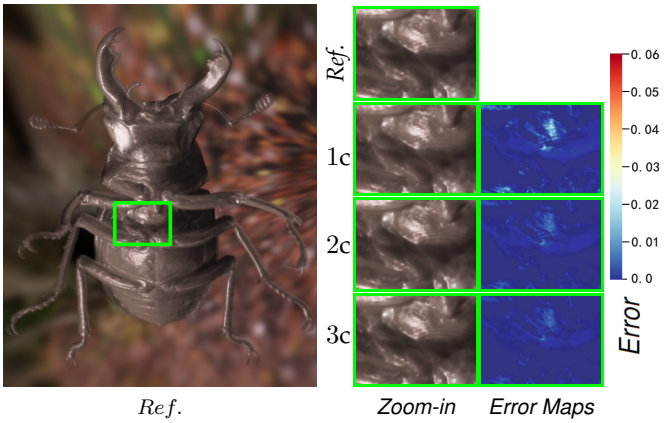


Fig. 7. The $Ref.$ is the reference image obtained without the radiance decoupling method. nc (n is 1, 2 or 3) indicates that the $F_{\mathbf{x}}$ samples were clustered into n groups. Error maps measure the absolute error between $Ref.$ and n-class clustering results. The images were all rendered with 1024 spp and were not denoised.

To better clarify our denoising strategy in Sec. 5.3, we use $\mathcal{C}_k$ to represent the k-th category of samples and n_k to represent the number of samples in category $\mathcal{C}_k$. Because the same denoising approach is applied to both diffuse and specular parts, mathematical symbols used are not differentiated between these two parts: both $F_{\mathbf{x}}(d_i)$ and $F_{\mathbf{x}}(s_i)$ are represented as $F_{\mathbf{x}}(i)$, and both $F_{\mathbf{L}}(d_i)$ and $F_{\mathbf{L}}(s_i)$ are represented as $F_{\mathbf{L}}(i)$. For the j-th pixel, taking the average of the samples within each category of each component:

$$F^j_{\mathbf{x}}(\mathcal{C}_k) := \frac{1}{n_k} \sum_{F^j_{\mathbf{x}}(i) \in \mathcal{C}_k} F^j_{\mathbf{x}}(i) \qquad F^j_{\mathbf{L}}(\mathcal{C}_k) := \frac{1}{n_k} \sum_{F^j_{\mathbf{x}}(i) \in \mathcal{C}_k} F^j_{\mathbf{L}}(i)$$

This is equivalent to averaging multiple samples as input for the denoiser [12, 41, 42] and and is elucidated in the denoising procedure outlined in Fig. 8.

5.2 Temporal reprojection

While reprojection may not be the most precise method for obtaining temporally correlated pixels, its computational speed makes it a common choice in real-time rendering systems. As DVR sampling cannot ascertain a unique intersection point in the volume space, the distance from the camera origin to the location of the first scattering event is often employed as the depth value required for reprojection [8, 9]. However, the reprojection strategy becomes less accurate in the presence of large areas with low particle density within the volume.

We revised their reprojection strategy in our approach. For the first real-scattering locations sampled in a pixel, the depth values where the particle density exceeds a certain threshold are recorded. The lowest depth among the recorded values (denoted as d^j for pixel j) are used for the reprojection. If the particle density at all sampled locations falls below this threshold, we use the median of these depth values instead. In practice, we set this threshold at $1/4$ of the maximum particle density within the volume. The reprojection formula is represented as $\pi(d^j) \rightarrow i$, indicating that the current frame's pixel j is reprojected onto the previous frame's pixel i.

5.3 Spatiotemporal denoising

The complexity of the description arises from the categorization of samples from a single pixel into multiple classes. To facilitate understanding of our denoising process, we have provided two points of explanation:

1) Each pixel comprises multiple $F_{\mathbf{x}}$ categories. For a specific category $F_{\mathbf{x}}(\mathcal{C}_k)$ within a pixel, other categories $F_{\mathbf{x}}(\mathcal{C}_{l,(l\neq k)})$ from this pixel can be considered as "neighborhood samples" with a pixel distance of 0, utilized for spatial denoising of $F_{\mathbf{x}}(\mathcal{C}_k)$.
2) Temporal denoising follows a similar rationale. The correlated pixels from the previous frame also encompass multiple categories. We calculate temporal weights for each of these categories and integrate them into the current frame's $F_{\mathbf{x}}(\mathcal{C}_k)$.

For temporal denoising, although using our reprojection approach may result in inaccuracies in highly transparent regions, our denoising technique can be used to minimize the existence of temporal ghosting artifacts and improve the visual quality of rendering results. Low-noise raw rendering results and adaptive temporal denoising can effectively help compensate for inaccuracies in the reprojection.

Terms and notations for explaining our denoising method in mathematics are given in Table 2. The denoising approaches for the HDLN component ($F^j_{\mathbf{x}}(\mathcal{C}_k)$) and the LDHN component ($F^j_{\mathbf{L}}(\mathcal{C}_k)$) are introduced separately in the following paragraphs. The denoised $F_{\mathbf{x}}$ and $F_{\mathbf{L}}$ are multiplied and then blended with background radiance to compose rendering outputs, as described in Fig. 5. The denoising process undergone by samples within a pixel is illustrated in Fig. 8.

5.3.1 Denoising of the high-detail-low-noise component

$F^j_{\mathbf{x}}(\mathcal{C}_k)$ usually has a small variance and can be denoised using the bilateral filter. The bilateral weights are calculated using d^j and $F_{\mathbf{x}}$:

$$w_{l,m} = \exp\left(\frac{-||F^l_{\mathbf{x}}(\mathcal{C}_m) - F^j_{\mathbf{x}}(\mathcal{C}_k)||}{\sigma_c}\right) \cdot \exp\left(\frac{-||d^l - d^j||}{\sigma_d}\right)$$
$$\widetilde{F^j_{\mathbf{x}}}(\mathcal{C}_k) \leftarrow \frac{1}{\sum_{l \in \mathcal{N}^j_{3\times 3}} w_{l,m}} \sum_{l \in \mathcal{N}^j_{3\times 3}} w_{l,m} F^l_{\mathbf{x}}(\mathcal{C}_m) \tag{11}$$

