

Advanced Lighting for Unstructured-Grid Data Visualization

Category: Research

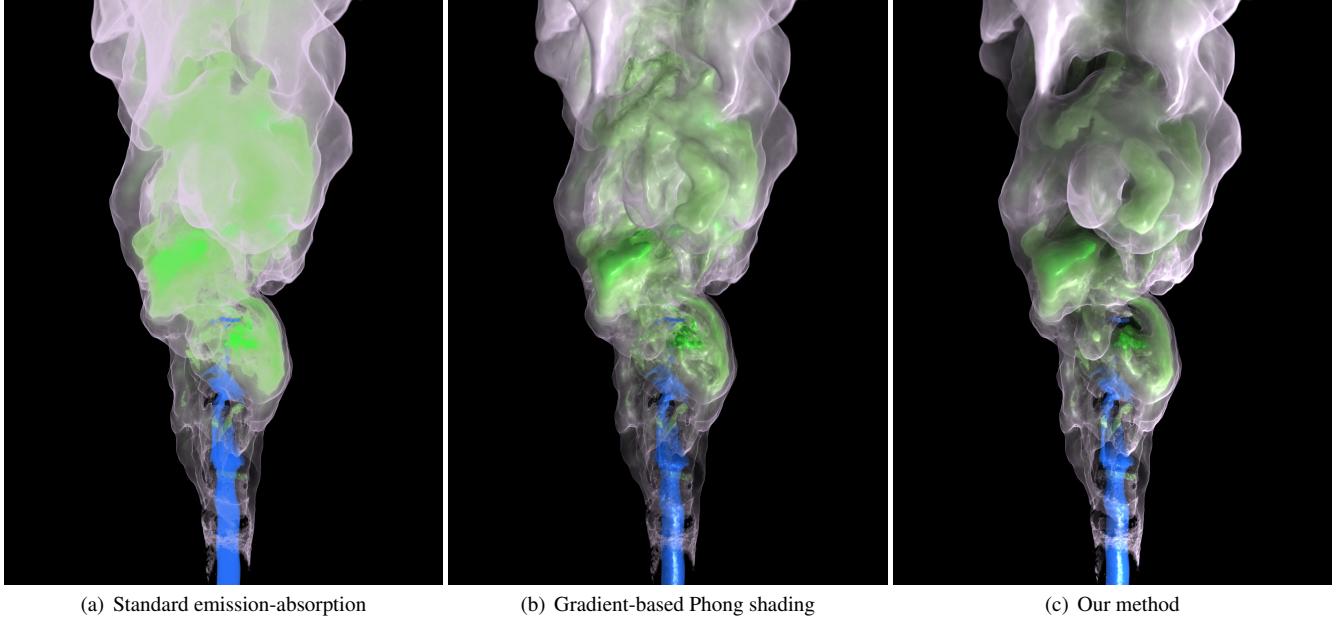


Figure 1: Unstructured grid volume visualization examples showing our method's advantages. The H_2O field of the Flame dataset is visualized. With our method, shadows enhance the visual perception of depth cue greatly.

ABSTRACT

The benefits of using advanced illumination models in volume visualization have been demonstrated by many researchers. Interactive volume rendering incorporated with advanced lighting has been achieved with GPU acceleration for regular-grid volume data, making volume visualization even more appealing as a tool for 3D data exploration. This paper presents an interactive illumination strategy, which is specially designed and optimized for volume visualization of unstructured-grid data. The basis of the design is a partial differential equation based illumination model to simulate the light propagation, absorption, and scattering within the volumetric medium. In particular, a two-level scheme is introduced to overcome the challenges presented by unstructured grids. Test results show that the added illumination effects such as global shadowing and multiple scattering not only lead to more visually pleasing visualization, but also greatly enhance the perception of the depth information and complex spatial relationships for features of interest in the volume data. This volume visualization enhancement is introduced at a time when unstructured grids are becoming increasingly popular for a variety of scientific simulation applications.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.3.8 [Computer Graphics]: Applications

1 INTRODUCTION

Direct volume rendering (DVR) is widely used to examine 3D field data in the area of scientific visualization. For effective visualization, an important aspect of volume rendering is the ability to provide visual cues to aid users in comprehending the spatial structures in the datasets. Although simple Phong shading is

fast, it is mainly computed with an approximation of local surface orientations, which gives limited realism and may introduce unwanted surface details. On the other hand, advanced illumination models that provide global lighting effects such as global shadows and multiple scattering can greatly enhance depth cues and better present the local thickness of layer features in the data, leading to better imagery results for depicting complex volumetric features. Another key aspect of volume rendering is interactivity. Thanks to the enormous processing power provided by modern graphics hardware, volume rendering can be performed in real-time today. Several techniques to apply visually plausible global illumination effects to volume rendering at interactive speed have been introduced [1, 12, 14, 15, 19, 23, 28].

However, most of interactive illumination techniques utilize the graphics hardware's ability to accelerate computing on regular rectilinear grids of volume elements. Volume rendering with high quality illumination for other data geometries such as warped grids or unstructured grids that are widely used in scientific simulation is still a challenge. The main difficulties with unstructured grids are

- the non-uniform distribution of the vertices,
- the high cost of sampling arbitrary points in the volume domain, and
- the non-convexity of the volume grid.

Unlike regular grids, the vertices in an unstructured grid are distributed unevenly, often clustered in a small area of the volume domain in order to provide high resolution at interesting regions of the dataset. Therefore, most of texture-based techniques are not appropriate because they cannot provide the required resolution adapt-

ability and may lose accuracy for finer cells in the grid. Also, the acceleration techniques relying on the regularity of the data such as summed area tables (SAT) are not capable of handling unstructured grids. Sampling arbitrary points in an unstructured grid is an expensive operation because it involves a search process within a spatial data structure to locate the point. The non-convexity of the volume grid may cause problems when simulating light transport. The cases where light goes out from the volume and then goes into the volume again must be carefully considered or else the results would be incorrect.

In this paper, we present a method for interactive volume rendering of unstructured-grid data with high quality illumination including shadowing and multiple scattering effects. Addressing the aforementioned challenges presented by unstructured grids, our contributions are mainly as follows: (i) a two-level approach which not only solves the non-convexity problem but also accelerates the computation, (ii) the derivation of the numerical scheme to solve the model equation on unstructured grids. In particular, in order to achieve numerical stability and high efficiency, a semi-Lagrangian scheme is used. Our method can be easily generalized to handle other types of grids including curvilinear, AMR, and even hybrid grids. Furthermore, since the method does not depend on heavy preprocessing, it is even suitable for time-varying data.

