Low-Pass Filtered Volumetric Shadows

Marco Ament, *Student Member, IEEE*, Filip Sadlo, *Member, IEEE*, Carsten Dachsbacher,
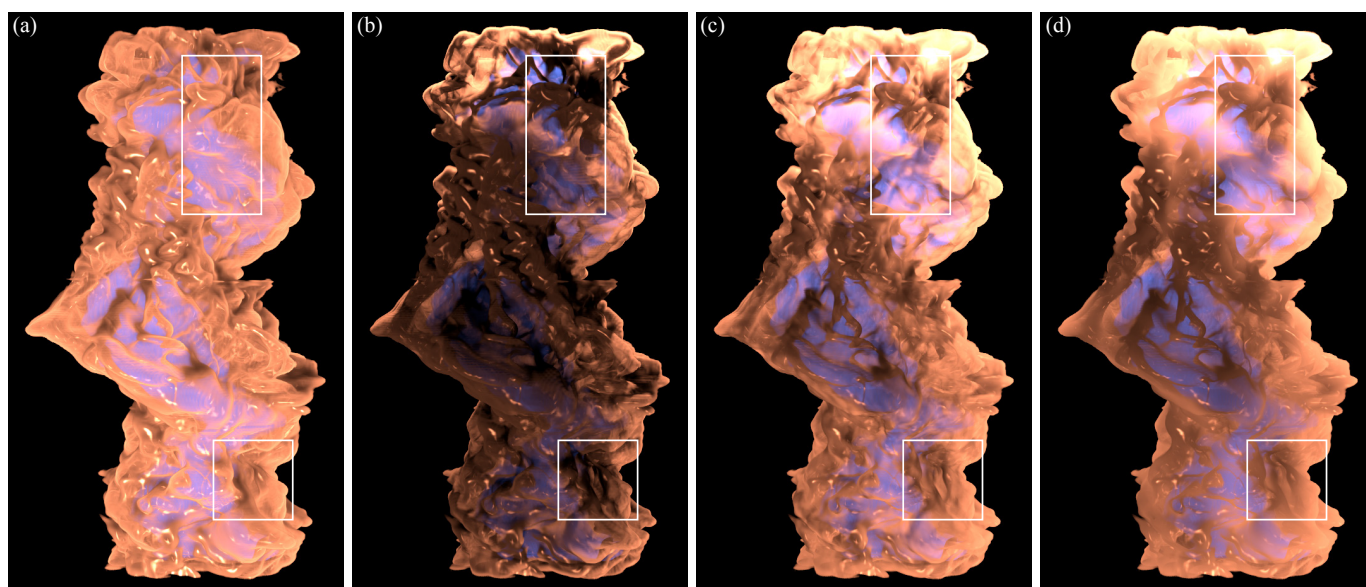and Daniel Weiskopf, *Member, IEEE Computer Society*

Fig. 1. Volume renderings of a numerical flow simulation data set. (a) Volumetric ambient occlusion and specular highlights provide only local visual cues for perception of spatial depth. (b) Illumination from three orthodirectional light sources with single scattering provides global visual cues but introduces disturbing patterns and is too dark. The same setup visualized with our novel model using (c) a small and (d) a large volumetric region for low-pass filtering to avoid disturbing high-frequency shadow patterns.

Abstract—We present a novel and efficient method to compute volumetric soft shadows for interactive direct volume visualization to improve the perception of spatial depth. By direct control of the softness of volumetric shadows, disturbing visual patterns due to hard shadows can be avoided and users can adapt the illumination to their personal and application-specific requirements. We compute the shadowing of a point in the data set by employing spatial filtering of the optical depth over a finite area patch pointing toward each light source. Conceptually, the area patch spans a volumetric region that is sampled with shadow rays; afterward, the resulting optical depth values are convolved with a low-pass filter on the patch. In the numerical computation, however, to avoid expensive shadow ray marching, we show how to align and set up summed area tables for both directional and point light sources. Once computed, the summed area tables enable efficient evaluation of soft shadows for each point in constant time without shadow ray marching and the softness of the shadows can be controlled interactively. We integrated our method in a GPU-based volume renderer with ray casting from the camera, which offers interactive control of the transfer function, light source positions, and viewpoint, for both static and time-dependent data sets. Our results demonstrate the benefit of soft shadows for visualization to achieve user-controlled illumination with many-point lighting setups for improved perception combined with high rendering speed.

Index Terms—Direct volume rendering, volume illumination, soft shadows, filtered shadows, summed area table

◆

1 INTRODUCTION

Shadows can provide important visual cues to improve the perception of objects and features. Several user studies have shown that

- *Marco Ament is with Karlsruhe Institute of Technology, Germany.*
  E-mail: ament@kit.edu.
- *Filip Sadlo is with University of Stuttgart, Germany.*
  E-mail: sadlo@visus.uni-stuttgart.de.
- *Carsten Dachsbacher is with Karlsruhe Institute of Technology, Germany.*
  E-mail: dachsbacher@kit.edu.
- *Daniel Weiskopf is with University of Stuttgart, Germany.*
  E-mail: weiskopf@visus.uni-stuttgart.de.

shadows are beneficial to determine the size and composition of objects [44] or to position objects relative to each other with improved accuracy [17]. The study by Langer and Bülthoff [24] demonstrated that human perception associates low luminance with spatial depth. For direct volume rendering (DVR), Lindemann and Ropinski [27] conducted a user-study that evaluated several advanced illumination techniques for interactive DVR, where participants had to estimate the relative depth and size of volumetric features. It was shown that subtle shadows are beneficial compared to local illumination [33].

One of the simplest methods to compute shadows is volumetric ambient occlusion [7, 15, 36, 37, 41] that estimates local extinction in a small ambient sphere around each sample point. Depending on the radius, occluders of different size create local soft shadows. Moreover, volumetric ambient occlusion can be computed very efficiently by summing up the local extinction with summed area tables (SAT) [39]. Ambient occlusion supports the visual identification of concavities and structures that have about the same size as the ambient radius. In con-

trast, directional shadows, e.g., from a single scattering model [29], account for occluders on a large scale and over long distances. However, on a small scale, directional shadows can lead to high-frequency patterns of alternating illuminated and shadowed areas. In computer graphics, crepuscular rays play an important role [9], but in visualization, they can be visually distracting.

A common alternative are directional soft shadows due to extended light sources or multiple scattering. For interactive DVR, simplified illumination models have been developed that approximate multiple scattering by sampling and integrating over volumetric primitives like cones [21, 39, 40, 43] or tubes [1] as illustrated in Figures 2(a)–(c). Common to all these methods is the repeated execution of local operations like blurring or averaging, which act as a low-pass filter inside the volumetric primitives. Although this effectively smoothes hard shadows, it can be computationally expensive because the volumetric primitives must be sampled either with some kind of shadow rays or slices. Moreover, these sampling schemes must be performed once for each light source, which strongly decreases performance.

For visualization, soft shadows are often more preferred compared to hard shadows, because the latter can lead to strong visual masking effects [11], which can diminish the perception and visual detectability of spatial details. Typically, hard shadows exhibit high contrast and they create disturbing visual patterns that cannot be distinguished easily from the actual volumetric features that cast the shadows. With time-dependent data sets or moving light sources, both effects often lead to noticeable visual flickering. The goal of this paper is to employ low-frequency shadows to reduce these masking effects for improved perception in volume visualization.

Our contribution is a novel optical model for directional soft shadows that builds on efficient spatial filtering and does not require expensive shadow ray marching. In particular, the performance of our approach neither depends on the sampling rate of shadow rays nor on the width of the filter as opposed to previous techniques. We can directly control the fuzziness of volumetric shadows for each ray sample by means of a single user-controlled parameter. We show that even a simple box filter is capable of generating high-quality shadows of arbitrary softness. Previous approaches [1, 39] employed SATs that are aligned with the data set to achieve similar effects. However, the main idea of our approach is to carefully align the SATs with the directional and point light sources, as briefly illustrated in Figures 2(d) and (e), which enables higher computational efficiency. As a unique feature, our approach supports up to six orthodirectional light sources without multiple shadow ray marching by reusing the same SAT, which offers high flexibility for many-point lighting setups with high performance and comparatively low memory consumption. Only if more than six ortho-directional light sources are required, multiple SATs need to be computed and stored. Furthermore, no precomputation step is required, which allows one to interactively explore, illuminate, and visualize even time-dependent data sets without any preprocessing.

2 RELATED WORK

The radiative transfer equation (RTE) by Chandrasekhar [4] describes physically based light transport in participating media. However, in visualization, simplified optical models [29] are often employed due to their higher performance. The survey by Jönsson et al. [20] provides a comprehensive overview of advanced illumination techniques that are commonly employed in interactive DVR.