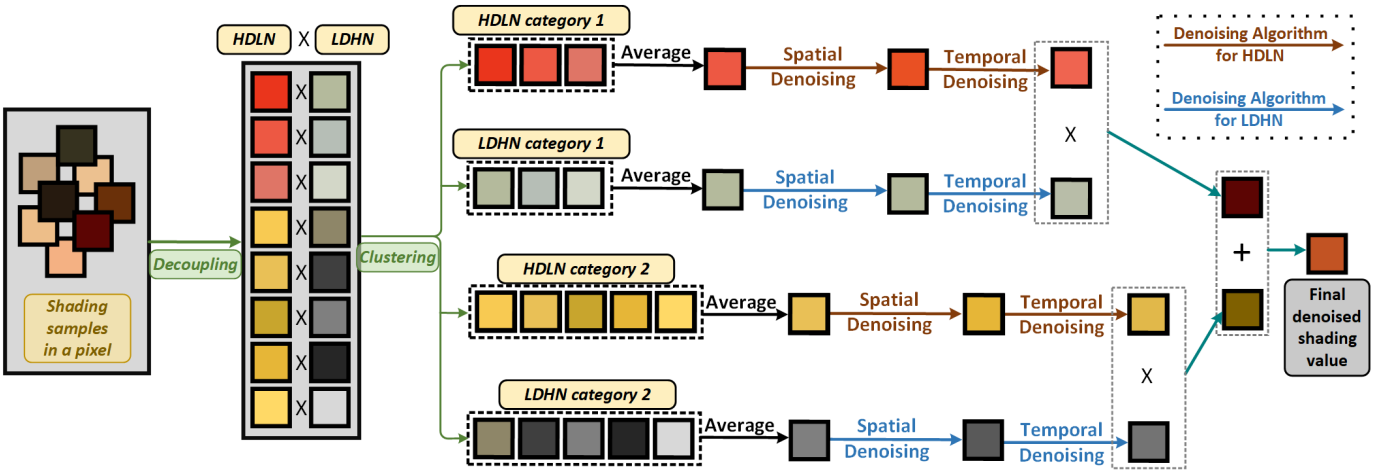


Fig. 8. This figure illustrates the denoising process for samples within a pixel. Each sample is decoupled into high-detail-low-noise (HDLN) and low-detail-high-noise (LDHN). The samples are then clustered into different categories based on their HDLN components, and the component values within each category are averaged. The HDLN and LDHN components are filtered using two different denoising algorithms, each labeled with a different color. The denoised results of all components are eventually combined to obtain the final desnoise shading value.

TABLE 2: Notations for denoising

Symbol	Description
n_k	the number of samples in $\mathcal{C}_k$ category
d^j	depth value for reprojection of j-th pixel
$\widehat{F^i_{\mathbf{x}}}$	$F_{\mathbf{x}}$ in the last frame at pixel i
$\widehat{F^i_{\mathbf{L}}}$	$F_{\mathbf{L}}$ in the last frame at pixel i
$\widetilde{F^j_{\mathbf{x}}}$	$F^j_{\mathbf{x}}$ value after spatial denoising
$\widetilde{F^j_{\mathbf{L}}}$	$F^j_{\mathbf{L}}$ value after spatial denoising
$\widehat{\alpha^i}$	transparency in the last frame at pixel i
$\overline{F^j_{\mathbf{x}}}$	$F^j_{\mathbf{x}}$ value after temporal denoising ($\widetilde{F^j_{\mathbf{x}}} \rightarrow \overline{F^j_{\mathbf{x}}}$)
$\overline{F^j_{\mathbf{L}}}$	$F^j_{\mathbf{L}}$ value after temporal denoising ($\widetilde{F^j_{\mathbf{L}}} \rightarrow \overline{F^j_{\mathbf{L}}}$)
$\mathbf{I}^j$	shading value of the j-th pixel
$\mathcal{N}^j_{n\times n}$	Neighborhood $n \times n$ pixels of pixel J
$lu^j(\mathcal{C}_k)$	luminance of $F^j_{\mathbf{L}}(\mathcal{C}_k)$

σ_c and σ_d represent the parameters of the joint bilateral filter, which are used to control the sensitivity of the weights to differences in color and depth values, respectively. The setup of $\sigma_c = 0.2$ and $\sigma_d = 2$ works well in our system. It is noteworthy that in Equ. 11, all $F_{\mathbf{x}}$ categories of neighboring pixels should be taken into account.

After reprojection $\pi(d^j) \rightarrow i$, $\widehat{F^i_{\mathbf{x}}}$ in the previous frame is used for denoising $F^j_{\mathbf{x}}(\mathcal{C}_k)$. The suitable weight of previous frame samples plays an important role in maintaining temporal stability and eliminating temporal delay. The following non-normalized weight is a proper option for blending:

$$\widehat{w}_l = \sqrt{\frac{1}{|n^j_k - \widehat{n}^i_l| + 1}} \exp\Big(-\frac{||\widetilde{F^j_{\mathbf{x}}}(\mathcal{C}_k) - \widehat{F^i_{\mathbf{x}}}(\mathcal{C}_l)||}{\sigma_e}\Big) \quad (12)$$

Since $\widehat{w}_l$ is related to the number of samples n_k in $\mathcal{C}_k$, the category containing more samples has a higher weight. σ_e can be used to adjust the weights of samples and is empirically set to 0.2 in our experiments. The sample values involved are weighted, summed, and subsequently divided by the sum of the weights to yield the denoised result.

5.3.2 Denoising of the low-detail-high-noise component

For spatial denoising, $F^j_{\mathbf{L}}(\mathcal{C}_k)$ is denoised using the edge-stopping extended kernel filtering technique [11]. The initial filtering step utilizes a 5×5 convolutional kernel, whereas the subsequent step employs a 9×9 dilated convolutional kernel. The d^j and denoised $F^j_{\mathbf{x}}(\mathcal{C}_k)$ are used as bilateral features in the edge-stopping function to guide the spatial denoising of $F^j_{\mathbf{L}}(\mathcal{C}_k)$.

For temporal denoising of $F^j_{\mathbf{L}}(\mathcal{C}_k)$, the weight of previous frame $\widehat{F^i_{\mathbf{L}}}(\mathcal{C}_m)$ is denoted as $\widehat{w}^i_m$ and the luminance [11] of $F^j_{\mathbf{L}}(\mathcal{C}_k)$ is denoted as $lu^j(\mathcal{C}_k)$. The non-normalized weight is calculated as the following:

$$\widehat{w}^i_m = \frac{1}{\mathcal{S}(a \cdot (||lu^j(\mathcal{C}_k) - \widehat{lu^i}(\mathcal{C}_m)|| - 1))}$$
$$w^j_k = 1 + \frac{b}{\epsilon + \sum \widehat{w}^i_m} \quad (13)$$

a indicates a constant that controls the shape of the $sigmoid$ function $\mathcal{S}$ and is empirically set to 3 in our experiments. ϵ is a very small positive number. Constant b controls the weight of $F^j_{\mathbf{L}}(\mathcal{C}_k)$ and is empirically set to 5 in our experiments.

After applying weight normalization, the temporal denoised result is achieved by blending the previous frame's result with the current frame, employing the weighted blending approach.

If the difference between $lu^j(\mathcal{C}_k)$ and $\widehat{lu^i}(\mathcal{C}_m)$ is substantial, the value of $\widehat{w}^i_m$ approaches 1, and the value of w^j_k approaches $1+b$. Consequently, the contribution of samples in the current frame becomes significantly greater compared to the previous frame. This approach effectively reduces the temporal delay between denoised frames.