2 RELATED WORK

The volume illumination involves solving the radiative transfer equation [3], which formalizes the evolution of light transport with participating media, including light absorption, scattering and emission effects. Max [18] presented an overview on the optical models used in direct volume rendering (DVR). Kajiya and von Herzen [13] presented a model for rendering volume densities which included multiple scattering computation. The scattering equation is solvable analytically only in some simple configurations. Several methods have been proposed to numerically solve the radiative transfer equation by using either ray tracing [16, 17] or radiosity [22]. These methods are capable of rendering various illumination effects with arbitrary phase functions and heterogeneous media. However, the computational cost is tremendous, and the speed is far from what is required for interactivity.

To simulate light transport in participating media, Jensen and Christensen proposed a two-pass method using volumetric photon mapping (VPM) [11]. In its first stage, the method traces photons throughout the volume and stores them into spatial hierarchical data structure. In the second pass, an adaptive ray marching algorithm is used and the pre-stored photons are gathered for final radiance. There are many methods aiming to improve the efficiency of VPM [2, 9]. However, the cost of creation and maintenance of the hierarchical structure prevents early VPM methods from achieving interactive performance. More recently, Jönsson et al. [12] presented an interactive method extending photon mapping. By tracking the history of photons, full recomputation of the photon map can be avoided when the transfer function is changed; instead, only the photons affected by the parameter changes are recomputed. However, full recomputation is still required when the parameter change affects all photons such as moving light source or when time-varying data is visualized. However, in the context of unstructured-grid DVR, photon mapping based methods may suffer from the adaptive nature of the unstructured grids. Large amount of photons may be needed in order to fulfill the accuracy requirement in the high resolution regions of the grid.

Many interactive techniques have been proposed to enhance volume rendering for visualization by supporting illumination effects such as shadowing and scattering. For example, obscurance and ambient occlusion (AO) methods provide relatively inexpensive ways to approximate indirect illumination. GPU-based screen space ambient occlusion (SSAO) [25] is a high-performance tech-

nique to render occlusion effects in polygonal shading by sampling the depth buffer in screen space. It can also be applied to volume rendering [8]. Although efficient, SSAO may not produce desired results in complex scenarios, for example, semitransparent objects, since only one depth value per pixel can be used. Object space AO, as shown in the paper of Ruiz et al. [21], provides better quality at the price of higher memory and computation cost than SSAO. The evaluation of occlusion/extinction can be accelerated through the use of SAT [8, 23], but the SAT based techniques cannot be used for unstructured grids. Recently, Schott et al. [24] demonstrated an ambient occlusion technique for interactive volume rendering that can render mutual occlusion effects of both volume and geometric primitives, which is also supported by our method.

Although multiple scattering is important for realistic rendering of participating media, Max has stated that it is overkill to compute it in most scientific visualization applications [18]. However, it has been shown that physically plausible approximation to the global illumination effects can improve perceptual quality of the visualization [19]. Kniss et al. [14, 15] proposed an interactive shading model that incorporated volumetric shadows and scattering effects, where scattering is restricted to forward scattering and estimated by applying blurring operations within a cone toward the light source. Using half-angle slicing, the lighting can be evaluated efficiently in the image space, but the light configuration is limited to a single light source. Extending the idea of Kniss et al., Ropinski et al. [19] applied the lighting computation directly in volume space. In their approach, the light volume is computed first, where the light propagation is evaluated slice by slice, and then used in the shading stage of rendering. Thus, the method is applicable to any volume rendering technique, including GPU-based ray casting. However, in the context of unstructured-grid DVR, the uniform slices may not be able to provide sufficient resolution in some regions of the volume since the mesh can be highly adaptive. Following the cone-sampling idea presented by Kniss et al., Schlegel et al. [23] approximated the soft shadows and scattering effects by aggregating the extinction values within the shadow cone. In their approach, the cone is approximated by a series of cuboids, and the extinction computation of the cuboids is accelerated by employing SAT. Ament et al. [1] presented a pre-integration method for scattering in DVR, which also utilizes SAT for fast extinction values summation. The pre-integration table, independent of dataset and transfer function, is obtained by solving the full light transport with a Monte-Carlo simulation within a set of spherical regions. One of the table parameters, the ambient extinction coefficient, can be efficiently calculated by SAT to enable interactive rendering speeds. However, SAT does not work for unstructured data, and therefore the evaluation of the extinction values can be expensive in unstructured-grid DVR. Zhang and Ma [28] proposed a grid-based volume illumination method using a partial differential equation (PDE) based illumination model, but their numerical scheme is designed for regular grids only.

Most of the interactive volumetric illumination approaches rely on the regularity of the data. To our knowledge, very few papers address interactive global illumination for unstructured grid volume rendering. Although gradient-based local Phong lighting has been used to help understand spatial structures in unstructured-grid data [6], the model is not sufficient to resolve the spatial ambiguities within complex volumetric features.

3 ILLUMINATION MODEL

We use a PDE-based illumination model [28] in our interactive unstructured-grid volume rendering approach. The model uses a single value to encode light energy at each spatial location for each light source. The light propagation, absorption and scattering are

modeled as a convection-diffusion equation

$$\frac{\partial}{\partial t} \rho(\mathbf{x}) = -c\mathbf{u}(\mathbf{x}) \cdot \nabla \rho(\mathbf{x}) - \sigma_a(\mathbf{x})\rho(\mathbf{x}) + \sigma_s \nabla^2 \rho(\mathbf{x}) \quad (1)$$

where $\rho(\mathbf{x})$ is the energy density at \mathbf{x} , c is the speed of light, $\mathbf{u}(\mathbf{x})$ is the unit light propagation direction at \mathbf{x} , $0 \leq \sigma_a(\mathbf{x}) \leq 1$ is the absorption coefficient at \mathbf{x} , and σ_s is the isotropic scattering coefficient. The solution of $\rho(\mathbf{x})$ of each light source is combined into a single light color field by

$$L(\mathbf{x}) = \sum_{k=1}^N \rho_k(\mathbf{x}) L_k \quad (2)$$

where L_k is the color of the k th light source. The light color field value L is used to shade the sample points in the rendering process. In order to exhibit the surface shape, the light value can be used with gradient-based Phong shading

$$L_{final}(\mathbf{x}) = L(\mathbf{x})(I_a(\mathbf{x}) + I_d(\mathbf{x}) + I_s(\mathbf{x})) \quad (3)$$

where I_a , I_d , and I_s are the ambient term, diffuse term, and specular term of Phong lighting, respectively.