Local shadows are a common alternative to directional shadows. Stewart [41] computes the local occlusion of diffuse ambient light in the vicinity of each point to shade isosurfaces. Moreover, ambient occlusion can also be employed for illustrative visualizations as demonstrated by Ruiz et al. [37]. Hernell et al. [14, 15] compute attenuation in the direct neighborhood of each voxel on a multiresolution grid by casting shadow rays over the sphere of directions, which can be expensive and must be recomputed whenever the transfer function changes. To obtain a vicinity representation that is independent of the transfer function Ropinski et al. [36] employ clustered histograms of the surrounding scalar values of each voxel. However, it can take hours to precompute and cluster the histograms of each data set. The fastest

ambient occlusion methods rely on the observation that no directional information of the surrounding occluders is accounted for. This allows one to employ order-independent summation of the extinction in the neighborhood of each voxel as shown by Díaz et al. [7] and Schlegel et al. [39], who exploit SATs [6] for fast summation. In addition, the computation of SATs can be strongly accelerated with the GPU-friendly algorithm by Hensley et al. [13].

Directional shadows provide additional cues for large-scale occlusion and are an important building block of many illumination algorithms in DVR. Behrens and Ratering [2] computed attenuation from a distant light source with a slice-based approach and superimposed a shadow volume with the original scalar field to render directional shadows. A different approach by Zhang and Crawfis [47] computes shadows with volume splatting. A more GPU-friendly technique is described by Hadwiger et al. [12], who adapt deep shadow maps [28] to GPU-based DVR. Salvi et al. [38] introduce adaptive volumetric shadow maps that store a piecewise linear approximation of the transmittance along shadow rays instead of a dense uniform array. For interactive DVR, Weber et al. [45] present a many-light algorithm that employs adaptive cube shadow maps [38] to efficiently evaluate transmittance for each virtual light source to compute multiple scattering. However, the method requires progressive updates when the transfer function changes. Kronander et al. [23] compute directional visibility information for each voxel on a multiresolution grid with truncated spherical harmonics at the cost of rather high storage requirements.

Soft shadows can occur in the presence of area light sources or when light arrives from more than one direction, e.g., due to multiple scattering. Kniss et al. [21] approximate multiple scattering by sampling a cone pointing toward the light source and forward-directed scattering is estimated by repeated blurring operations using half angle slicing, which can also be parallelized [30]. In general, sampling a volumetric cone is a common element of several subsequent papers. Schott et al. [40] present directional occlusion shading that computes attenuation in a cone pointed toward the observer to approximate backward-directed scattering for a single head light. Šoltészová et al. [43] presented a more general approach that iteratively filters opacity within tilted light cones on a view-aligned slice stack to support more directions for illumination. Patel et al. [32] further extended the latter method for polygonal models combined with a faster convolution computation. However, both approaches [32, 43] are restricted in the placement of the light sources to a hemisphere. In contrast, our approach allows for arbitrary placement of light sources. Ropinski et al. [35] blur light within a given cone centered around the incoming light direction and utilize an illumination volume for ray casting. Schlegel et al. [39] efficiently aggregate extinction in an order-independent fashion by approximating a cone with a series of cuboids that are used to evaluate a volume-aligned SAT. However, in addition to the computation of the SAT, their method still requires the sampling of the shadow cone. In contrast, our main contribution is to employ light source-aligned SATs to avoid this expensive sampling. Crassin et al. [5] sample a prefiltered voxelized cone to estimate occlusion for geometry-based scenes by means of mipmaps and a hierarchical octree representation. Ament et al. [1] employ preintegrated scattering to compute soft shadows along a tube of finite radius. However, similar to all cone-based techniques, the tube needs to be sampled by casting rays to all light sources. In contrast, our method does not require any shadow ray marching at all and supports efficient handling of multiple light sources, which is important for three-point lighting setups [49]. Traditionally, soft shadows and their efficient computation play also an important role in surface-based graphics, e.g., by employing shadow maps with percentage-closer filtering [10]. Lauritzen [25] employed SATs in combination with variance shadow maps; however, filtering is applied to the moments of the depth distribution for smooth 2D shadows, whereas our approach aims for volumetric shadows and we refer the reader to the course notes by Eisemann et al. [8] for a more detailed discussion.

Sundén et al. [42] employ plane sweeping in image space to render advanced lighting effects with low memory demands, but only one point or directional light source is supported. The volume ren-
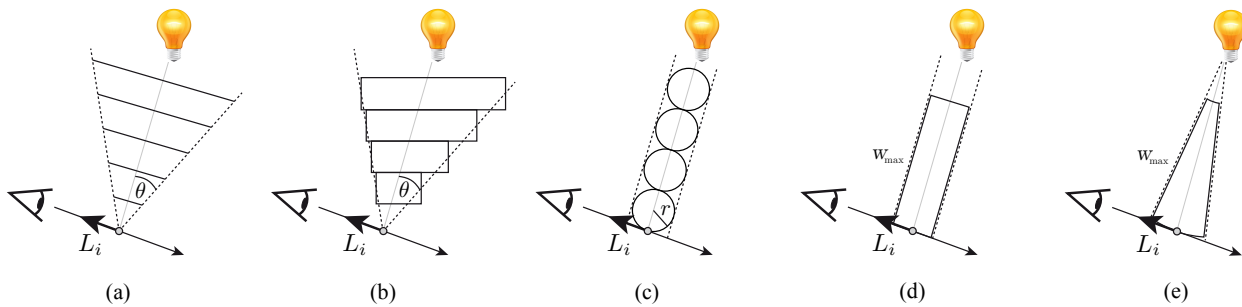
Fig. 2. Illustrative comparison of techniques for approximating soft shadows in interactive DVR. (a) Slice-based sampling of a cone with repeated blurring operations. (b) Cuboid-based sampling of a cone with a volume-aligned SAT [39]. (c) Sphere-based sampling of a tube with preintegrated scattering [1]. (d) Novel technique for directional light sources. Our first approach introduces a light source-aligned SAT and does not require additional sampling with ray marching. (e) Novel technique for point light sources. Our second approach introduces a spherical SAT located at the position of the point light source and also does not require any shadow ray marching.

dering technique by Zhang and Ma [48] approximates global illumination with a convection–diffusion model by assuming an optically thick medium. The method achieves interactive performance, but the softness of the shadows cannot be controlled. For visualization, full global illumination of participating media is usually too expensive, too hard to control, or does not allow interactive control over all parameters, which can be very time-consuming for data exploration. For a detailed discussion of full global illumination of participating media, the reader is referred to the survey of Cerezo et al. [3]. Wyman et al. [46] employ spherical harmonics to precompute global illumination, but the method is limited to isosurfaces. Interactivity of volumetric Monte Carlo simulations is usually achieved by heavy restrictions, e.g., by constraining the expensive scattering computations to selected isosurfaces [34] or by relying on progressive refinement [22]. Jönsson et al. [19] present an algorithm that allows one to interactively edit the transfer function for volumetric photon mapping [18] by tracking the history of photon trajectories. However, time-dependent data sets or moving light sources still require an expensive recomputation of all photons.

3 BASICS OF VOLUMETRIC ILLUMINATION

Before we introduce our approach in Section 4, we briefly recapitulate the notation and basic principles of volumetric illumination. We build our approach on the single scattering illumination model [29] in participating media. The total radiance $L(x,\omega)$ at position x in direction ω is the sum of the attenuated radiance $L_b(x_b,\omega)$ from a boundary condition at position x_b and the integrated in-scattered radiance $L_i(x,\omega)$:

$$L(x,\omega) = T(x_b,x)L_b(x_b,\omega) + \int_{x_b}^{x} T(x',x)\sigma_t(x')L_i(x',\omega)\,dx', \quad (1)$$

where $\sigma_t(x)$ is the extinction coefficient. The transmittance $T(x_1,x_2)$ between any two points x_1 and x_2 is:

$$T(x_1,x_2) = e^{-\tau(x_1,x_2)} = e^{-\int_{x_1}^{x_2} \sigma_t(x')\,dx'}, \quad (2)$$

where $\tau(x_1,x_2)$ is the optical depth. Furthermore, in Eqn. (1), $L_i(x,\omega)$ describes the amount of radiance that arrives from all directions at point x and that is scattered into direction ω:

$$L_i(x,\omega) = \int_{\Omega} P(x,\omega',\omega)T(x_b,x)L_b(x_b,\omega')\,d\omega', \quad (3)$$

where Ω denotes the sphere of all directions and the normalized phase function $P(x,\omega',\omega)$ describes the probability density of radiance being scattered from the incident direction ω' to direction ω. With single scattering, only one scattering event is accounted for; hence, in Eqn. (3), incoming radiance depends only on the attenuated radiance $L_b(x_b,\omega)$ from the boundary conditions such as light sources or the background.

4 THE TECHNIQUE

In this section, we present our novel technique for efficient soft shadows. First, we discuss the theory of our filtered volumetric shadows in Section 4.1, before we introduce an efficient algorithm that computes volumetric shadows of arbitrary softness in Sections 4.2 and 4.3. Afterward, we show how interactive DVR with our shadow technique is performed in Section 4.4. Since our approach is related to a series of previous methods that also employ filter-like operations, we provide a discussion of similarities and differences in Section 4.5.