5.3.3 Compositing denoised DVR results

The amount of sampling rays emitted from a pixel per frame is denoted as N^j, i.e. N^j spp. Among N^j sampling rays, the number of rays that have undergone scattering events is marked as N^j_p. When VPT samples of a pixel are divided into K categories, $\sum_{k=1}^{K} n_k = N^j_p$. N^j_p/N^j can be considered as the opacity α^j of a pixel. It is also necessary to perform spatiotemporal denoising on α^j. For the spatial

denoising of α^j, a 5×5 bilateral filter is applied. The temporal denoising of α^j can be formulated as following:

$$\alpha^j \Leftarrow \frac{\alpha^j - \exp(-[1 + |\alpha^j - \widehat{\alpha^i}|])\widehat{\alpha^i}}{1 + \exp(-[1 + |\alpha^j - \widehat{\alpha^i}|])} \tag{14}$$

In VPT, sampling rays have a high probability of passing through highly transparent volume areas without scattering. In such a case, the radiance in the background HDR environment map shall be sampled. In our denoising pipeline, the background radiance samples are not input into the denoiser, as shown in Fig. 5. Additionally, our rendering approach adheres to the null-scattering theory [5], where light does not refract in the presence of variations in medium density. Consequently, the denoised shading values can be blended with the sampled background radiance using α^j. The background radiance sampled along the camera ray $\mathbf{r}^j$ is denoted as B^j. The final DVR output $\overline{L}$ is computed based on the following equation:

$$\overline{L} = \alpha^j \frac{\sum_{k=1}^{K} n_k \cdot \left(\overline{F_{\mathbf{x}}^j}(\mathcal{C}_k) \cdot \overline{F_{\mathbf{L}}^j}(\mathcal{C}_k)\right)}{\sum_{k=1}^{K} n_k} + (1 - \alpha^j)B^j \tag{15}$$

Avoiding the input of sampled values containing the background color into the denoiser effectively prevents temporal ghosting artifacts, as demonstrated in Sec. 6.4 and supplementary material #1. Our method employs N_p^j/N^j to estimate opacity. This is equivalent to using Delta-Tracking to estimate the extinction ratio, which requires a high spp to ensure the effectiveness of the estimation. When only a low number of spp is used for rendering, we recommend employing the Ray-Marching method to calculate opacity.

The configuration of some parameters in the denoising formula relies on empirical considerations. We observed that once appropriate parameters were identified for one scene, they demonstrated equal efficacy when applied to other scenes. However, the depth value parameter is distance-dependent. In our DVR system, the unit length in the coordinate space is defined as "decimeters," and the acquired depth values are also in "decimeters." If the unit length in the coordinate space is converted to "meters" or "centimeters," corresponding adjustments to parameter values are necessary.

In interactive DVR, volume rotation can occasionally lead to significant disparities between corresponding pixels across frames, particularly in highly transparent areas where temporal correlation is minimal. In such a case, samples in previous frame may carry a negligible weight and fail to guide temporal denoising. However, significant temporal changes will reduce the observer's perception of temporal flickering noise [53]. By applying both high sampling rates and the proposed spatiotemporal denoising technique, our method can effectively and consistently maintain a high visual quality of VPT-based DVR in real time.

6 Experiments and Evaluations.

In this section, we first describe the implementation details, followed by the description of experimental settings and validation of the effectiveness of our method. We not only conducted tests on the overall system performance but also compared our denoising method with state-of-the-art.

6.1 Implementation Details

Our enhanced VPT sampling method with less variance and novel spatiotemporal denoising technique are implemented and integrated into the realistic DVR pipeline of our prototyping volume visualization system. It is attached as an executable program with some demo datasets in our supplementary materials for free experiencing. In our realistic DVR renderer, the function of data clipping is set along the x, y and z axes to better visualize internal structures of volume data. In addition, preset area light sources can rotate around the volume or flash back and forth automatically. These functionalities can greatly facilitate the demonstration of the real-time DVR effect that volumetric structures are visualized with illuminations varying over time.

Illumination. Our system supports the utilization of multiple light sources for illumination and also accommodates the use of HDR panoramic environment map. When the HDR environment map is used for DVR illumination, light probe method can be applied to approximate the illumination [8]. To avoid excessively dark rendering results caused by low radiance values within certain areas of the HDR environment map, the average radiance of the HDR map is added to the shading results as a uniform ambient illumination. In ExposureRenderer [48], only single scattering is supported. We implement multiple scattering in our work, the scattering of each sampling ray is limited to no more than twice in order to balance the computational efficiency and rendering quality. The relationship between the number of ray bounces and rendering appearance is discussed in supplementary material #1.

Depth of field. To minimize the impact of the intricate background on user observation of the volumetric data, we employ an additional camera equipped with a large aperture for background sampling in our experiments.

Error maps. Error maps were all calculated in HDR space to compare and analyze the impact of denoising on the original estimation results without tone mapping.

6.2 Experiment Design for Evaluation

Hardware used in experiments. All of our experiments were performed on a PC workstation with an Intel i7-8700K CPU, 16GB RAM and an Nvidia GeForce GTX 1080Ti GPU with 11GB graphics RAM.

Rendering parameters. The image resolution of our volume rendering results was set to 1440×1190. For efficiency tests at different resolutions, please refer to the supplementary material #1. To maintain high temporal stability, images were rendered with 30 spp. Our PC workstation can provide a stable interactive rendering frame rate of no less than 35 frames per second (fps). The reference images used for our evaluations were all rendered with 1024 spp under the same conditions regarding corresponding MC estimation results and denoised results.

TABLE 3: Datasets used for evaluation

Dataset	Resolution	Voxel Dimension
Manix	$256 \times 256 \times 230$	$2.0, 2.0, 2.8\ (mm)$
Macoessix	$256 \times 256 \times 178$	$2.8, 2.8, 2.8\ (mm)$
Anocelia	$512 \times 512 \times 213$	$1.0, 1.0, 3.0\ (mm)$
Smoke	$200 \times 200 \times 80$	$1.0, 1.0, 1.0$

Datasets. Table 3 outlines experimental datasets used in our evaluations. These datasets were obtained from internet

sources of [54] and [55]. The *Manix* data features a smooth skull and translucent muscle, allowing for verifying the rendering realisticness. The *Macoessix* data contains large blood vessels that cast soft shadows on bones, facilitating a better perception of depth and illumination. The *Anocelia* data presents tiny and intricate structures, which can be used to verify whether overblurring is introduced to the denoised results. The *Smoke* data were generated through physical simulation. Additional examinations related to semi-transparent appearances and simulation datasets are presented in supplementary material #1.

Evaluation scheme. Our work aims to develop a real-time realistic DVR method capable of producing consistently high-quality rendering results with negligible noise for interactive scientific data visualization applications. Therefore, our evaluation includes two phases. The performances of our renderer were first evaluated at the system level. Second, the performance of our denoising algorithm was compared with state-of-the-art denoisers. We conducted tests to assess the numerical accuracy and temporal stability of denoised results. An in-depth analysis of both advantages and limitations of these methods was conducted based on our experimental results.

6.3 Performance of overall system

This section discusses the performance assessment of the overall rendering system, in terms of computational efficiency, numerical accuracy of rendering results and rendering quality in various interaction modes.

6.3.1 Trade-off between sampling rate and quality

The efficiency of our volume rendering framework was evaluated by measuring the fps and corresponding numerical accuracy with different sample sizes. The volume was moved and rotated along a predetermined trajectory and an area light source was set to rotate around the volume. Reference results were obtained by rendering 1024 spp under the same conditions. The peak signal-to-noise ratio (PSNR) between denoised results and reference images was calculated to measure the numerical accuracy. A higher PSNR value indicates greater numerical accuracy of the rendering results.

As shown in Fig. 9, our rendering system facilitates the efficient computation of multiple VPT samples for real-time interactive DVR. The implementation of this multi-sampling strategy does not rely on advanced graphics hardware but on our sampling optimizations. On more advanced graphics hardware, such as the Nvidia RTX 4070Ti GPU, the rendering frame rate can reach up to 130 fps with 30 spp. According to hardware capabilities of our PC workstation used in our experiments, a sampling rate of 30 spp is recommended for a good balance between computational efficiency and rendering quality. This allows for maintaining sufficiently high temporal stability even during rapid volume rotations.