To reduce computational cost, Eq. 1 can be split into a convection equation

$$\frac{\partial}{\partial t} \rho(\mathbf{x}) = -c\mathbf{u}(\mathbf{x}) \cdot \nabla \rho(\mathbf{x}) - \sigma_a(\mathbf{x})\rho(\mathbf{x}), \quad (4)$$

which models light propagation as well as absorption and needs to be evaluated for each light source, and a diffusion equation

$$\frac{\partial}{\partial t} \rho(\mathbf{x}) = \sigma_s \nabla^2 \rho(\mathbf{x}) \quad (5)$$

which approximates the multiple scattering effects and is independent of light sources and needs to be solved only once for multiple light sources.

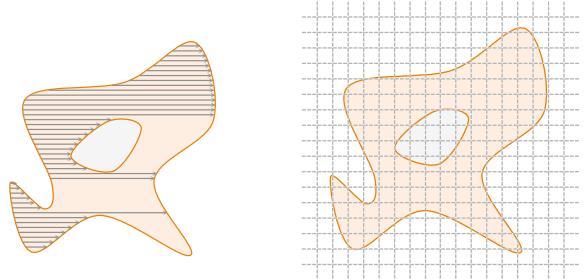
4 LIGHT VOLUME CALCULATION

In this section we show how we solve Eq. 1 for unstructured grids.

A straightforward approach to apply the illumination method described in the previous section to an unstructured grid is to resample the data onto a regular grid for light simulation. However, since an unstructured grid can be highly adaptive, the vertices of the grid may be distributed unevenly and clustered in a small area of the volume domain. Thus, the light field evaluated according to the resampled volume may not be able to provide sufficient accuracy and may fail to capture high frequency shadows for the regions with originally finer cells. To best fit the accuracy requirement, the volume domain should be discretized according to the original unstructured grid for light simulation.

However, simulating light directly on unstructured grids is challenging. A significant issue is the handling of non-convex meshes. When ray casting non-convex meshes, the concaves and holes in the mesh may cause re-entry problems. The light field evaluation process may suffer from a similar problem; if the calculation process does not specially take care of the concaves and holes of the mesh, the light propagation may be evaluated incorrectly as shown in Figure 2 (a). If we write extra codes to explicitly check these cases in the calculation process, the performance may be greatly impacted. Another issue is that the operations on unstructured grids are generally costlier than that on regular grids. For example, the interpolation operation on unstructured grids is much more expensive than that on regular grids since it cannot be done by hardware.

We solve the problem by introducing a two-level approach, where the first level is a resampled regular Cartesian grid (henceforth referred to as the *level-0* grid), and the second level is the original unstructured grid (henceforth referred to as the *level-1* grid).



(a) Unstructured grid with concaves and a hole

(b) Original grid (*level-1*) with *level-0* regular grid

Figure 2: (a) Assuming that the space in the concaves and holes is transparent, the light should be propagated across the concaves and holes. If this situation does not handle properly, the simulation result will be incorrect. (b) We add the *level-0* regular grid to solve the problem.

To calculate the light field values, we first simulate the light propagation on the *level-0* grid, so that the light can propagate across concaves and holes. Then the solution on the *level-0* grid is interpolated onto the *level-1* grid as Dirichlet boundary values and initial interior values. Then we evaluate the light field values at the vertices of the *level-1* grid. The *level-0* grid not only solves the non-convexity problem, but also serves as an acceleration structure for *level-1* grid. Since the solution of *level-0* grid already provides a rough light simulation result for the initialization of *level-1* calculation, only a few iterations are required to achieve convergence on the *level-1* grid. The *level-1* grid, on the other hand, captures the detailed lighting effects such as high frequency shadows that are absent in the *level-0* result. Combining the advantages of regular grid and unstructured grid, the two-level approach provides high efficiency as well as high accuracy.

Next we give the detailed process of our method. As a preprocessing step, the data values on the original unstructured grid are resampled onto the *level-0* regular grid. The *level-0* grid covers the whole volume domain. A mask volume is also created to indicate whether a grid cell is inside the volume domain, is a blank cell, or is inside a geometric occluder in the case that there are geometric primitives embedded in the volume. Usually we do not need a very high resolution *level-0* grid since the solution of *level-0* grid only serves as the initial values of the *level-1* grid in the numerical solving process.

To simulate light propagation on the *level-0* grid, we use the first-order upwind scheme [7] to numerically solve Eq. 4. The boundary conditions are set according to the light settings for each light source. The resulting ρ values are interpolated onto the *level-1* grid as boundary conditions for boundary vertices and the initial values for inner vertices, respectively. Although high-order interpolation method can improve the quality, we use typical tetrahedral barycentric interpolation due to efficiency. The convection equation is then solved on the *level-1* grid using a semi-Lagrangian scheme [26]. The solutions for all light sources are combined into a single light color field, and then the Gauss-Seidel iteration method [10] is used to solve the diffusion step for simulating scattering effects. To shade the point samples in the rendering process, only the light color field solution on the *level-1* grid is necessary. The light color values are interpolated on the *level-1* grid and multiplied with the light value obtained from the standard emission-absorption model or the Phong model to shade the point samples. The *level-0* light field can be used as a 3D texture for shading geometric primitives, which is an optional feature which enables applying the occlusion effects from the volume to the geometric primitives.

4.1 Light Propagation

Our first trial to solve Eq. 4 on the *level-1* grid is to apply the upwind scheme to the unstructured *level-1* grid, but we found the method suffers from the stability problem. With this method, the time step Δt must be set to a value small enough relatively to the discretization grid size (the CFL condition) in order to achieve stable results. However, in an unstructured grid, the distance between two vertices can be very small since the interesting area is likely to be represented by denser cells. To achieve numerical stability, the Δt must be chosen according to the size of the finest cell of the mesh, which is not practical because using a small time step means a large number of steps are needed to achieve convergence.