4.1 Filtered Volumetric Shadows

We begin our approach with Eqn. (3). Traditionally, radiance $L_b(x,\omega)$ from the boundary conditions is attenuated by integrating the extinction along a 1D ray, i.e, the optical depth τ, as shown in Eqn. (2). The core element of our novel technique is to substitute τ with the filtered optical depth $\bar{\tau}$:

$$\tau \rightarrow \bar{\tau}(x_1,x_2) = \frac{1}{A}\int_{-U}^{U}\int_{-V}^{V} \tau\left(x_1'(u,v),x_2'(u,v)\right)f(-u,-v)\,dv\,du, \quad (4)$$

where f is an arbitrary 2D filter function over a finite area A with parameter range $[-U,U]\times[-V,V]$ (both measured in unit distances). We specify the domain for filtering with two parametrized surface patches $x_1'(u,v)$ and $x_2'(u,v)$ that will be defined later in more detail to control the type of the light source and the shape of the soft shadows. For now, we only require that $x_1 = x_1'(0,0)$ and $x_2 = x_2'(0,0)$.

Figure 3 shows two examples of how Eqn. (4) can be employed for directional and point light sources, respectively. For each sample point of an eye ray, a local area patch is aligned with its normal vector toward the light source. The area patches span volumetric primitives (cuboid and spherical pyramid) that are conceptually sampled with shadow rays to compute $\tau\left(x_1'(u,v),x_2'(u,v)\right)$. Afterward, this 2D distribution is convolved with the filter f as indicated by the concentric circles in Figure 3 and the filter controls the relative contribution of τ for total attenuation. In this way, soft shadows can be achieved with a low-pass filter and the softness can be controlled by the size and shape of the filter kernel.

The computation of the filtered optical depth in Eqn. (4) requires the solution of an integral over a 3D domain: For each pair of points x_1' and x_2', the solution of the 1D integral in Eqn. (2) is needed. A numerical solution of the integrals could be obtained with Monte Carlo integration or, as in our case, with Riemann sums, which fits well to our approach with SATs in the next section:

$$\bar{\tau}(x_1,x_2) \approx \frac{1}{A}\sum_{i=1}^{I}\sum_{j=1}^{J} \tau\left(x_1'(u_i,v_j),x_2'(u_i,v_j)\right)f(-u_i,-v_j)\Delta v\Delta u, \quad (5)$$

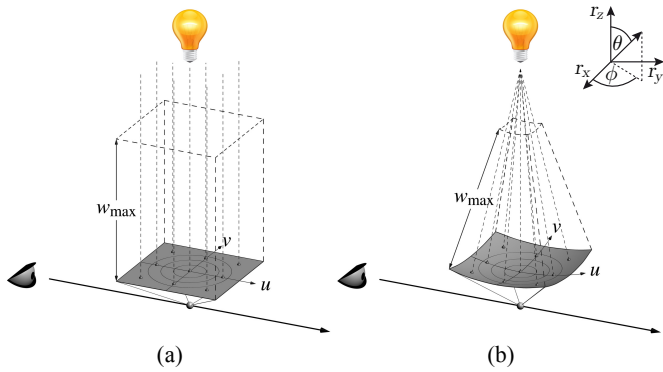where I and J are the number of discrete samples of the 2D filter ker-

Fig. 3. Illustration of filtered volumetric shadows for one sample point of the eye ray. (a) For directional light sources, a planar area patch spans a cuboid pointed toward the light source. Conceptually, the patch is discretized and for each point a shadow ray is casted to compute the optical depth τ. Then, a 2D filter is employed to compute $\bar{\tau}$ according to Eqn. (4). (b) For point light sources, a curved area patch spans a spherical pyramid and the filter is applied on the curved patch.

nel. Similarly, the non-filtered optical depth becomes:

$$\tau\left(x_1', x_2'\right) \approx \sum_{k=1}^{K} \sigma_t \left(x_1' + w_k \frac{x_2' - x_1'}{\left\|x_2' - x_1'\right\|}\right) \Delta w, \qquad (6)$$

where K is the number of discrete ray samples. In general, computing the solutions of Eqns. (5) and (6) is very expensive because three dimensions need to be sampled in contrast to only one dimension for classic single scattering. Moreover, the computation must be performed for each eye ray sample. Even when a shadow cache is employed to accelerate rendering, an expensive recomputation is necessary whenever the transfer function, lighting setup, or time step changes. Therefore, we introduce an efficient technique that avoids this repeated shadow ray marching in the next section.

4.2 Efficient Box Filtering

Our goal for filtered volumetric shadows is an efficient and direct control over the softness of shadows. A user must be able to suppress high-frequency patterns due to hard shadows, but in a controlled manner without introducing excessive blur. Therefore, we focus on low-pass filters and their efficient application. A simple low-pass filter is the convolution with a box kernel, i.e., each sample in the filter domain is weighted with a constant value.

In interactive DVR, directional and point light sources are probably the two most commonly employed types of light sources because they allow flexible lighting setups and they can be sampled efficiently. Directional light sources generate parallel volumetric shadows and provide illumination from far distant emitters. In contrast, point light sources generate perspective volumetric shadows and they can be placed inside the data set for close-range illumination effects. In the following sections, we show how box filtering can be computed efficiently for both types of light sources.

4.2.1 Directional Light Sources

By definition, light from a directional light source arrives from parallel directions at each point. As illustrated in Figure 3(a), we model the parallel shape of the resulting shadows with a cuboid-shaped filter domain that is spanned by the following two surface patches:

$$x_1'(u,v) = x_1 + u \cdot r_u + v \cdot r_v, \qquad (7)$$
$$x_2'(u,v) = x_1'(u,v) + w_{\max} \cdot r_w, \qquad (8)$$

where $w_{\max}$ is the distance from $x_1'(0,0)$ to the intersection point with the bounding box of the data set and r_u, r_v, and r_w are orthonormal

unit vectors:

$$r_u \times r_v = r_w = l, \qquad (9)$$

where l is the light vector that points in direction of the light source. Then, the set of points:

$$C = \left\{ x_1'(u,v) + w \cdot r_w : (u,v,w) \in [-U,U] \times [-V,V] \times [0, w_{\max}] \right\} \qquad (10)$$

defines a cuboid that is aligned with the directional light source as illustrated in Figure 3(a). Furthermore, we set the generic filter function in Eqn. (5) to a box kernel:

$$f(u,v) = 1. \qquad (11)$$

Note that the normalization factor $1/A$ is already included in Eqn. (5). Then, the box-filtered optical depth becomes:

$$\bar{\tau}(x_1, x_2) \approx \frac{1}{A} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \sigma_t \left(x_{i,j,k} \in C\right) \Delta V, \qquad (12)$$

where $\Delta V = \Delta u \Delta v \Delta w$ is the size of the volume element. Essentially, Eqn. (12) is a 3D box filter with the cuboid C as filtering domain. However, the triple sum in Eqn. (12) is still expensive to calculate with naive looping. Therefore, we employ a 3D-SAT and align it with the directional light source so that one of its six faces is orthogonal to the light vector l. The bounding box of our SAT contains the entire volume, but it is not aligned with the orientation of the data set and has its own Cartesian coordinate system defined by the unit vectors r_u, r_v, and r_w, according to Eqn. (9). Note that this is a crucial element of our approach and is different from all previous approaches that also employed SATs for DVR [1, 7, 39]. For the sake of clarity, we call our approach Cartesian SAT (CSAT) for the remainder of this paper. The CSAT sums up $\sigma_t(x_{i,j,k}) \Delta V$ for a set of discrete positions $x_{i,j,k}$ of the data set, but in its own frame of reference and once it is computed, the box-filtered optical depth simplifies to:

$$\bar{\tau}(x_1, x_2) \approx \frac{1}{A} \text{CSAT}(C), \qquad (13)$$

where $\text{CSAT}(C)$ evaluates the cuboid region C of the CSAT in constant time with only eight elementary data references. No additional marching along shadow rays is required as opposed to the methods by Schlegel el al. [39] or by Ament et al. [1] as illustrated in Figures 2(b) and 2(c), respectively. Furthermore, we employ a quadratic surface patch for filtering, which leads to $-V = -U$, $V = U$, and $J = I$. In this way, only a single parameter is required to control the size of the filtering domain and thereby the softness of the resulting shadows. Another important feature of our approach is that computation time is independent of the filter size and thereby independent of the softness of the shadows.

4.2.2 Orthodirectional Light Sources

A straightforward extension of the previously introduced algorithm allows us to compute box-filtered shadows for up to six orthogonal directional light sources with the same CSAT. In the following, we denote the i-th light source with a subscript index i and we set the light vector of the primary light source to $l_1 = l$. With six mutual orthogonal light vectors $l_i \cdot l_j = 0$ for $i, j = 1, \ldots, 6$ and $i \neq j$, we can define six coordinate systems:

$$r_{u,i} \times r_{v,i} = r_{w,i} = l_i. \qquad (14)$$

Following the same steps as in the previous section, we obtain similar expressions for Eqns. (7), (8), and (10). Finally, the box-filtered optical depth for all six directional light sources becomes:

$$\bar{\tau}_i(x_1, x_2) \approx \frac{1}{A} \text{CSAT}(C_i), \qquad i = 1, \ldots, 6. \qquad (15)$$

In this way, we can efficiently compute filtered shadows for many-light settings to illuminate a data set from six orthogonal directions and to better convey the 3D shape of volumetric features. Note that if more than six orthodirectional light sources are required, multiple CSATs become necessary. However, in many cases, three light sources already yield good results.