6.3.2 The quality of interactive DVR results

The performance of our rendering system was assessed in different user interaction modes. The light sources used to illuminate the volumetric data included an HDR environment map and an area light. In our experiment, the area light was set to continuously rotate around the volume. To introduce noticeable variations in the rendered appearance between adjacent frames, we executed substantial rotations of the volume or adjusting transfer functions.

As shown in Fig. 10, during volume rotation, the soft shadows presented in the *Macoessix* data rendering exhibited a high level of quality without overblurring. Moreover, there is no noticeable temporal ghosting introduced during camera rotation or adjustment of the transfer function. For example, in Fig. 10, when manipulating the transfer function of *Manix* data to instantaneously make muscle tissues transparent, there is no noticeable residue left in the denoised result. Our system is capable of producing high-quality rendering results even for simulated data with non-surface appearances, such as the *Smoke* dataset. It is noteworthy that the introduced blurring from the denoising of the *Smoke* rendering is less perceptible, given the inherent lack of pronounced high-frequency details in *Smoke* itself. Further examples are provided in the supplementary material #1.

6.4 Comparison between our denoiser and state-of-the-art denoisers

The performance of our denoising algorithm was evaluated based on a comparison with state-of-the-art real-time denoisers, including [12], [9] and [8].

6.4.1 Implementations of denoising methods

SVGF [12] performs a wide range filtering by gradually iterating the expanding filter kernel and calculates the contribution of previous frames according to the gradient of illumination temporal change. In our implementation, noisy G-buffers and the spatial variance of luminance were used for denoising. It takes about 22 ms to denoise a single rendered frame on our workstation.

FDAE [9] refers to the *feature reprojection dual autoencoder* network. To prepare datasets used for training networks, we generated 30 image sequences, each of which contained 1000 continuous frames. The detailed settings adhere to the specifications are outlined in [9], as well as their source code. The FDAE network was trained for 900 epochs. The average time required for denoising a single frame is 97 ms on our PC workstation.

WRLS [8]. WRLS is a linear fitting denoising method. In our experiments, WRLS was implemented following the guidelines given in [8]. It takes about 15 ms to denoise each rendered frame on our workstation.

These denoising methods all employ our reprojection scheme discussed in Sec. 5.2.

6.4.2 Performance comparison of denoising methods

In this experiments, the lighting conditions are the same as those described in Sec. 6.3.2. The volume was set to rotate along the preset trajectory. A sampling rate of 30 spp was used to generate shading results as input images to all denoisers to ensure a fair performance comparison and analysis of denoising.

Shading results and their denoised outputs are shown in Fig. 11. The numerical similarity between the denoised results and their corresponding reference images in consecutive frames is illustrated in Fig. 12. Learned perceptual image patch similarities (LPIPS) [56] measures the similarity between two images in a way that aligns with human judgment and was used to evaluate the rendering quality. A

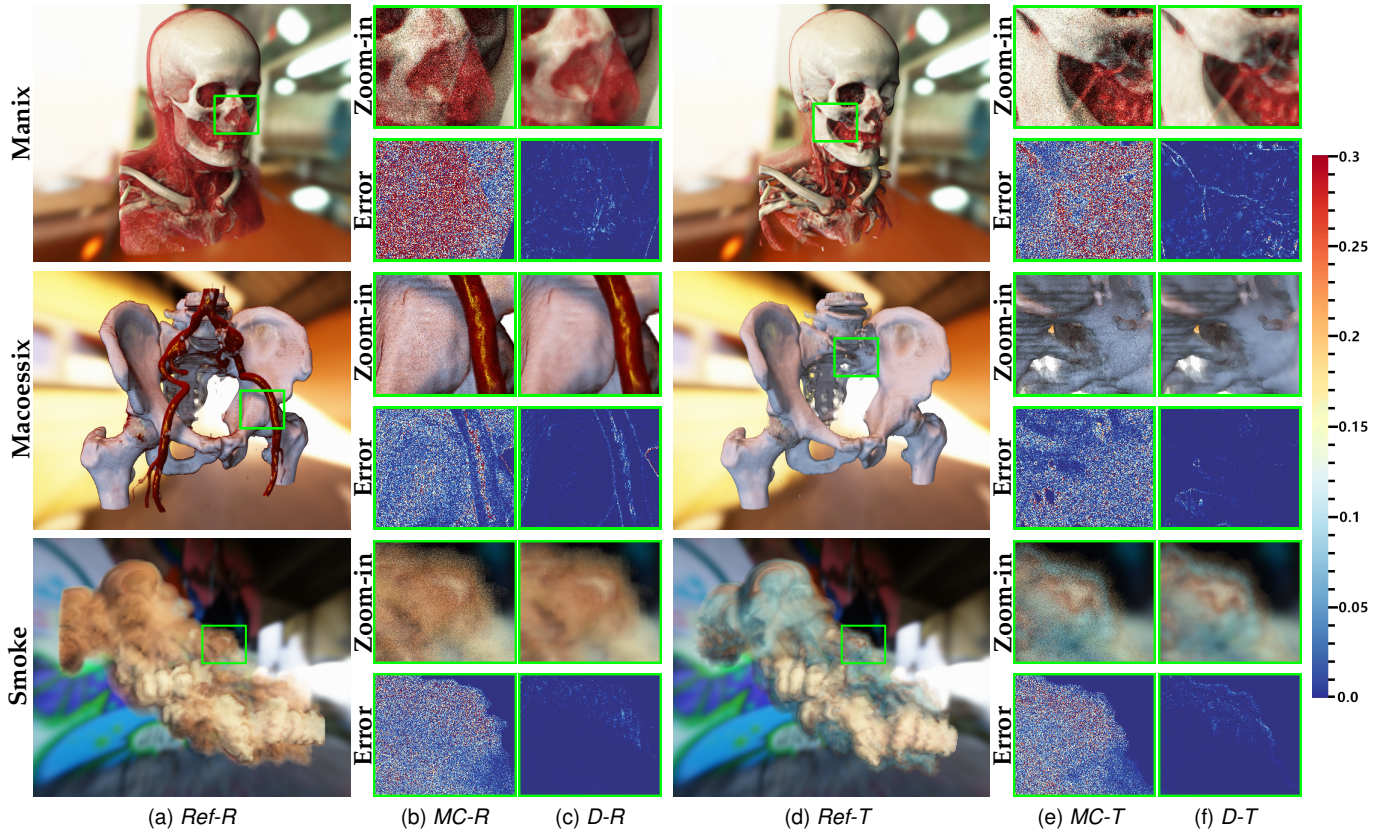


Fig. 10. *MC* indicates the VPT results rendered with 30 spp without denoising, and corresponding denoised outputs are represented as *D*. *Ref* represents the reference images obtained by rendering 1024 spp under the same conditions as *MC*. The suffix "-*R*" indicates results were obtained during the rotation of volume, while the suffix "-*T*" indicates results were obtained during the interaction with the transfer function.