In order to avoid the stability problem, we use a semi-Lagrangian scheme [26], which allows a larger time step while maintaining numerical stability, to solve Eq. 4 for the unstructured *level-1* grid. Since the light speed and direction are constant along the particle trajectory, for the propagation part of Eq. 4, we have the solution for the energy density at time t_{n+1} :

$$\rho_{n+1}(\mathbf{x}) = \rho_n(\mathbf{x}_d) \quad (6)$$

where the energy density at the departure point

$$\rho_n(\mathbf{x}_d) = \rho_n(\mathbf{x} - \mathbf{c}\mathbf{u}\Delta t) \quad (7)$$

can be evaluated by interpolation. We then take absorption into account:

$$\rho_{n+1}(\mathbf{x}) = \rho_n(\mathbf{x}_d)(1 - \alpha)^{\tilde{\sigma}_a} \quad (8)$$

where α is the opacity value which integrates the extinction along the segment between \mathbf{x} and \mathbf{x}_d . Transfer function pre-integration [20] can be used to evaluate α efficiently and accurately. $\tilde{\sigma}_a$ is a user-defined value which controls the overall absorption strength.

Typical tetrahedral barycentric interpolation method is used to evaluate the $\rho(\mathbf{x}_d)$ value in Eq. 7, or a high-order interpolation method can be used when efficiency is not critical. Since some cells can be very small, the step between \mathbf{x} and \mathbf{x}_d may cross multiple cells, and therefore a cell traversal procedure similar to ray casting is required. The time step Δt should be small enough so that important features are not likely to be bypassed.

Usually the calculation on *level-1* grid only needs to cover the light propagation within the range between two adjacent sample points of the *level-0* grid in order to capture detailed lighting effects. We set the iteration number according to the ratio of the cell size of the *level-0* grid to the average edge length of the *level-1* grid. The calculation of Eq. 8 for each vertex can be fully parallelized, so it is suitable for a GPU implementation.

4.2 Scattering

Before solving Eq. 5 for approximating scattering effects, the solutions of Eq. 4 for all light sources on the *level-1* grid are combined into a single light color field simply by

$$L(\mathbf{x}) = \sum_{k=1}^N \rho_k(\mathbf{x}) L_k \quad (9)$$

where ρ_k is the solution of Eq. 4 for the k th light source, and L_k is the color of the k th light source, so that we only need to solve a single diffusion equation (Eq. 5) even for multiple light sources. Therefore, Eq. 5 is modified to a vector form

$$\frac{\partial}{\partial t} L(\mathbf{x}) = \hat{\sigma}_s \nabla^2 L(\mathbf{x}) \quad (10)$$

where $\hat{\sigma}_s$ consists of three scattering coefficients for different color channels. We approximate the Laplacian of light color field at vertex v on the *level-1* grid by

$$\nabla^2 L(\mathbf{x}_v) \approx \sum_{i \in N(v)} w_i (L(\mathbf{x}_i) - L(\mathbf{x}_v)) \quad (11)$$

where $N(v)$ is the neighborhood of v , and w_i is the weighting factor of neighbor i , which is inversely proportional to the square of the distance between vertices i and v , so that a closer neighbor has higher weight.

Thus, using the Gauss-Seidel iteration method [10] to solve Eq. 10, in each iteration we apply

$$L(\mathbf{x}_v) \leftarrow \frac{L(\mathbf{x}_v) + \Delta t \hat{\sigma}_s \sum_{i \in N(v)} w_i L(\mathbf{x}_i)}{1 + \Delta t \hat{\sigma}_s \sum_{i \in N(v)} w_i} \quad (12)$$

to each vertex. We let $\tilde{\sigma}_s = \Delta t \hat{\sigma}_s$ be a user-defined value so that the user can determine the scattering strength.

5 IMPLEMENTATION

Our rendering system enables the user to edit the transfer function and light settings interactively. Once the transfer function or the light settings are changed, the light volume is updated. When changing the transfer function, the pre-integrated transfer function table also needs to be updated. We compute the pre-integration table fully on the GPU. Since the light propagation for each light source is evaluated independently of other light sources, if only some of the light sources are changed, the results of propagation simulations of the unchanged light sources do not need to be updated. For each changed light source, the light propagation is evaluated on the *level-0* and *level-1* grids as described in the previous section. Then the light volumes for the light sources are combined into a single light color volume, and the scattering simulation is performed on the light color volume. In the rendering stage, the light color value is interpolated as an attribute of the vertices, and the Phong shading result is multiplied by the light color value to shade the material. Algorithm. 1 summarizes the procedure.

Algorithm 1 Unstructured-grid volume illumination

```

1: for each frame do
2:   if transfer function is changed then
3:     update pre-integration table;
4:     mark all lights as changed;
5:   end if
6:   for each light source do
7:     if the light is changed then
8:       initialize the level-0 light volume;
9:       simulate light propagation and absorption on level-0;
10:      initialize the level-1 light volume using the solution of
11:        level-0;
12:      simulate light propagation and absorption on level-1;
13:    end if
14:   end for
15:   if any light changed then
16:     combine the level-1 light volumes;
17:     simulate light scattering on level-1 light color volume;
18:   end if
19:   render the frame using ray casting;
end for

```

We use NVIDIA CUDA to implement the light simulation, and volume rendering is done by combined CUDA and OpenGL. The pre-integrated transfer function table, grid data, and the light volume are all kept in the GPU memory. The per vertex calculation including propagation and absorption simulation, light volume combination, and scattering simulation are all computed in parallel by GPU kernels.