4.2.3 Point Light Sources

Emitted light from a point light source naturally follows a spherical geometry. As illustrated in Figure 3(b), we model the perspective shape of the resulting shadows by using a spherical pyramid instead of a cuboid as filtering domain. Then, we transform the spherical geometry of propagating rays to computational space with a Cartesian coordinate system where rays propagate again in parallel and we can employ 3D-SATs for point light sources as well.

We define a spherical coordinate system with azimuth angle ϕ, polar angle θ, and radius r and set its origin $(\phi = 0, \theta = 0, r = 0)$ to the position x_L of the point light source. First, we span the tangential plane at sample point x_1 with spherical coordinates $x_1^S = (\phi_1, \theta_1, r_1)$:

$$x_t(u,v) = x_1 + u \cdot r_\phi(\phi_1) + v \cdot r_\theta(\phi_1, \theta_1), \qquad (16)$$

where r_ϕ and r_θ are orthonormal tangential vectors:

$$r_\phi(\phi) = -\sin(\phi)r_x + \cos(\phi)r_y, \qquad (17)$$
$$r_\theta(\phi, \theta) = \cos(\theta)\cos(\phi)r_x + \cos(\theta)\sin(\phi)r_y - \sin(\theta)r_z, \qquad (18)$$

and r_x, r_y, and r_z are the orthonormal vectors defining the Cartesian world space. Furthermore, the radial unit vector r_r points in the opposite direction of the light vector l:

$$r_\phi \times r_\theta = r_r = -l. \qquad (19)$$

In the second step, we project all points of the tangential plane $x_t(u,v)$ onto the surface of the sphere and we get two curved area patches:

$$x_1'(u,v) = x_L + R \cdot \frac{x_t(u,v) - x_L}{\|x_t(u,v) - x_L\|}, \qquad (20)$$
$$x_2'(u,v) = x_1'(u,v) - w_{\max} \cdot r_r, \qquad (21)$$

where $R = \|x_t(0,0) - x_L\|$ is the radial distance from the sample position $x_1 = x_t(0,0)$ to the point light source x_L and $w_{\max}$ is the distance from the sample position to the intersection point with the bounding box of the data set. Then, the set of points:

$$S = \left\{ x_1'(u,v) - w \cdot r_r : (u,v,w) \in [-U,U] \times [-V,V] \times [0, w_{\max}] \right\} \qquad (22)$$

defines a (cropped) spherical pyramid with its peak located at the position of the point light source as illustrated in Figure 3(b). Note that we use this more complicated description in Cartesian coordinates to obtain spherical pyramids of equal base area A in each direction, which would not be possible in spherical coordinates. However, we can now transform all points $x \in S$ from Cartesian coordinates to spherical coordinates $x^S \in S$. Then, we can write the box-filtered optical depth for point light sources as follows:

$$\bar{\tau}(x_1, x_2) \approx \frac{1}{A} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \sigma_t \left(x_{i,j,k}^S \in S \right) \Delta V, \qquad (23)$$

where $\Delta V = r^2 \sin(\theta) \Delta r \Delta \theta \Delta \phi$ is the size of the spherical volume element. We observe that Eqn. (23) has a form similar to Eqn. (12) and we can also employ a 3D-SAT to accelerate filtering. However, in this case, summation needs to be performed over a spherical pyramid and not a cuboid. Therefore, we employ a SAT with spherical coordinate system r_ϕ, r_θ, and r_r, according to Eqn. (19). Moreover, in contrast to the CSAT, the origin of this spherical SAT (SSAT) is located at the position of the point light source. The spherically shaped bounding box of the SSAT contains again the entire volume. However, if the point light is located outside of the data set, only a fraction of the entire sphere is required. Note that the SSAT is actually a regular grid in the computational domain where the SSAT sums up $\sigma_t(x_{i,j,k}^S) \Delta V$ for a set of discrete positions. However, these positions must be transformed to Cartesian world space before sampling the data set. As soon as the SSAT is computed, the box-filtered optical depth becomes:

$$\bar{\tau}(x_1, x_2) \approx \frac{1}{A} \text{SSAT}(S), \qquad (24)$$

where SSAT(S) evaluates the spherical pyramid S of the SSAT in constant time and again no additional shadow rays are required. Note that, in contrast to orthodirectional light sources, every point light source requires its own SSAT. Therefore, for large data sets, the number of point light sources can be limited by computation time and memory requirements; in this case, a fallback to orthodirectional light sources with a single CSAT is possible.

4.3 Higher-Order Filtering

The Fourier transform of a box kernel is a Sinc function and its infinite support and oscillations in frequency domain could become problematic for certain frequencies of the filtered function τ of Eqn. (5). Therefore, we evaluate higher-order filters to compare their quality and performance with box filtering. In general, filtering with higher-order kernels f cannot be achieved directly with a 3D-SAT because the weight of the filter depends on the position on the surface patch. However, instead of shadow ray marching for each pair of points x_1' and x_2' in Eqn. (4), we can still employ the same 3D-SATs of the previous sections to quickly approximate the non-filtered optical depth τ of the innermost 1D integral. For the directional light source, we decompose the cuboid $C = \bigcup C_{i,j}$ of Eqn. (10) into a set of disjoint smaller cuboids $C_{i,j}$. By looking again at our general model of Eqn. (5), we obtain the filtered optical depth with:

$$\bar{\tau}(x_1, x_2) \approx \frac{1}{A} \sum_{i=1}^{I} \sum_{j=1}^{J} \text{CSAT}(C_{i,j}) f(-u_i, -v_j). \qquad (25)$$

Similarly, the spherical pyramid of Eqn. (22) can be decomposed with $S = \bigcup S_{i,j}$. In this way, the repeated shadow ray marching can be avoided, but the summation over the surface area of the 2D filter still needs to be computed explicitly, in contrast to box filtering.

4.4 Interactive Direct Volume Rendering

For volume rendering, we assume that the data set is stored on a discrete Cartesian grid that samples a scalar field $s : \mathbb{R}^3 \rightarrow \mathbb{R}$ and a transfer function maps these scalar values to color and extinction.

Our algorithm consists of two passes that are both summarized in Algorithm 1. First, depending on the type of light source, the CSAT or SSAT is recomputed whenever the transfer function, lighting setup, or data set changes. Then, we compute the filtered optical depth with filter sizes A_{CSAT} and A_{SSAT} by evaluating the CSAT or SSAT, following Eqns. (15) and (24), to compute the filtered optical depth according to Eqn. (4) and to store the resulting transmittance in a shadow cache. Similar caches are employed regularly in interactive DVR [1, 2, 35, 39]. We follow this practice to conduct a fair comparison, although our algorithm is capable of interactive frame rates even without any caching. Note that for all six orthodirectional lights, the CSAT must be computed only once, whereas the SSAT must be computed for each point light.

In the second pass, ray casting from the camera is employed. At each ray sample position x, the emitted radiance L_b from all active light sources is attenuated with the transmittance from the shadow cache and multiplied with the phase function, according to Eqn. (3). In an alternative implementation without any shadow cache, the CSAT and SSAT would be directly evaluated at this place and the first pass would only recompute the SATs. Finally, the color and extinction are read from the transfer function for compositing.

4.5 Discussion

There are several previous methods [1, 21, 32, 35, 39, 40, 43] that employ building blocks similar to our technique for rendering soft shadows. The one that is closest to our approach is the method by Schlegel et al. [39]. The authors compute a SAT that is aligned with the data set. For each ray sample, a cone is directed toward each light source and is approximated by a series of cuboids with ray marching. For each cuboid, the SAT is evaluated to compute the aggregated extinction. The authors report 40 to 80 cone samples for typical data set sizes and transfer functions with a minimum of at least 12 cone

Algorithm 1 Our low-pass shadows rendering algorithm.