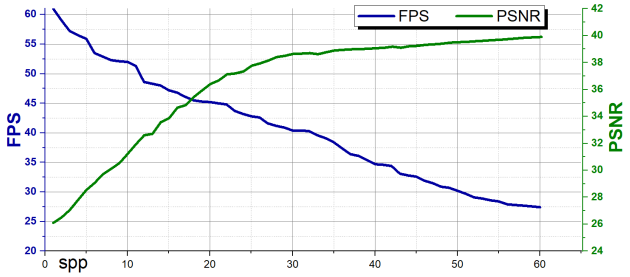


Fig. 9. The relationship between VPT sampling rate and rendering frame rate, as well as the relationship between sampling rate and numerical accuracy of rendering results. *Manix* data was utilized for testing.

lower value of LPIPS represents a better rendering quality. PSNR was also used to measure the numerical accuracy between reference images and denoised results.

It can be observed in Fig. 11 that our denoising method is able to consistently maintain high rendering quality with minimal overblurring artifacts during interactions. FDAE, WRLS and SVGF introduced visually noticeable blurring in the denoised results of *Manix* and *Macoessix*. Overblurring is caused by the utilization of neighborhood samples for spatial denoising. Most state-of-the-art methods use auxiliary noisy G-buffers to denoise the radiance samples. SVGF relies on the variance of radiance samples to calculate parameters of the filter. However, both spatial variances of transmittance estimation and illumination shading estimation are large for structurally complex volumetric data. This can fail the variance-based guidance and subsequently introduce blurring artifacts. In order to reduce noise, WRLS incorporates bilateral filtering in the final stage of denoising. However, blurring artifacts can be introduced. For FDAE, although the U-Net architecture is more capable of preserving details compared to autoencoder architecture [13], it still lacks generalization. When training data and testing data have different lighting conditions, noticeable blurring normally occurs. In addition, since network inputs typically encompass non-color features such as normals and depth, it is prone to introducing "nonexistent" colors in the denoising output and thus compromising its accuracy. We delve into this issue in our supplementary material #1. For our denoiser, the purpose of separately denoising the different components obtained by decoupling shading samples is to remove noise without loss of important details.

Temporal stability involves two aspects: the degrees of temporal flickering noise and temporal ghosting artifacts. The quantitative experimental evaluation of flickering noise will be described in Sec. 6.4.3. The introduction of ghosting artifacts is due to the contribution of outdated previous frame results on the current frame. As the sampling rays traverse through the volume space and sample the surrounding environment, a pixel's radiance can consist of both volumetric scattering radiance and environment radiance for translucent volume. Significant variations of the transparency of corresponding pixels between consecutive frames can lead to ghosting artifacts in the denoised results, particularly for SVGF denoiser. In Fig. 11, noticeable temporal latency can be observed in *Manix* and *Macoessix*

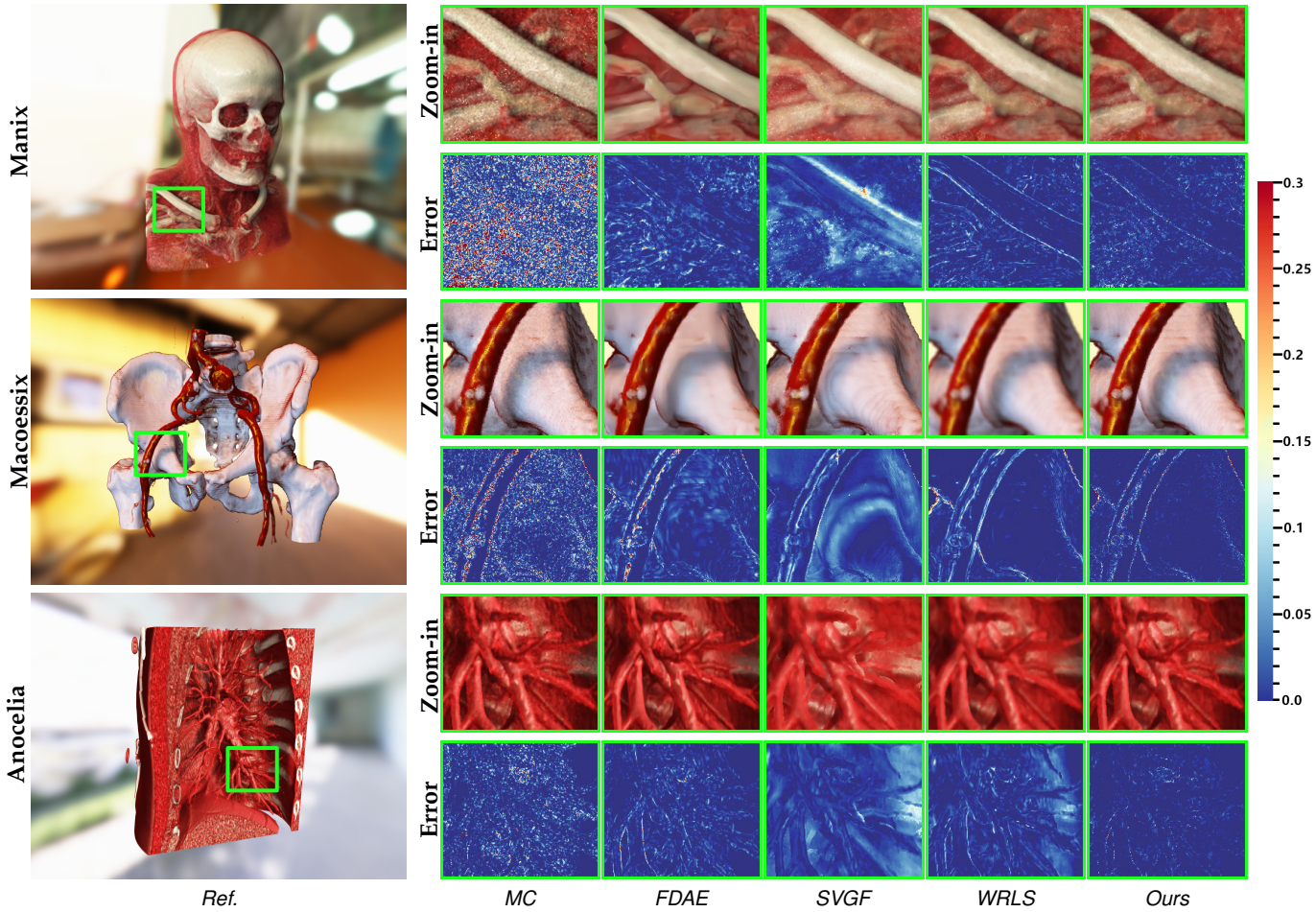


Fig. 11. *MC* are VPT-based DVR results rendered with 30 spp. The the denoised results were generated using *FDAE* [9], *SVGF* [11], *WRLS* [8] and our denoiser labeled as *Ours*. Reference images labeled as *Ref.* were rendered with 1024 spp under the same conditions as *MC*.