Our lighting method can be easily integrated into any unstructured-grid rendering algorithm since the light volume is computed independent of the rendering process. We use the ray casting method for volume rendering in our system. A tetrahedral

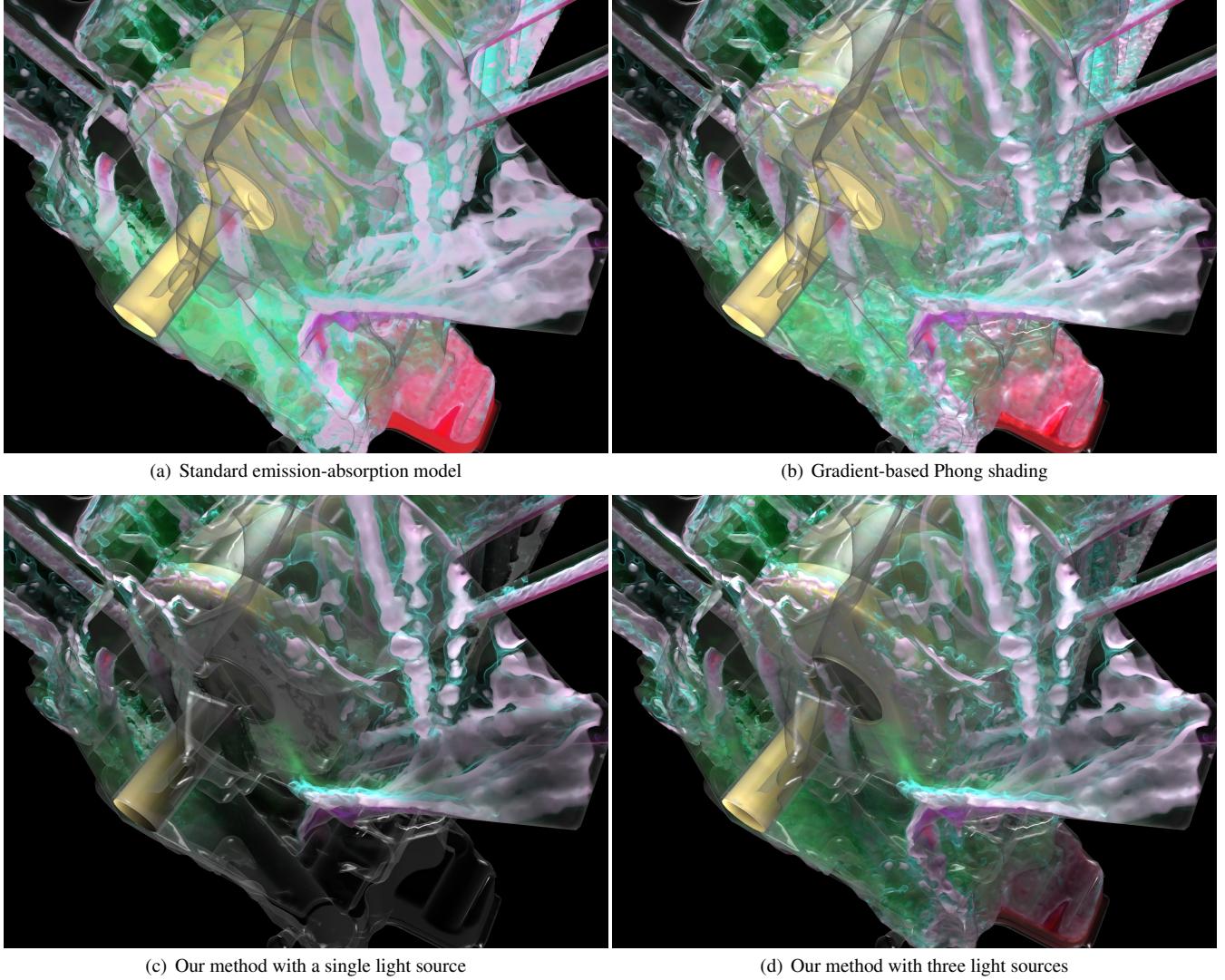


Figure 3: Volume rendering of the Engine dataset with different optical models. With our lighting method, the structure of the flow around the crankshaft is much clearer, and the spatial ambiguity around the pink features is well resolved. Advanced lighting also helps better illustrate the shape and structure of the engine body.

ray caster similar to the one presented by Weiler et al. [27] is implemented. However, [27] and our implementation are different in that they use convexification to solve re-entry problem while we use depth peeling to deal with both re-entry and embedded geometry. The grids containing variant types of cells (pyramids, prisms, and hexahedra) are divided into tetrahedra first and rendered by ray casting as ordinary tetrahedral meshes.

6 RESULTS AND DISCUSSION

Figures 1 and 3 show the volume rendering results with different optical models. In Figure 1, the H_2O field of the Flame dataset is visualized, while Figure 3 shows the visualization of the Engine dataset. The standard emission-absorption model (Fig. 1(a) and 3(a)) provides very little information about the surface shape and depth cue. With the Lambertian diffuse and specular reflectance, the gradient-based Phong shading better reveals the surface shape (Fig. 1(b) and 3(b)). But the Phong shading model is still insufficient to resolve some spatial ambiguities. Combined with the gradient-based Phong shading, our advanced lighting method can provide not only the surface shape information, but also the visual

cues of the spatial structures within the data (Fig. 1(c) and 3(c)). However, in Fig. 3(c), some regions of the volume become too dark due to the shadow of the crankshaft. Our method supports multiple light sources, so that we can use some secondary light sources in addition to the primary light source to brighten the regions that are too dark. In Fig. 3(d), three light sources are used. In addition to the primary light source on the top, two secondary light sources are placed at the two sides to illuminate the over-occluded regions.

To evaluate the usefulness of our two-level approach, we applied the light simulation of the *level-0* grid only, and compared the result with the full *level-0 + level-1* simulation as shown in Figure 4, where the velocity magnitude of the highly adaptive Aircraft1 dataset is visualized. Although the resulting images are roughly the same, the *level-0*-only simulation cannot accurately capture the lighting effects in the fine regions of the dataset. Increasing the resolution of the *level-0* grid may solve the problem, but in order to resolve the finest regions of a highly adaptive unstructured grid, a very high resolution of *level-0* grid may be required, and therefore the computational cost and memory consumption may be not reasonable. The approaches that rely on the regularity of the grid (such