Require: filter size A_{CSAT}, filter size A_{SSAT}, transfer function TF, directional light DL, orthodirectional light DL, point light PL, data set DS, phase function PF, shadow cache SC, radiance L_b (background and emitters)

```
 1: /* First pass */
 2: if hasChanged(TF, DL, PL, DS) then
 3:     CSAT = recomputeCSAT(DL, DS, TF);
 4:     for all orthodirectional light sources OL ∈ DL do
 5:         SC = updateShadowCache(OL, CSAT, A_CSAT);
 6:     end for
 7:     for all point light sources PL do
 8:         SSAT = recomputeSSAT(PL, DS, TF);
 9:         SC = updateShadowCache(PL, SSAT, A_SSAT);
10:     end for
11: end if
12: /* Second pass */
13: for all pixels P do
14:     L = 0; T = 0;
15:     ray = computeEyeRay(P);
16:     while rayCastingNotFinished(ray) do
17:         x = getRayPosition(ray); L_i = 0;
18:         for all active OL ∈ DL do
19:             L_i = L_i+ PF(x, OL) · SC(x, OL) · L_b(OL);
20:         end for
21:         for all active PL do
22:             L_i = L_i+ PF(x, PL) · SC(x, PL) · L_b(PL);
23:         end for
24:         (L, T) = composite(x, TF, L, T, L_i);
25:     end while
26:     return T · L_b + L;
27: end for
```

samples until artifacts become visible. With our approach, we also require the computation of the CSAT or SSAT, which is computationally equivalent to the SAT by Schlegel et al. assuming the same resolution. However, we do not require shadow ray marching and we gain a theoretical performance speedup of one order of magnitude because we require only one lookup in the SAT instead of several ones. In Section 6, we compare this method with our approach.

The method by Ament et al. [1] also shares common elements with our approach because the authors employ a tube-shaped volume and SATs for illumination. Just like Schlegel et al. [39], their SAT is aligned with the volume. However, in contrast to our approach, the tube is approximated by a series of spheres and for each sphere the ambient extinction coefficient must be sampled. In addition, a preintegration table must accessed in each step to evaluate scattering in each sphere. The authors report spherical radii of up to 14 voxels, which results in step sizes of up to 28 voxels. For typical data set sizes, this still requires about 10 sphere samples for shadow ray marching. Again, with our approach, we do not require such sampling steps and we do not require any preprocessing or preintegration tables. However, our method does not explicitly simulate multiple scattering. Therefore, we evaluate the quality and performance of both methods in Section 6.

The techniques by Šoltészová et al. [43] and Patel et al. [32] also achieve efficient soft shadows, albeit by iterative convolution within tilted light cones that are intersected with a view-aligned slice stack. The main advantage of these methods is that no storage for a SAT is required, although some additional memory is also needed for a layered depth image [32]. However, the most significant difference to our approach is that both techniques [32, 43] are restricted in the positioning of light sources. With front-to-back rendering, light sources can be located only in the hemisphere behind the observer or, alternatively, in the hemisphere in front of the observer with back-to-front rendering. Consequently, light sources cannot be anchored with the data set but only with the observer, which is not the case with our approach.

5 IMPLEMENTATION

For our implementation on the GPU, we employ CUDA and we make use of 3D textures for storing the data set, the SATs, and the optional shadow cache. Furthermore, we provide gradient-based Phong illumination [33]. The computation of the CSAT and SSAT on the GPU is performed in computational space, which is a regular grid in both cases. Therefore, the double-buffering algorithm by Hensley et al. [13]

can be directly applied to our approach by simply transforming the positions from computational space into world space. In general, low floating point precision is a known issue for SATs, which can lead to artifacts. Therefore, we implemented the bias technique by Hensley et al. [13] to improve precision. For performance comparison, we also implemented box-filtering without SATs by directly casting shadow rays. Furthermore, we implemented Gaussian filtering with both shadow rays and SATs according to Section 4.3.

In addition, we implemented other volume rendering algorithms that are similar to our approach and that are also capable of generating volumetric shadows for interactive DVR. We compare with volumetric ambient occlusion based on SATs [39] due to its high performance and relevance. For directional shadows, we implemented single scattering as a basic reference and to demonstrate that hard shadows can be problematic for volume visualization. For soft shadows, we implemented the methods by Schlegel et al. [39] and by Ament et al. [1]. All comparison methods are also written in CUDA and for a fair comparison, we also employ shadow and illumination caches.

6 RESULTS

We present results with different data sets exhibiting volumetric structures of different scale and frequency to evaluate our approach. We study our low-pass filtering of shadows with a Fourier analysis of one artificial and six real-world data sets. Furthermore, we compare our soft shadows with single scattering to demonstrate the benefit for visualization in terms of perception, especially for time-dependent data, which is shown in the supplemental video. Finally, we discuss the benefit of our method in terms of quality and performance with respect to two previous state-of-the-art techniques [1, 39].

All results were obtained with an Intel Core i7 3.4 GHz CPU, 16 GB RAM, and an NVIDIA GTX Titan GPU. For a fair comparison, we employ a shadow cache with the same resolution (equal to the corresponding data set) for all results and methods. In particular, for our comparisons with the SAT-based methods by Schlegel et al. [39] and Ament et al. [1], we use the same resolutions for our CSAT and SSAT as for their respective volume-aligned SATs, namely half the resolution of the corresponding data set. In this way, memory consumption and computation times can be compared directly.

Figure 1 shows volume renderings of one time step of a numerical shock wave simulation ($256 \times 256 \times 256$). In Figure 1(a), SAT-based volumetric ambient occlusion [39] and specular highlights are employed for local shadows and illumination to visualize the surface characteristics and concavities, but the perception of spatial depth is limited. In Figure 1(b), three orthodirectional light sources and single scattering are employed to compute directional shadows, which provide additional cues for the depth order, but the hard and dark shadows limit their usefulness. In Figure 1(c), we visualize the same setup with our method using one CSAT and a small filter width of 3 voxels (with respect to the resolution of the data set), which already helps us balance illumination. By increasing the filter width to 9 voxels in Figure 1(d), the remaining shadow streak patterns are filtered and only the large-scale structures cast soft shadows, which help understand the spatial arrangement. The accompanying video further demonstrates the benefit when the data set is rotated relative to the light sources. Due to the shadow cache, ray casting from the camera runs at 92 frames per second (fps) for all four results. However, shadow computation times vary significantly: Ambient occlusion requires 9 ms, single scattering 201 ms, and our approach 17 ms for both filter sizes.

In Figure 4, we evaluate the filtering quality of our low-pass shadows with an artificial data set ($177 \times 27 \times 64$), using signal-processing methodology [31]. A directional light source illuminates a volumetric occluder from above and casts shadows on an opaque plane on the bottom. The opacity of the occluder is modeled with a chirp function and we provide nonfiltered single scattering as a baseline for comparison. Rendering with box filtering and a filter size of 9 voxels leads to extinction of certain frequencies. In the frequency domain, the box filter is a Sinc function and the multiplication with the Fourier transform of the chirp function shows that two frequencies are deleted (red arrows). The inverse Fourier transform illustrates the spatial location
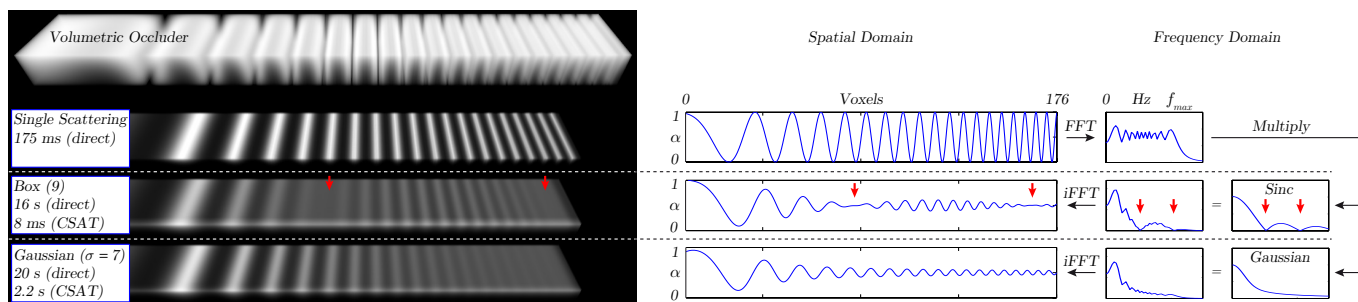
Fig. 4. Comparison of different filter kernels with an artificial data set and incoming light from above of a directional light source. A volumetric occluder casts shadows on the opaque plane on the bottom. The opacity of the occluder is given by a chirp function with varying frequencies.
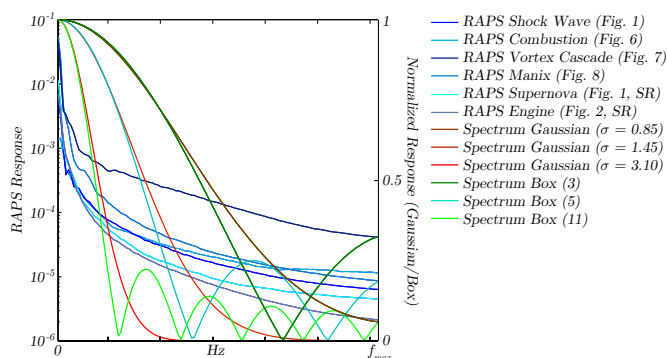