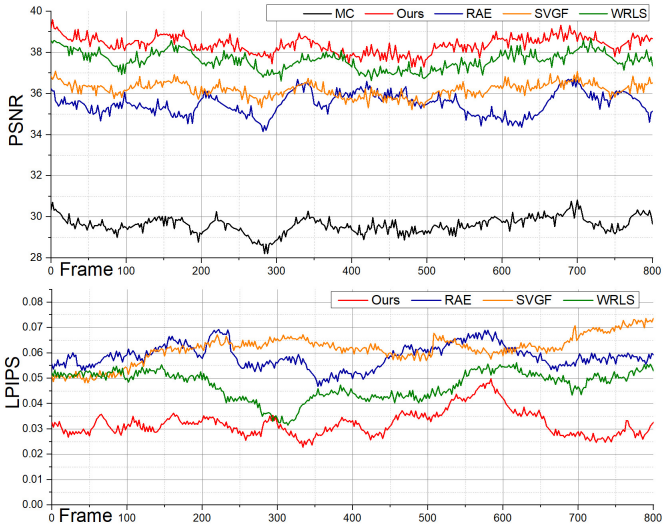


Fig. 12. Numerical accuracy comparison of the DVR results denoised by our denoising method and state-of-the-art denoisers. The testing utilized the *Manix* dataset.

for SVGF. Ghosting artifacts induced by significant volume rotation are shown in supplementary material #1. Compared to SVGF, the denoised results achieved with WRLS exhibit a reduced occurrence of ghosting artifacts. However, it is important to note that WRLS is associated with a higher level of flickering noise, as shown in Sec. 6.4.3 and our supplementary video. The ghosting artifacts introduced by FDAE are not significant. In our method, the advantages of multisampling are used to estimate the pixel opacity and perform denoising on it. For low sampling density, noise in opacity estimation can be effectively reduced via iterative extended kernel filtering.

In recent years, there has been extensive utilization of neural network denoising. To enhance their effectiveness, a crucial aspect is their integration with rendering theory. If the network solely relies on basic geometric features as inputs, it may lack the ability to generalize effectively. For example, the denoised results of Nvidia AI denoiser [13] were intolerably blurring in our experimental scenarios. There are various interaction modes [1] in scientific visualization applications. It requires sufficient training of the neural network for each interaction mode to achieve high rendering quality. This procedure necessitates a considerable allocation of computational resources and may potentially jeopardize the stability of the denoiser. Enhancing generalization capabilities can be achieved if the network can learn an approximate rendering process based on the intermediate information obtained from rendering [57].

6.4.3 *Variance of denoised DVR results*

An additional experiment was conducted to evaluate the temporal stability of the denoised DVR results, as it is closely associated with the variance of the denoised images.

The primary objective of denoising in MC-based rendering is to decrease image variance. Specifically, when multiple estimations are conducted in the same scene using different random sequences, resulting in multiple noisy results, the denoised results should exhibit high consistency. Otherwise, it suggests inadequate temporal stability.

In one test, we first rendered 29 frame images as historical rendering results. Then, for the 30-th frame, 50 different random number sequences were used to generate 50 shading estimation images. All these images for the 30-th frame share the same historical results. Subsequently, a denoising method was applied to these shading estimation results to generate 50 denoised results of the 30-th. The sum of pixel-wise variance was used as variance of these denoised images. For each denoiser, we conducted the test 30 times and plotted the experimental results on the box-plot in Fig. 13, where "images variance" corresponds to the variance of denoised outputs. The outputs of our denoiser have a smaller variance than other denoisers, which ensures that our denoised results obtained during interaction display reduced flickering noise and improved visual quality.

6.4.4 Comparison of convergence speed

Our evaluation also quantitatively assessed how fast the denoised result can converge to present a stable eventual rendering quality in the interactive volume visualization.

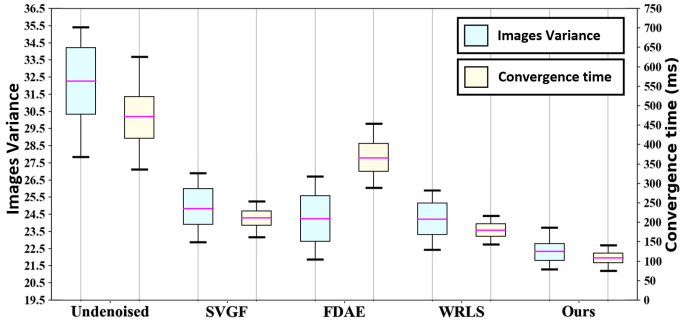


Fig. 13. The box plot describes the variance of denoiser's outputs (images variance) and the convergence time of denoised results (convergence time) by different denoisers. The pink line represents the median. Testing was performed using the *Manix* data.

In our experiment, user interaction is randomly halted, and the time elapsed from the moment of interruption until the rendering results converge is recorded. When the Root Mean Square Error (RMSE) between consecutive frame images is less than 0.01, we consider the rendering results to have converged at that point. For each denoiser, 50 tests were conducted and the results were demonstrated using a box-plot, as shown in Fig. 13. Our method exhibits the shortest convergence time with lowest fluctuations, which can be attributed to the short operation time of the denoiser and high temporal stability in denoised results. The FDAE denoiser necessitates a comparatively lengthy execution time for each denoising iteration, leading to an extended convergence time for the rendering results. The WRLS denoised results demonstrate a slightly lower convergence time compared to SVGF, mainly because of the notable temporal delay observed in the SVGF denoised results.

7 Conclusions and Future Work

Our research work tries to accomplish real-time realistic volume rendering, ensuring the preservation of high-quality rendering results even during continuous interaction. Our method not only employs optimized sampling algorithms to reduce the noise in shading results but also uses a novel spatiotemporal denoiser to achieve high-quality VPT-based DVR. With negligible additional computational expenses, shading results generated using our "surface-medium coexistence" model have less noise. Our method implements a hybrid acceleration structure based on macrocell and octree, which allows the processing of more VPT samples while ensuring a high interaction rate. Our denoiser decouples radiance samples into high-detail and low-detail components, each of which can be separately denoised. This can effectively eliminate noise while preserving important details and is beneficial for maintaining temporal stability and avoiding temporal ghosting artifacts.

Our prototype DVR system can produce rendering results with the high numerical accuracy and consistently superior visual quality. Additionally, our method is capable of maintaining visual comfort in challenging interaction situations, such as rapid volume rotating and light source flickering. Compared with existing realistic DVR methods, our method has advantages in keeping higher interaction fluency, maintaining better temporal stability and generating better visual effects of the rendering results. Our research presents promising prospects for the development of high-quality and real-time interactive rendering techniques.

There are still some limitations that shall be addressed in our future work. To ensure high temporal stability, multiple VPT samples are required. As the volume size increases, the time cost required for sampling one spp also increases, resulting in a decrease of rendering frame rate. Our workstation supports a frame rate of only about 26 fps with 30 spp when rendering a $512 \times 512 \times 512$ volume grid. Our future research aims to integrate the proposed VPT-based DVR rendering method into interactive mixed reality applications, which impose higher demands on interaction fluency and visual comfort. We will try to enhance rendering quality while reducing the sampling rate. Additionally, our denoiser currently necessitates manual parameter adjustments, which may require readjustments when the brightness level changes within a scene to achieve optimum performance. To further enhance the efficiency of our method, we plan to incorporate adaptive weighting strategies to eliminate the necessity for manual readjustments.

Acknowledgments

The authors extend their gratitude to T. Kroes for providing ExposureRenderer. Our sincere appreciation goes to the authors of PBRT for their inspiration in implementing some of the techniques mentioned in this paper. We would like to express our gratitude to P. Klacansky and OsiriX website for providing the dataset used in this study, as well as to all the authors who contributed to the collection of this valuable volume data. Additionally, we would like to extend our thanks to the sIBL Archive website for providing the exceptional HDR environment map used in our research.

REFERENCES

[1] B. Preim and C. P. Botha, *Visual Computing for Medicine, Second Edition: Theory, Algorithms, and Applications*. Morgan Kaufmann Publishers Inc, 2013.