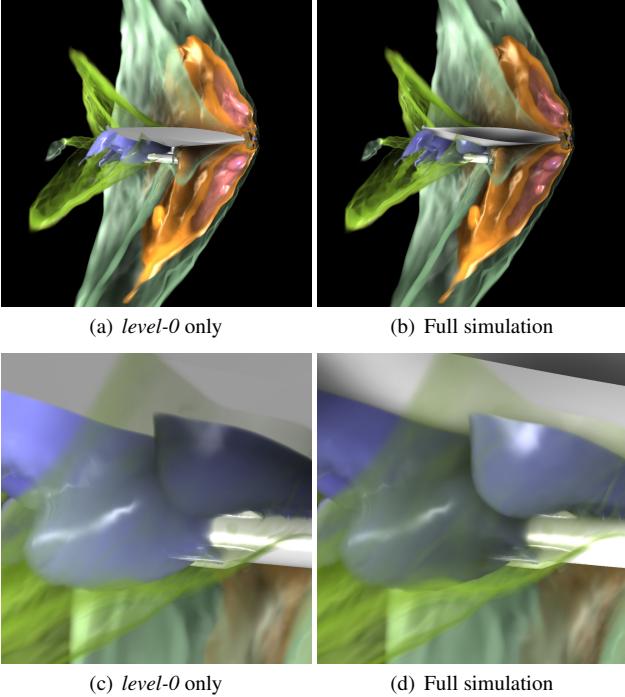


Figure 4: Volume rendering of the Aircraft1 dataset. (c) and (d) are the close-up view of (a) and (b), respectively. The images show that simulating the light only on the *level-0* grid cannot accurately capture the lighting effects at the fine regions of a highly adaptive grid.

as [1, 8, 19, 23, 28]) may suffer from the same problem when they are applied to unstructured data.

In our approach, we let user select the values of $\bar{\sigma}_a$ and $\bar{\sigma}_s$. $\bar{\sigma}_a$ determines the overall absorption level, while $\bar{\sigma}_s$ determines the strength of scattering within the media. We rendered images with different $\bar{\sigma}_a$ and $\bar{\sigma}_s$ values to show how these parameters affect the visual appearance of the material. In Figures 5(a) and (b), the rendering results with different $\bar{\sigma}_a$ values are compared. Obviously, with higher $\bar{\sigma}_a$ value, the darker the occluded regions become. The user can adjust this value to control the overall brightness of the image. Figure 5(c) shows that we can use high $\bar{\sigma}_a$ value with high light intensity to create an image with high contrast. The images with different $\bar{\sigma}_s$ values are compared in Figure 6 ($\bar{\sigma}_s = 0.1$ in (a) and $\bar{\sigma}_s = 10$ in (b)). Two directional light sources are used: a blue light on the left side and an orange back light. With stronger scattering, Fig. 6(b) is slightly brighter, and the different light colors are blended more, especially at the boundary where the light from the back light can scatter through the material. Also, the shadows in Fig. 6(b) are softer; we can adjust the $\bar{\sigma}_s$ value to control the softness of the shadows.

Figure 8 shows more rendering examples using our method.

6.1 Performance

We tested our method on an NVIDIA GeForce GTX Titan graphics card with 6 GB of video memory, installed on a PC with an Intel Core i7 3.5 GHz processor and 32 GB of main memory. Table 1 shows the characteristics of the datasets as well as the performance measurements; the timing numbers of the propagation calculation and the scattering calculation for the *level-0* and *level-1* grids are listed respectively. The total processing times, the summation of the above numbers, are also given in the table. In all test cases, a single directional light source is used. When multiple light sources are used, the propagation calculation is performed for each light

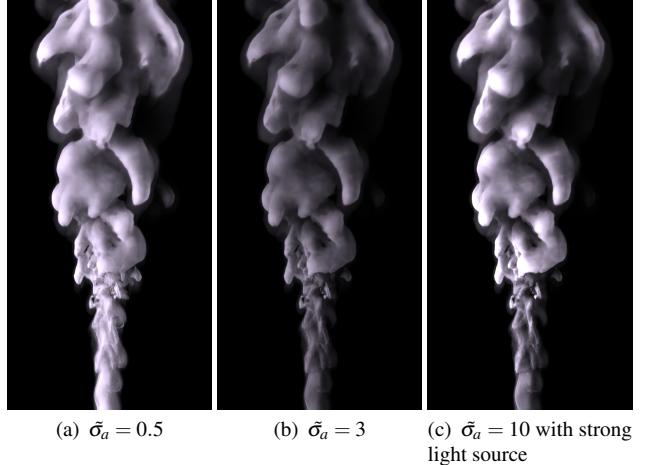


Figure 5: Rendering results with different absorption strength $\bar{\sigma}_a$. As $\bar{\sigma}_a$ increases, the shadows become darker.

source, while the scattering calculation needs to be done only once. The overall processing takes several hundred milliseconds in our test cases, which is fast enough for interactivity. With our implementation, the light volume calculation time is usually even shorter than the rendering time (using ray casting) when the image size is over 1024×1024 . Also notice that the light simulation is view-independent and is not necessarily computed for every frame. Instead, only when the transfer function or the light setting is changed, the light volume is needed to be updated. And since our propagation and scattering computation is iterative, the computation burden can be separated to multiple frames in order to improve the interactivity of the rendering system.

The resolution of the *level-0* grid used in Table 1 is $128 \times 128 \times 128$, which is sufficient to generate acceptable results in our test cases. The computational cost for the *level-0* grid is dependent upon the resampling resolution. We tested different resolutions and the timing results are shown in Table 2.

For the *level-1* grid, the light propagation and scattering calculation are applied at the vertices of the grid. Therefore, the computational cost should be linearly proportional to the grid size. For scattering, the computation takes place between a vertex and each adjacent vertex of it, so the degree of the vertices also effects the cost. However, the degree of vertices does not increase with the data size and usually can be seen as constant. In order to verify our time complexity estimation, we applied the method to a set of synthesized grids. We resampled the Vortex regular-grid dataset onto tetrahedral grids of different sizes (from 500K to 30M). The calculation times of propagation and scattering on *level-1* grids are shown in Figure 7. It can be clearly seen that both propagation and scattering times are about linearly proportional to the grid size. And the total calculation time for the largest grid we tested is still under one second.

In the rendering stage, for each sample point, an additional few memory fetches are needed to sample the light volume in order to apply our illumination technique. We measured the rendering time for a single frame with and without enabling our illumination method, respectively, to evaluate the impact of the technique on the rendering performance. The experimental results show that the impact is insignificant; When enabling the advanced illumination effects, the rendering time only increases slightly (under 10% in all tested datasets) compared to the rendering time without the effects.