Fig. 5. Radially averaged power spectra (RAPS) of all real-world data sets of this paper and of the supplemental results (SR) as well as spectra of typical box and Gaussian filter kernels.
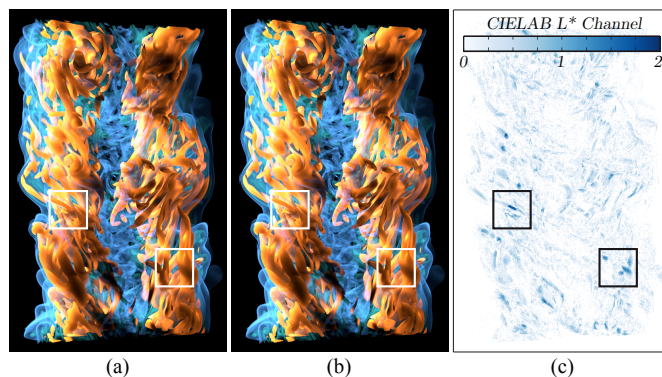

Fig. 6. Visualization of the time-dependent Combustion data set. (a) Box filter: changing the transfer function or lighting setup requires 63 ms. Rendering is performed with 81 fps. (b) Comparable Gaussian filter: changing the transfer function or lighting setup requires 12.2 s. Rendering is also performed with 81 fps. (c) Difference image of (a) and (b).

of the erased frequencies, which closely matches the rendered image. For comparison, we use the Gaussian filter because it has optimal fall-off in both frequency and spatial domains. Multiplication with the Fourier-transformed chirp function leads to continuous damping and the inverse Fourier transform also exhibits steady smoothing. However, there is a significant drop of performance. Direct sampling with shadow rays is about four orders of magnitude slower than our optimized box filtering and even with CSAT acceleration, according to Section 4.3, higher-order filtering remains too expensive for interactive DVR. Moreover, the visual impact of postaliasing remains subtle with box filtering, even with such an artificially constructed data set, because no disturbing artifacts are introduced for visualization.

The previous example was specifically designed as most extreme stress test for filtering. However, real-world data sets do not exhibit such artificial frequency patterns. In Figure 5, we provide a spectral analysis of real-world data sets. According to the Fourier projection-slice theorem [26], an attenuation-only volume rendering from a certain direction can be obtained by extracting (and inverse-transforming) a perpendicular slice that passes through the origin of the frequency domain representation of the classified volume. In our case, we employ the direction of incoming light for integration of optical depth. However, since we are interested in all possible lighting setups, we require an aggregated evaluation of all possible slices (incoming light directions) that pass through the origin, which is equivalent to the entire 3D Fourier transform of the classified data set. For a compact analysis, we compute radially averaged power spectra (RAPS) of the classified and Fourier-transformed data sets of this paper and plot them logarithmically in Figure 5. By comparing to the spectra of typical box and Gaussian filters, it can be observed that most of the energy of all RAPS plots is left of the problematic oscillations of the Sinc functions. Therefore, only negligible energy contributes to postaliasing and, on average, box filtering yields quality close to the Gaussian filter.

In Figure 6, we compare box and Gaussian filtering with the time-dependent Combustion[1] data set ($480 \times 720 \times 120$), showing vorticity magnitude. In Figure 6(a), we employ a box kernel with a width of 9 voxels and a comparable Gaussian kernel ($\sigma = 7$) in Figure 6(b). Furthermore, in Figure 6(c), we provide a false-color image that shows the difference of luminance of Figures 6(a) and (b) in CIELAB color space (values ranging from 0 to 100). The boxes highlight areas where luminance differs the most; however, the absolute difference remains well below 2 % of the possible range, which can be hardly distinguished in the original images. The supplemental video shows that the difference remains in the 10 % range for all 122 time steps.

Figure 7 shows one time step of a discontinuous Galerkin flow simulation [16], resampled on a uniform grid ($529 \times 529 \times 529$) with the λ_2 vortex criterion for volume visualization. The simulation shows the creation of a turbulent cascade from two single, perpendicular large-scale vortices that break up, reconnect, and develop into a cascade of vortices. In Figure 7(a), single scattering and illumination from two point light sources are employed; however, shadows are too dark, which strongly limits visibility. For a fair comparison, we manually reduce optical depth for shadow computation in Figure 7(b) by a factor of ten to allow more light to enter the medium. Although visibility is improved, the hard shadows and vortex features visually interfere with each other due to similar scale, hampering perception of depth. In Figure 7(c), we employ our method with two SSATs and a small filter size of 3 voxels, which already reduces the distracting streak patterns. By successively increasing the filter size to 7 voxels in Figure 7(d) and 11 voxels in Figure 7(e), shadows become increasingly softer, which helps perceive the spatial structure. Furthermore, the performance of

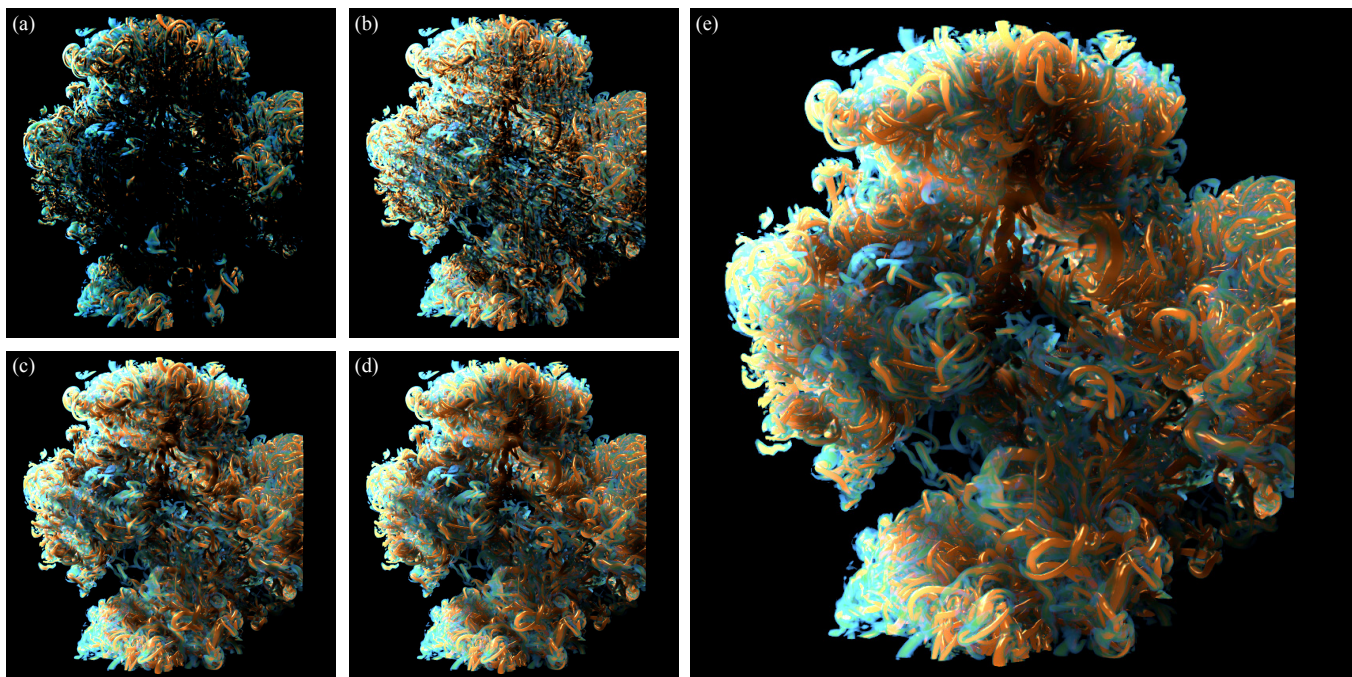[1]http://vis.cs.ucdavis.edu/VisFiles/pages/combustion.php

Fig. 7. Visualization of the time-dependent Vortex Cascade data set using two point light sources. (a) Illumination with single scattering leads to dark and hard shadows. (b) Manual down-scaling of the optical depth for shadow computation illuminates the inner structures, but introduces noise-like patterns due to high-frequency shadows and high contrast. Our low-pass filtered shadows with two SSATs and increasing box filter sizes of (c) 3, (d) 7, and (e) 11 voxels successively remove the disturbing light and shadow streaks. The supplemental video shows the entire animation.

Table 1. Overview of performance for all real-world data sets of this paper using a viewport size of 512^2 pixels. The computation times for changing the transfer function (TF) and moving the light sources (LS) are not included in the rendering performance. Rendering methods: single scattering (SS), SAT-based ambient occlusion (AO), SAT-based cone shadows (CS), SAT-based ambient scattering (AS), our filtered shadows (FS) with box kernel. Light sources: orthodirectional light (OL), point light (PL). The Engine and Supernova are shown in the supplemental results (SR).