[2] L. Linsen, B. Hamann, and H.-C. Hege, *Visualization in Medicine and Life Sciences III: Towards Making an Impact*. Springer, 2016.

[3] S. P. Rowe, P. T. Johnson, and E. K. Fishman, "Initial experience with cinematic rendering for chest cardiovascular imaging," *The British journal of radiology*, vol. 91, no. 1082, p. 20170558, 2018.

[4] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2023.

[5] J. Novák, I. Georgiev, J. Hanika, and W. Jarosz, "Monte Carlo Methods for Volumetric Light Transport Simulation," *Computer Graphics Forum*, 2018.

[6] T. Kroes, F. Post, and C. Botha, "Interactive direct volume rendering with physically-based lighting," *Eurographics*, pp. 1–10, 2012.

[7] K. Engel, "Real-time monte-carlo path tracing of medical volume data," in *NVIDIA GPU Technology Conference (GTC)*, 2016.

[8] J. A. Iglesias-Guitian, P. S. Mane, and B. Moon, "Real-Time Denoising of Volumetric Path Tracing for Direct Volume Rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 01, pp. 2734–2747, Nov. 2020.

[9] N. Hofmann, J. Martschinke, K. Engel, and M. Stamminger, "Neural denoising for path tracing of medical volumetric data," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 3, no. 2, pp. 1–18, 2020.

[10] P. V. Radziewsky, T. Kroes, and M. Eisemann, "Efficient Stochastic Rendering of Static and Animated Volumes Using Visibility Sweeps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 9, pp. 2069–2081, 2016.

[11] C. Schied, A. Kaplanyan, C. Wyman, A. Patney, C. R. A. Chaitanya, J. Burgess, S. Liu, C. Dachsbacher, A. Lefohn, and M. Salvi, "Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination," 2017, pp. 1–12.

[12] C. Schied, C. Peters, and C. Dachsbacher, "Gradient estimation for real-time adaptive temporal filtering," in *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 2, 2018, pp. 1–16.

[13] C. R. A. Chaitanya, A. S. Kaplanyan, C. Schied, M. Salvi, A. Lefohn, D. Nowrouzezahrai, and T. Aila, "Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–12, 2017.

[14] H. Fan, R. Wang, Y. Huo, and H. Bao, "Real-time monte carlo denoising with weight sharing kernel prediction network," in *Computer Graphics Forum*, vol. 40, no. 4. Wiley Online Library, 2021, pp. 15–27.

[15] N. Hofmann, J. Hasselgren, P. Clarberg, and J. Munkberg, "Interactive Path Tracing and Reconstruction of Sparse Volumes," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 4, no. 1, pp. 1–19, 2021.

[16] D. Lin, C. Wyman, and C. Yuksel, "Fast volume rendering with spatiotemporal reservoir resampling," *ACM Transactions on Graphics*, vol. 40, no. 6, pp. 1–18, 2021.

[17] A. Knoll, "A survey of octree volume rendering methods." *VLUDS*, pp. 87–96, 2006.

[18] L. Szirmay-Kalos, B. Tóth, and M. Magdics, "Free path sampling in high resolution inhomogeneous participating media," *Computer Graphics Forum*, vol. 30, no. 1, pp. 85–97, 2011.

[19] J. Amanatides, A. Woo *et al.*, "A fast voxel traversal algorithm for ray tracing." in *Eurographics*, vol. 87, no. 3. Citeseer, 1987, pp. 3–10.

[20] D. Jönsson, E. Sundén, A. Ynnerman, and T. Ropinski, "A Survey of Volumetric Illumination Techniques for Interactive Volume Rendering," *Computer Graphics Forum*, vol. 33, no. 1, pp. 27–51, Oct. 2014.

[21] D. Jönsson and A. Ynnerman, "Correlated photon mapping for interactive global illumination of time-varying volumetric data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 901–910, 2016.

[22] O. Igouchkine, Y. Zhang, and K.-L. Ma, "Multi-material volume rendering with a physically-based surface reflection model," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 12, pp. 3147–3159, 2017.

[23] J. G. Magnus and S. Bruckner, "Interactive dynamic volume illumination with refraction and caustics," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 984–993, 2017.

[24] D. Bauer, Q. Wu, and K.-L. Ma, "Photon field networks for dynamic real-time volumetric global illumination," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[25] S. Stegmaier, M. Strengert, T. Klein, and T. Ertl, "A simple and flexible volume rendering framework for graphics-hardware-based raycasting," in *Fourth International Workshop on Volume Graphics, 2005.* IEEE, 2005, pp. 187–241.

[26] M. Galtier, S. Blanco, C. Caliot, C. Coustet, J. Dauchet, M. El Hafi, V. Eymet, R. Fournier, J. Gautrais, A. Khuong *et al.*, "Integral formulation of null-collision monte carlo algorithms," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 125, pp. 57–68, 2013.

[27] J. Novak, A. Selle, and W. Jarosz, "Residual ratio tracking for estimating attenuation in participating media," *ACM Transactions on Graphics*, vol. 33, no. 6CD, pp. 179.1–179.11, 2014.

[28] B. Miller, I. Georgiev, and W. Jarosz, "A null-scattering path integral formulation of light transport," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–13, 2019.

[29] Ž. Lesar, C. Bohak, and M. Marolt, "Real-time interactive platform-agnostic volumetric path tracing in webgl 2.0," in *Proceedings of the 23rd International ACM Conference on 3D Web Technology*, 2018, pp. 1–7.

[30] A. A. Javed, R. W. Young, J. R. Habib, B. Kinny-Köster, S. M. Cohen, E. K. Fishman, and C. L. Wolfgang, "Cinematic rendering: novel tool for improving pancreatic cancer surgical planning," *Current Problems in Diagnostic Radiology*, vol. 51, no. 6, pp. 878–883, 2022.

[31] E. Denisova, L. Manetti, L. Bocchi, and E. Iadanza, "Ar2t: Advanced realistic rendering technique for biomedical volumes," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2023, pp. 347–357.

[32] Q. Wu, D. Bauer, M. J. Doyle, and K.-L. Ma, "Interactive volume visualization via multi-resolution hash encoding based neural representation," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[33] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon, "Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering," *Computer Graphics Forum*, vol. 34, no. 2, pp. 667–681, 2015.

[34] Y. Huo and S.-e. Yoon, "A survey on deep learning-based Monte Carlo denoising," *Computational Visual Media*, vol. 7, pp. 169–185, 2021.

[35] H. Zimmer, F. Rousselle, W. Jakob, O. Wang, D. Adler, W. Jarosz, O. Sorkine-Hornung, and A. Sorkine-Hornung, "Path-space motion estimation and decomposition for robust animation filtering," in *Computer Graphics Forum*, vol. 34, no. 4. Wiley Online Library, 2015, pp. 131–142.

[36] B. Moon, J. A. Iglesias-Guitian, S.-E. Yoon, and K. Mitchell, "Adaptive rendering with linear predictions," *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 1–11, 2015.

[37] B. Bitterli, F. Rousselle, B. Moon, J. A. Iglesias-Guitián, D. Adler, K. Mitchell, W. Jarosz, and J. Novák, "Nonlinearly weighted first-order regression for denoising monte carlo renderings," in *Computer Graphics Forum*, vol. 35, no. 4, 2016, pp. 107–117.