In order to achieve better speed up, we can trade visual quality for performance. On the *level-0* grid, downsampled light volumes for light simulation can be used since that usually a light volume

Table 1: Performance measurements for the datasets in our study. The light volume calculation times are measured using a single directional light source. The resolution of the *level-0* grid is $128 \times 128 \times 128$.

Dataset	#Vertices	#Tetrahera	<i>level-0</i> Propagation	<i>level-0</i> Scattering	<i>level-1</i> Propagation	<i>level-1</i> Scattering	Total
Aircraft1	103 K	567 K	73 ms	12 ms	6 ms	20 ms	111 ms
Aircraft2	165 K	935 K	58 ms	11 ms	6 ms	24 ms	99 ms
Aircraft3	804 K	4.6 M	53 ms	10 ms	25 ms	136 ms	224 ms
Flame	538 K	3.2 M	57 ms	10 ms	15 ms	88 ms	170 ms
Engine	1.2 M	6.0 M	64 ms	11 ms	28 ms	152 ms	255 ms
Train	1.0 M	6.1 M	60 ms	11 ms	19 ms	123 ms	213 ms
Car	4.2 M	24.7 M	56 ms	11 ms	126 ms	561 ms	754 ms

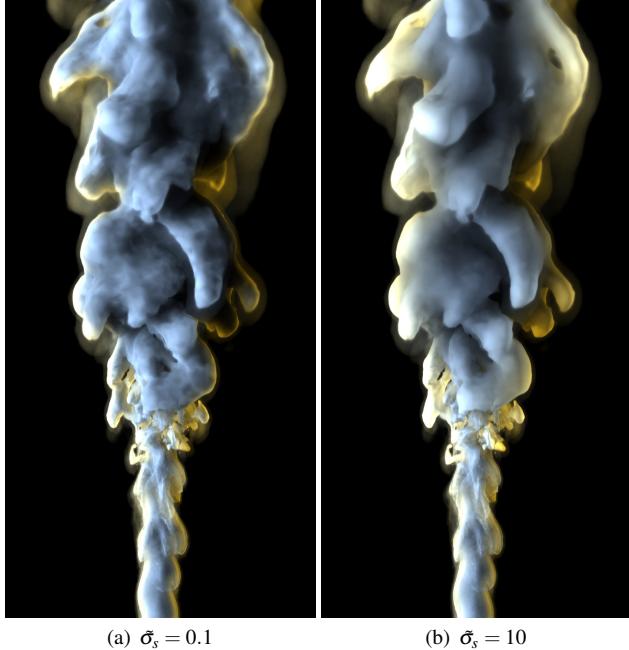


Figure 6: Rendering results with different scattering coefficient $\bar{\sigma}_s$.

with half the resolution of original volume data is able to produce visually plausible results. For the *level-1* grid, an unstructured grid simplification method such as [4] or [5] can be used to generate a simplified grid for light simulation.

7 CONCLUSION AND FUTURE WORK

We developed an illumination method optimized for volume visualization of unstructured-grid data, which uses a two-level light volume calculation scheme. First, a rough but fast light simulation is performed on the *level-0* grid, a regular grid resampled from the original data. Then, the light simulation on the unstructured *level-1* grid is performed to provide detailed lighting effects. The lighting calculation on the *level-1* grid uses a novel numerical scheme that is fully GPU accelerated to solve the model equation. The added global shadowing and multiple scattering effects enhance depth cues and perception of complex spatial relationships for volume visualization. Test results show that the performance of our method is sufficient for interactive volume exploration.

Our method does not rely on heavy precomputation. The only necessary precomputation for the method is the resampling for the *level-0* grid. Compared to other essential precomputation tasks such as gradient estimation used for gradient-based Phong shading, the preprocessing time for our method poses a negligible computational

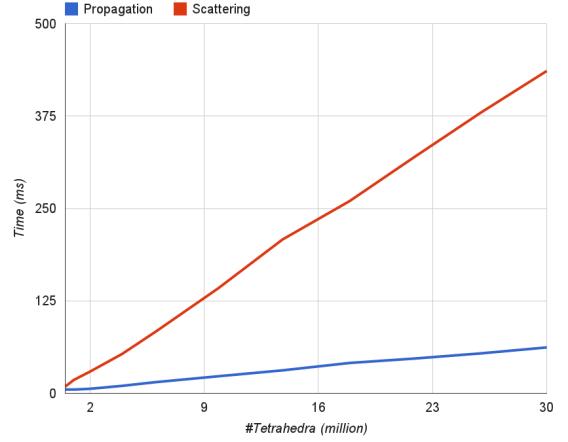


Figure 7: *level-1* grid calculation times for different data sizes. The calculation time increases linearly as the data size (in terms of the number of tetrahedra) increases.

Table 2: Measurements of simulation time for different sizes of *level-0* grid. The Vortex dataset is used.

<i>level-0</i> size	Propagation	Scattering
$32 \times 32 \times 32$	22 ms	1 ms
$64 \times 64 \times 64$	38 ms	3 ms
$128 \times 128 \times 128$	55 ms	10 ms
$256 \times 256 \times 256$	88 ms	100 ms
$384 \times 384 \times 384$	143 ms	334 ms

burden. This makes it suitable for use with time-varying data visualization.

It is not hard to extend our method to support other types of grids including curvilinear grids, adaptive mesh refinement (AMR) grids, and even hybrid grids. Under our framework, the *level-1* grid can be replaced by the desired grid type. For AMR grids, a multi-level scheme can be used to fit the grid resolution.

In the future, we would like to support illumination of large-scale, time-varying unstructured grid data. Thanks to the parallel nature of our technique, it is straightforward to implement the algorithm on multi-GPU machines or GPU clusters. Also, as mentioned in Section 6.4, we would like to support level-of-detail to facilitate interactivity for visualization of large datasets. Another topic of future research would be the anisotropic scattering of light, which provides more realistic appearances of materials while our current illumination model considers only isotropic scattering.