Data set	Resolution	Fig.	#(LS)	SAT [MB]		TF change [ms]					LS movement [ms]					Rendering [fps]	
				AO/CS/AS	FS	SS	AO	CS	AS	FS	SS	AO	CS	AS	FS	SS/AO/CS/FS	AS
Shock Wave	$256 \times 256 \times 256$	1	3 OLs	8.0	8.0	201	9	238	187	17	201	-	229	136	17	92	76
Vortex Cascade	$529 \times 529 \times 529$	7	2 PLs	71.0	142.0	2336	68	1221	810	170	2336	-	1185	772	170	55	47
Manix (hard)	$512 \times 512 \times 512$	8	3 OLs	64.0	64.0	2902	61	1484	912	90	2902	-	1458	886	90	67	54
Manix (soft)	$512 \times 512 \times 512$	8	3 OLs	64.0	64.0	2902	61	1484	790	90	2902	-	1458	764	90	67	54
Combustion	$480 \times 720 \times 120$	6	2 PLs	19.8	39.6	855	27	480	405	63	855	-	468	391	63	81	69
Supernova	$432 \times 432 \times 432$	1 (SR)	2 PLs + 1 OL	38.4	115.3	1821	39	912	589	131	1821	-	879	558	131	76	67
Engine	$256 \times 256 \times 110$	2 (SR)	1 PL / 1 OL	3.4	3.4	48	5	37	31	11	48	-	33	28	11	115	97

shadow computation remains independent of the filter size. The benefit of our method can be experienced best in the supplemental video, where the time-dependency of the data set leads to strong visual flickering for single scattering or too small filter sizes. In addition, our method (170 ms) outperforms single scattering (2336 ms) in terms of shadow computation time by more than an order of magnitude.

In Figure 8, we compare our method with SAT-based cone shadows by Schlegel et al. [39] and ambient volume scattering by Ament et al. [1]. We employ the Manix[2] data set ($512 \times 512 \times 512$) and three orthodirectional light sources. Figure 8(a) is visualized with cone shadows using a small cone angle for distinct but slightly smoothened shadows as shown in the highlighted box. For artifact-free rendering, a step size of 15 voxels is necessary for cone marching and shadow computation requires 1484 ms. In Figure 8(b), the same setup is rendered with our method using one CSAT and a filter size of 5 voxels to obtain similar quality and softness of the shadows. Full computation of all shadows requires 90 ms. In Figure 8(c), we employ ambient scattering and visually resemble a similar quality and softness of the shadows. However, since this method approximates multiple scattering, a direct comparison is difficult. We achieved the best visual match by using a scattering albedo of 0.6. The softness of the shadows is inherently

coupled to the ambient radius of the method, which in turn is coupled to the step size of shadow ray marching. For this result, we employed a radius of 8 voxels, which leads to a step size of 16 voxels and shadow recomputation takes 912 ms. Due to the shadow cache, all methods render with similar performance. Our method and cone shadows render at 67 fps, whereas ambient scattering runs only at 54 fps due to the additional view-dependent lookup in the scattering table. We repeat the latter evaluation in Figures 8(d)–(f) but with a larger cone angle, filter size, and ambient radius, respectively, to demonstrate that all three methods can increase the softness of the shadows with similar quality. However, shadow computation times significantly differ again with 1484 ms in Figure 8(d), 90 ms in Figure 8(e), and 790 ms in Figure 8(f).

In terms of performance, our method is superior because no shadow ray marching is required. However, for a thorough comparison, we increase the step sizes of both other methods [1, 39] until the performance of shadow computation (for example when the transfer function changes) is similar to our technique, which requires a very large step size. Figure 8(g) shows the result for ambient scattering. Since the ambient sphere of each point is almost as large as the entire data set, directional shadows are missing almost completely. In Figure 8(h), the result for cone shadows is presented. The cone of each sample point in

---

[2]http://www.osirix-viewer.com/datasets/

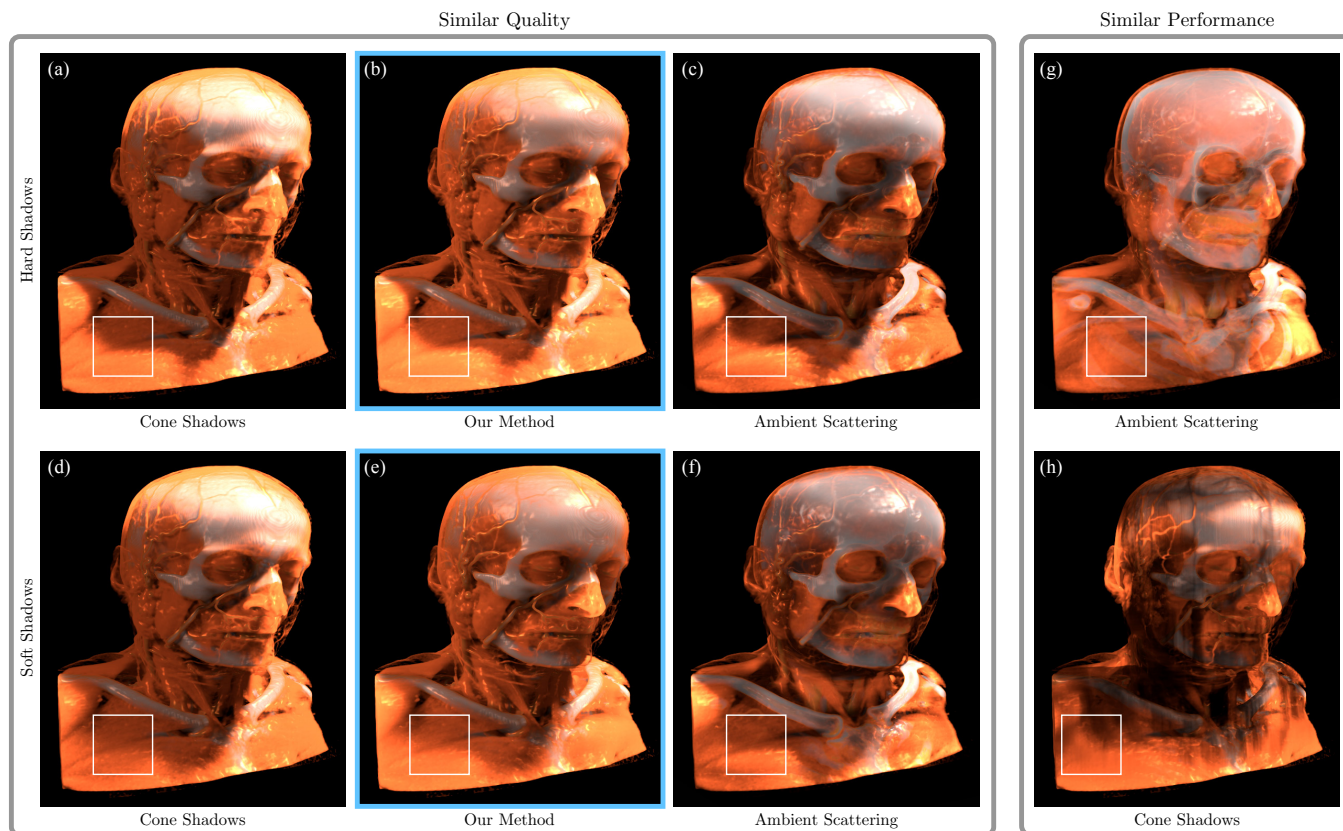Similar Quality   Similar Performance

Fig. 8. Visualization of the Manix data set using three orthodirectional light sources and three different methods. Similar quality: (a) Cone shadows [39] with a small cone angle for hard shadows. (b) Our method with a small filter size and comparable shadows. (c) Ambient scattering [1] with a small ambient radius. (d) Cone shadows with a large cone angle for softer shadows. (e) Our method with a larger filter size for similar shadows. (f) Ambient scattering with larger radius. Similar performance: (g) Ambient scattering with similar performance than our approach; directional shadows are largely missing. (h) Cone shadows with similar performance than our approach; shadows are inconsistent.

the volume is approximated with only one cuboid to achieve the same performance as our method. However, this leads to strongly inconsistent shadows because of the volume-aligned SAT and the direction of all shadows is inherently aligned with the main axes of the data set. Furthermore, the size of the cuboid varies strongly for different sample points, which leads to noticeable artifacts.

In Table 1, we summarize performance measurements and memory requirements of the SATs. In all cases, our approach with filtered shadows (FS) can reach a significant performance boost compared to single scattering (SS), cone shadows (CS), and ambient scattering (AS) when the transfer function changes. A minor disadvantage of our method is that the CSAT and SSAT need to be recomputed when the light sources move. In this case, cone shadows and ambient scattering only need to update the shadow cache; however, shadow ray marching is still much more expensive than our full recomputation, since the SATs can be computed very efficiently in contrast to ray marching. For example, with the Manix data set, recomputation of the CSAT requires only 26 ms, whereas ray marching is in the order of 1 s. In fact, our approach is almost as fast as ambient occlusion (AO) as long as one SAT is sufficient. Only if multiple SATs are required, like for the Vortex Cascade, AO remains notably faster than our method, but it does not provide directional shadows. Nonetheless, the transfer function or lighting setup are typically not changed for each frame by a user and a recomputation of the shadows is not necessary. In this case, the compared methods (SS, CS, AS, AO) still provide high rendering performance.

7 CONCLUSION

We have introduced an efficient technique for low-pass filtered volumetric shadows for interactive DVR to reduce disruptive hard shadow edges and noise-like visual patterns in the presence of fine structures due to single scattering illumination. Moreover, soft shadows are particularly beneficial for time-dependent data or when the data set is rotated relative to the light sources. The unique feature of our approach is that shadows of arbitrary softness can be computed without explicit shadow ray marching and at no extra cost.