[38] M. Koskela, K. Immonen, M. Mäkitalo, A. Foi, T. Viitanen, P. Jääskeläinen, H. Kultala, and J. Takala, "Blockwise multi-order feature regression for real-time path-tracing reconstruction," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–14, 2019.

[39] S. Bako, T. Vogels, B. Mcwilliams, M. Meyer, J. Novk, A. Harvill, P. Sen, T. Derose, and F. Rousselle, "Kernel-predicting convolutional networks for denoising Monte Carlo renderings," *ACM Transactions on Graphics*, vol. 36, no. 4CD, pp. 1–14, 2017.

[40] M. Gharbi, T.-M. Li, M. Aittala, J. Lehtinen, and F. Durand, "Sample-based Monte Carlo denoising using a kernel-splatting network," *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 1–12, 2019.

[41] B. Xu, J. Zhang, R. Wang, K. Xu, Y.-L. Yang, C. Li, and R. Tang, "Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation." *ACM Transactions on Graphics*, vol. 38, no. 6, pp. 224–1, 2019.

[42] X. Meng, Q. Zheng, A. Varshney, G. Singh, and M. Zwicker, "Real-time Monte Carlo Denoising with the Neural Bilateral Grid." in *Computer Graphics Forum*, 2020, pp. 13–24.

[43] M. Işık, K. Mullia, M. Fisher, J. Eisenmann, and M. Gharbi, "Interactive monte carlo denoising using affinity of neural features," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–13, 2021.

[44] J. Martschinke, S. Hartnagel, B. Keinert, K. Engel, and M. Stamminger, "Adaptive temporal sampling for volumetric path tracing of medical data," in *Computer Graphics Forum*, vol. 38, no. 4. Wiley Online Library, 2019, pp. 67–76.

[45] D. Bauer, Q. Wu, and K.-L. Ma, "Fovolnet: Fast volume rendering using foveated deep neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 515–525, 2022.

[46] J. Taibo and J. A. Iglesias-Guitian, "Immersive 3d medical visualization in virtual reality using stereoscopic volumetric path tracing," in *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2024, pp. 1044–1053.

[47] Z. Misso, Y. K. Li, B. Burley, D. Teece, and W. Jarosz, "Progressive null-tracking for volumetric rendering," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–10.

[48] T. Kroes, F. H. Post, and C. P. Botha, "Exposure Render: An Interactive Photo-Realistic Volume Rendering Framework," *PLoS ONE*, vol. 7, no. 7, pp. e38 586–, 2012.

[49] M. Eid, C. N. De Cecco, J. W. Nance Jr, D. Caruso, M. H. Albrecht, A. J. Spandorfer, D. De Santis, A. Varga-Szemes, and U. J. Schoepf, "Cinematic rendering in ct: a novel, lifelike 3d visualization technique," *American Journal of Roentgenology*, vol. 209, no. 2, pp. 370–379, 2017.

[50] E. Veach, *Robust Monte Carlo methods for light transport simulation*. Stanford University, 1998.

[51] Museth, Ken, "NanoVDB: A GPU-friendly and portable VDB data structure for real-time rendering and simulation," in *ACM SIGGRAPH 2021 Talks*, 2021, pp. 1–2.

[52] Xu, Rui and Wunsch, D., Ii , "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[53] J. H. Mueller, T. Neff, P. Voglreiter, M. Steinberger, and D. Schmalstieg, "Temporally adaptive shading reuse for real-time rendering and virtual reality," *ACM Transactions on Graphics*, vol. 40, no. 2, pp. 1–14, 2021.

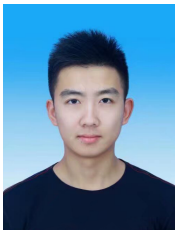
[54] P. Klacansky. Open scientific visualization datasets. [Online]. Available: https://klacansky.com/open-scivis-datasets

[55] A. Rosset and O. Ratib. Osirix dicom image library. [Online]. Available: https://www.osirix-viewer.com/

[56] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[57] J. Hu, C. Yu, H. Liu, L. Yan, Y. Wu, and X. Jin, "Deep real-time volumetric rendering using multi-feature fusion," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–10.

Chunxiao Xu received his B.Sc. degree from Shandong University in 2019. He then started to pursue his M.Sc. and Ph.D. degree at University of Science and Technology of China. Since July 2019, he has been conducting his scientific research work of GPU based 3D volume rendering and applications of VR and AR, at Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, Suzhou, China. He has developed a prototype of an advanced medical visualization engine using CUDA and is currently working on 3D volume rendering with enhanced illumination model for real-time VR and AR applications.



Haojie Cheng is currently a research fellow at the Yong Loo Lin School of Medicine, National University of Singapore (NUS). He is primarily conducting research works at the College of Design and Engineering, NUS. Before he joined NUS, he received his Ph.D. degree from the University of Science and Technology of China, Hefei, China. His research interests include realistic scene rendering, GPU-based interactive 3D medical visualization, the preoperative surgical simulation and navigation using VR and AR.

Zhenxin Chen received his B.Sc. degree from Harbin Institute of Technology, Harbin, China, in 2016. Then he received his M.Sc. degree from Shandong University, Jinan, China, in 2019. He is currently pursuing his Ph.D. degree in the Healthcare Information Technology Innovation Center, Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, Suzhou, China. His research interests include interactive 3D medical visualization, biological signal and medical image processing.

Jiajun Wang received his M.Sc. degree from Southeast University in 2019. During this period, he conducted his research on acoustic emission in the Research Center of Condition Monitoring and Fault Diagnosis. Then he started to pursue his Ph.D. degree at Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences. He is currently focusing on the research of 3D medical visualization and artificial intelligence for clinical applications. He is also performing his work as a software engineer.

Yibo Chen received his M.Sc. degree in Software Engineering from Harbin Institute of Technology, and an engineering degree in computer science at ISIMA in 2011. In May 2015, he obtained his Ph.D. degree from University of Blaise Pascal, Clermont-Ferrand II, France. Since 2015, he has been working in Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, and is now working as an associated research fellow. His main interests include intelligence in the IoT, cloud computing, embedded system and biomedical image processing.

Lingxiao Zhao worked as a M.Sc. student and then a Ph.D. researcher in the Computer Graphics and Visualization group of the Delft University of Technology (TU Delft), the Netherlands, from 2002 to 2008. He received his M.Sc. and Ph.D. degrees in 2004 and 2011 respectively. During this period, he conducted his research work as a part of the advanced development of CT imaging and visualization for computer-aided diagnosis in virtual colonography inside of Philips Healthcare, Best, the Netherlands. In 2009, he began his career in commercial healthcare software development. In 2011, he joined the Imaging Dept. of Philips Healthcare and continued to work for the R&D of medical visualization algorithms and software products. In 2014, he moved to Philips Research, HTC, Eindhoven, the Netherlands, and contributed for two incubating projects of Philips Healthcare. Since September 2015, he started his new position in Suzhou Institute of Biomedical Engineering and Technology (SIBET), Chinese Academy of Sciences. He is currently a professor and the director of the Healthcare Information Technology Innovation Center in SIBET. He is also a guest professor in the Biomedical Engineering Dept. of University of Science and Technology of China. His current research mainly focuses on image processing, 3D graphics and visualization, cloud computing, machine learning and deep learning.