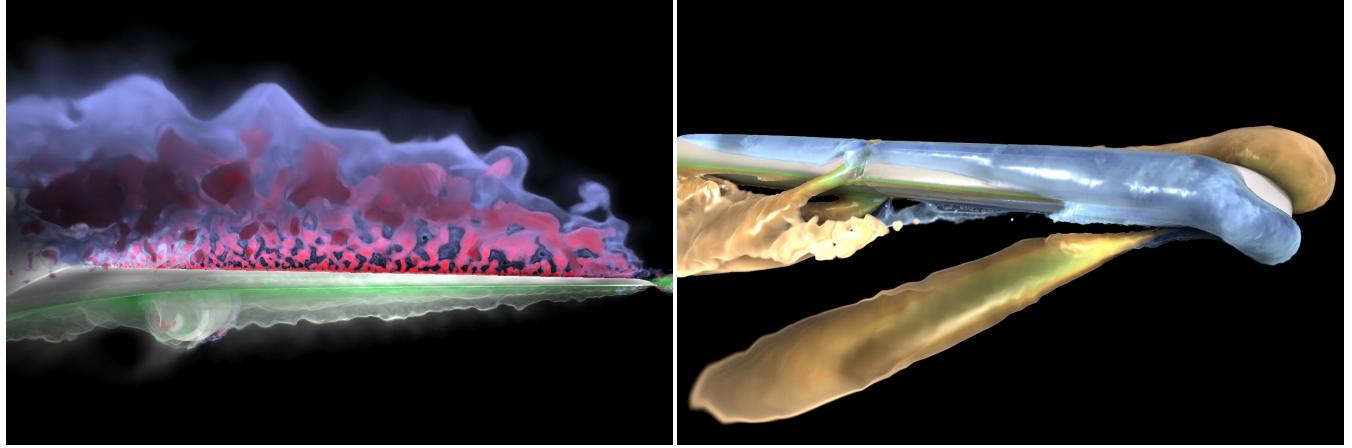


Figure 8: Additional rendering examples using our method.

REFERENCES

- [1] M. Ament, F. Sadlo, and D. Weiskopf. Ambient volume scattering. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2936–2945, Dec 2013.
- [2] A. Boudev, P. Pitot, D. Pratmarty, and M. Paulin. Photon splatting for participating media. In *Proc. of GRAPHITE*, pages 197–204, 2005.
- [3] S. Chandrasekhar. *Radiative Transfer*. Dover Publications, New York, 1960.
- [4] P. Chopra and J. Meyer. Ttfusion: an algorithm for rapid tetrahedral mesh simplification. In *Proceedings of IEEE Visualization 2002 Conference. IEEE*, pages 133–140.
- [5] P. Cignoni, D. Costanza, C. Montani, C. Rocchini, and R. Scopigno. Simplification of tetrahedral meshes with accurate error evaluation. In *Visualization 2000. Proceedings*, pages 85–92, Oct 2000.
- [6] C. D. Correa, R. Hero, and K.-L. Ma. A comparison of gradient estimation methods for volume rendering on unstructured meshes. *IEEE Transactions on Visualization and Computer Graphics*, 17(3):305–319, May 2011.
- [7] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, 5(3):243–255, 1952.
- [8] J. Díaz, P. Vázquez, I. Navazo, and F. Duguet. Real-time ambient occlusion and halos with summed area tables. *Computers & Graphics*, 34(4):337–350, 2010.
- [9] W. Jarosz, M. Zwicker, and H. W. Jensen. The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum*, 27(2):557–566, 4 2008.
- [10] S. Jeffreys and B. Jeffreys. *Methods of mathematical physics*. Cambridge University Press, 1966.
- [11] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. of SIGGRAPH*, pages 311–320. ACM, 1998.
- [12] D. Jonsson, J. Kronander, T. Ropinski, and A. Ynnerman. Historygrams: Enabling interactive global illumination in direct volume rendering using photon mapping. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2364–2371, Dec 2012.
- [13] J. T. Kajiya and B. P. Von Herzen. Ray tracing volume densities. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques, SIGGRAPH '84*, pages 165–174, New York, NY, USA, 1984. ACM.
- [14] J. Kniss, S. Premonoze, C. Hansen, and D. Ebert. Interactive translucent volume rendering and procedural modeling. In *Visualization, 2002. IEEE*, pages 109–116, Nov 2002.
- [15] J. Kniss, S. Premonoze, C. Hansen, P. Shirley, and A. McPherson. A model for volume lighting and modeling. *Visualization and Computer Graphics, IEEE Transactions on*, 9(2):150–162, 2003.
- [16] E. P. Laforte and Y. D. Willems. Rendering participating media with bidirectional path tracing. In *Proc. of EGWR*, pages 91–100, 1996.
- [17] M. Levoy. Efficient ray tracing of volume data. *ACM Trans. Graph.*, 9(3):245–261, July 1990.
- [18] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, June 1995.
- [19] T. Ropinski, C. Döring, and C. Rezk-Salama. Interactive volumetric lighting simulating scattering and shadowing. In *PacificVis*, pages 169–176, 2010.
- [20] S. Röttger, M. Kraus, and T. Ertl. Hardware-accelerated volume and isosurface rendering based on cell-projection. In *Proceedings of the Conference on Visualization '00*, pages 109–116, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.
- [21] M. Ruiz, I. Boada, I. Viola, S. Bruckner, M. Feixas, and M. Sbert. Obscure-based volume rendering framework. In *Proceedings of Volume Graphics 2008*, pages 113–120, Aug. 2008.
- [22] H. E. Rushmeier and K. E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. *SIGGRAPH Comput. Graph.*, 21:293–302, 1987.
- [23] P. Schlegel, M. Makhinya, and R. Pajarola. Extinction-based shading and illumination in GPU volume ray-casting. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):1795–1802, Dec. 2011.
- [24] M. Schott, T. Martin, A. Grossset, S. Smith, and C. Hansen. Ambient occlusion effects for combined volumes and tubular geometry. *Visualization and Computer Graphics, IEEE Transactions on*, 19(6):913–926, 2013.
- [25] P. Shanmugam and O. Arikan. Hardware accelerated ambient occlusion techniques on GPUs. In *Proc. of I3D*, pages 73–80, 2007.
- [26] C. Smith. *The Semi Lagrangian Method in Atmospheric Modelling*. PhD Thesis, University of Reading, 2000.
- [27] M. Weiler, M. Kraus, M. Merz, and T. Ertl. Hardware-based ray casting for tetrahedral meshes. In *Proceedings of the 14th IEEE Visualization 2003 Conference*, pages 333–340, 2003.
- [28] Y. Zhang and K.-L. Ma. Fast global illumination for interactive volume visualization. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '13*, pages 55–62, 2013.