With our current approach, a user needs to adjust the filter size interactively. However, in future work, it could be possible to automatically determine an ideal filter size, e.g., based on signal theory or with a perception-driven metric. In this way, the kernel size could be adapted to the transfer function or viewpoint, which would allow one to automatically filter disturbing patterns, but preserve as much details as possible by avoiding too strong blurring when no high-frequency shadows are present. Furthermore, a user study could evaluate and quantify the benefit of such an approach.

Our current implementation can be limited by memory constraints if more than six orthodirectional lights or several point lights are used. However, this is only a limitation of our implementation, which could be improved with an interleaved computation scheme so that only a single SAT is required in terms of memory consumption: as soon as one SAT is computed, the contribution of each light source could be added to the shadow cache and then the same SAT could be employed to compute the shadows of the next light source.

REFERENCES

[1] M. Ament, F. Sadlo, and D. Weiskopf. Ambient volume scattering. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2936–2945, 2013.

[2] U. Behrens and R. Ratering. Adding shadows to a texture-based volume renderer. In *Proc. of the IEEE Symposium on Volume Visualization*, pages 39–46, 1998.

[3] E. Cerezo, F. Pérez, X. Pueyo, F. J. Seron, and F. X. Sillion. A survey on participating media rendering techniques. *Vis. Comput.*, 21(5):303–328, 2005.

[4] S. Chandrasekhar. *Radiative Transfer*. Dover, 1960.

[5] C. Crassin, F. Neyret, M. Sainz, S. Green, and E. Eisemann. Interactive indirect illumination using voxel cone tracing. *Comput. Graph. Forum*, 30(7):1921–1930, 2011.

[6] F. C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Computer Graphics*, 18(3):207–212, 1984.

[7] J. Díaz, P.-P. Vázquez, I. Navazo, and F. Duguet. Real-time ambient occlusion and halos with summed area tables. *Computers & Graphics*, 34(4):337–350, 2010.

[8] E. Eisemann, U. Assarsson, M. Schwarz, and M. Wimmer. Casting shadows in real time. In *ACM SIGGRAPH Asia Courses*, 2009.

[9] T. Engelhardt and C. Dachsbacher. Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 119–125, 2010.

[10] R. Fernando. Percentage-closer soft shadows. In *ACM SIGGRAPH Sketches*, 2005.

[11] J. A. Ferwerda, P. Shirley, S. N. Pattanaik, and D. P. Greenberg. A model of visual masking for computer graphics. In *Proceedings of ACM SIGGRAPH*, pages 143–152, 1997.

[12] M. Hadwiger, A. Kratz, C. Sigg, and K. Bühler. GPU-accelerated deep shadow maps for direct volume rendering. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Graphics Hardware*, pages 49–52, 2006.

[13] J. Hensley, T. Scheuermann, G. Coombe, M. Singh, and A. Lastra. Fast summed-area table generation and its applications. *Comput. Graph. Forum*, 24(3):547–555, 2005.

[14] F. Hernell, P. Ljung, and A. Ynnerman. Interactive global light propagation in direct volume rendering using local piecewise integration. In *Proc. of the IEEE/EG International Symposium on Volume and Point-Based Graphics*, pages 105–112, 2008.

[15] F. Hernell, P. Ljung, and A. Ynnerman. Local ambient occlusion in direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 16(4):548–559, 2010.

[16] F. Hindenlang, G. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C.-D. Munz. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids*, 61:86–93, 2012.

[17] G. S. Hubona, P. N. Wheeler, G. W. Shirah, and M. Brandt. The relative contributions of stereo, lighting, and background scenes in promoting 3d depth visualization. *ACM Trans. Comput.-Hum. Interact.*, 6(3):214–242, 1999.

[18] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. of ACM SIGGRAPH*, pages 311–320, 1998.

[19] D. Jönsson, J. Kronander, T. Ropinski, and A. Ynnerman. Historygrams: Enabling interactive global illumination in direct volume rendering using photon mapping. *IEEE Trans. Vis. Comput. Graph.*, 18:2364–2371, 2012.

[20] D. Jönsson, E. Sundén, A. Ynnerman, and T. Ropinski. Interactive volume rendering with volumetric illumination. In *Eurographics STAR program*, pages 53–74, 2012.

[21] J. Kniss, S. Premože, C. Hansen, and D. Ebert. Interactive translucent volume rendering and procedural modeling. In *Proc. of IEEE Visualization*, pages 109–116, 2002.

[22] T. Kroes, F. H. Post, and C. P. Botha. Exposure render: An interactive photo-realistic volume rendering framework. *PLoS ONE*, 7(7):e38586, 2012.

[23] J. Kronander, D. Jönsson, J. Low, P. Ljung, A. Ynnerman, and J. Unger. Efficient visibility encoding for dynamic illumination in direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 18(3):447–462, 2012.

[24] M. Langer and H. Bülthoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29(6):649–660, 2000.

[25] A. Lauritzen. Chapter 8: Summed-area variance shadow maps. In H. Nguyuen, editor, *GPU Gems 3*, pages 157–181. Addison-Wesley Professional, 2007.

[26] M. Levoy. Volume rendering using the Fourier projection-slice theorem. In *Proc. of the Conference on Graphics Interface*, pages 61–69, 1992.

[27] F. Lindemann and T. Ropinski. About the influence of illumination models on image comprehension in direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 17(12):1922–1931, 2011.

[28] T. Lokovic and E. Veach. Deep shadow maps. In *Proc. of ACM SIGGRAPH*, pages 385–392, 2000.

[29] N. Max. Optical models for direct volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 1(2):99–108, 1995.

[30] B. Moloney, M. Ament, D. Weiskopf, and T. Möller. Sort first parallel volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 17(8):1164–1177, 2011.

[31] A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, 1998.

[32] D. Patel, V. Šoltészová, J. M. Nordbotten, and S. Bruckner. Instant convolution shadows for volumetric detail mapping. *ACM Trans. Graph.*, 32(5):154:1–154:18, 2013.

[33] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[34] C. Rezk-Salama. GPU-based Monte-Carlo volume raycasting. In *Proc. of the Pacific Conference on Computer Graphics and Applications*, pages 411–414, 2007.

[35] T. Ropinski, C. Döring, and C. Rezk-Salama. Interactive volumetric lighting simulating scattering and shadowing. In *Proc. of IEEE Pacific Visualization Symposium*, pages 169–176, 2010.

[36] T. Ropinski, J. Meyer-Spradow, S. Diepenbrock, J. Mensmann, and K. Hinrichs. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Comput. Graph. Forum*, 27(2):567–576, 2008.

[37] M. Ruiz, I. Boada, I. Viola, S. Bruckner, M. Feixas, and M. Sbert. Obscurance-based volume rendering framework. In *Proc. of the IEEE/EG International Symposium on Volume and Point-Based Graphics*, pages 113–120, 2008.

[38] M. Salvi, K. Vidimce, A. Lauritzen, and A. E. Lefohn. Adaptive volumetric shadow maps. *Comput. Graph. Forum*, 29(4):1289–1296, 2010.

[39] P. Schlegel, M. Makhinya, and R. Pajarola. Extinction-based shading and illumination in GPU volume ray-casting. *IEEE Trans. Vis. Comput. Graph.*, 17(12):1795–1802, 2011.

[40] M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, and K. Bouatouch. A directional occlusion shading model for interactive direct volume rendering. *Comput. Graph. Forum*, 28(3):855–862, 2009.

[41] A. J. Stewart. Vicinity shading for enhanced perception of volumetric data. In *Proc. of IEEE Visualization*, pages 355–362, 2003.

[42] E. Sundén, A. Ynnerman, and T. Ropinski. Image plane sweep volume illumination. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2125–2134, 2011.

[43] V. Šoltészová, D. Patel, S. Bruckner, and I. Viola. A multidirectional occlusion shading model for direct volume rendering. *Comput. Graph. Forum*, 29(3):883–891, 2010.

[44] L. Wanger, J. Ferwerda, and D. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Comput. Graph. Appl.*, 12(3):44–58, 1992.

[45] C. Weber, A. Kaplanyan, M. Stamminger, and C. Dachsbacher. Interactive direct volume rendering with many-light methods and transmittance caching. In *Proc. of the International Workshop on Vision, Modeling, and Visualization*, pages 195–202, 2013.

[46] C. Wyman, S. Parker, P. Shirley, and C. Hansen. Interactive display of isosurfaces with global illumination. *IEEE Trans. Vis. Comput. Graph.*, 12(2):186–196, 2006.

[47] C. Zhang and R. Crawfis. Shadows and soft shadows with participating media using splatting. *IEEE Trans. Vis. Comput. Graph.*, 9(2):139–149, 2003.

[48] Y. Zhang and K.-L. Ma. Fast global illumination for interactive volume visualization. In *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 55–62, 2013.

[49] Y. Zhang and K.-L. Ma. Lighting design for globally illuminated volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2946–2955, 2013.