Correlated Photon Mapping for Interactive Global Illumination of Time-Varying Volumetric Data

Daniel Jönsson and Ynnerman Anders

Journal Article

N.B.: When citing this work, cite the original article.

©2016 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Daniel Jönsson and Ynnerman Anders, Correlated Photon Mapping for Interactive Global Illumination of Time-Varying Volumetric Data, IEEE Transactions on Visualization and Computer Graphics, 2016. (), pp..

<http://dx.doi.org/10.1109/TVCG.2016.2598430>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-131022>

Correlated Photon Mapping for Interactive Global Illumination of Time-Varying Volumetric Data

Daniel Jönsson, *Student Member, IEEE*, and Anders Ynnerman, *Associate Member, IEEE*

Abstract— We present a method for interactive global illumination of both static and time-varying volumetric data based on reduction of the overhead associated with re-computation of photon maps. Our method uses the identification of photon traces invariant to changes of visual parameters such as the transfer function (TF), or data changes between time-steps in a 4D volume. This lets us operate on a variant subset of the entire photon distribution. The amount of computation required in the two stages of the photon mapping process, namely tracing and gathering, can thus be reduced to the subset that are affected by a data or visual parameter change. We rely on two different types of information from the original data to identify the regions that have changed. A low resolution uniform grid containing the minimum and maximum data values of the original data is derived for each time step. Similarly, for two consecutive time-steps, a low resolution grid containing the difference between the overlapping data is used. We show that this compact metadata can be combined with the transfer function to identify the regions that have changed. Each photon traverses the low-resolution grid to identify if it can be directly transferred to the next photon distribution state or if it needs to be recomputed. An efficient representation of the photon distribution is presented leading to an order of magnitude improved performance of the raycasting step. The utility of the method is demonstrated in several examples that show visual fidelity, as well as performance. The examples show that visual quality can be retained when the fraction of retraced photons is as low as 40%-50%.

Index Terms—Volume rendering, photon mapping, global illumination, participating media.

1 INTRODUCTION

Interactive visualization of volumetric data through Direct Volume Rendering (DVR) is an important tool for both presentation and exploration in a wide range of domains. The data that need to be handled is of ever-increasing size, resolution and complexity. There is an increasing need to handle data with an inherent time dimension, consisting of a large number of time steps from a simulation or a data collection. Interactive rendering of such spatio-temporal data poses several challenges in terms of data transport and rendering since each slice of data can be of a size normally challenging in static DVR.

The increasing need to analyze time-resolved data is underlined by the rapid development of medical imaging modalities such as Computed Tomography (CT) scanners that capture data with increasingly high spatial and temporal resolution. This opens up new diagnostic opportunities and enable new studies on the functioning of organs.

An important step in DVR is the computation of illumination, which in general, plays an important role in human interpretation of visual information. Studies have shown that illumination can be used for volumetric data to improve shape and depth perception [24], as the light setup has a strong impact on visual cues and can reveal information that would otherwise be difficult to see. The work presented here deals with the challenge of enabling advanced volumetric illumination for time-varying data.

Our method builds on the identification of correlation of the light information between the time-steps in a data set and between changes in visualization parameters. Our starting point is the use of the well-known concepts of photon maps generated by tracing a multitude of photons that are scattered and deposit light in a volumetric illumination map. By finding the correlation between maps we can reduce the number of photons that need to be recomputed by retaining photon traces that are unaffected by the change of time or visualization parameters. Figure 1 highlights two different cases of correlation within time. Region A exhibits small amounts of change and should therefore

Fig. 1: Two time steps from a time-varying 3D ultrasound data set. The goal of our approach is to enable utilization of the correlation in the photon map as seen in the two different frames. Region A includes little visible change and should therefore be able to use the photons computed in the previous frame. Region B, on the other hand, includes a new feature not apparent in the previous frame and should therefore receive a high priority within the photon mapping computation.

not require much attention from the user or the algorithm. Region B, on the other hand, has a feature that disappears and is therefore more likely to be noticed by the user. It should therefore be ensured that the illumination is of high quality in that area. These types of decisions for illumination using photon mapping have mainly been explored for offline rendering [19, 4, 42] and it poses a challenge on how to be able to perform them in an interactive setting utilizing graphics processing units (GPUs). One of the most computationally expensive steps within the photon mapping technique is the gathering of photons. We present an approximation for this step, which enables inexpensive hardware reconstruction to be used to gather the photons, and supports incremental updates of the photon distribution.

The main contributions of this work are:

- A method that enables interactive photon mapping in DVR during parameter and data changes.
- Introduction of a visual importance map, and corresponding metrics, to identify photon traces subject to recomputation.
- Efficient approaches to photon map generation and raycasting with order of magnitude performance improvements compared to previous volumetric photon mapping techniques.

• Daniel Jönsson and Anders Ynnerman are with Linköping University, Norrköping, Sweden. E-mail: {daniel.jonsson, anders.ynnerman}@liu.se.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx/

- Investigation of the perceptual impact of reducing temporal resolution of photon maps using real-world examples.

2 RELATED WORK

Although methods exist for DVR that are capable of recomputing the illumination of the entire scene when the volume data changes, they are severely limited in the types of illumination effects they can support [16]. High quality interactive volumetric illumination methods that have few or no limitations on the illumination complexity are too slow to interactively explore time-varying volumetric data [21]. A comprehensive overview of interactive global illumination techniques can be found in surveys within the topic [16, 31]. Here, we focus on interactive methods for volume rendering and progressive methods.

Interactive Illumination for DVR. Multiple interactive volume illumination techniques have been proposed over the years with the goal of improving realism and the perceptual qualities of volume rendering. Most of the calculation can be precomputed for the ambient occlusion effect only. Hernell et al. [9] calculate the ambient occlusion over the local spherical neighborhood, while Ropinski et al. [33] compute local data histograms. However, these methods do not incorporate physically based global illumination effects. Schlegel et al. [35] used summed area tables (SATs) to achieve interactive dynamic complex lighting. Ament et al. [3] also utilize SATs with different sampling patterns to support coherent calculation of up to six orthodirectional light sources. However, all the light sources need to be recomputed if parameters change. Sunden et al. [37] present a selective update scheme to avoid re-calculation of static light sources and thereby scale up the number of light sources without a significant increase in memory. Our method also supports selective light updates, since the photons are emitted individually per light source, while not placing any restriction on its type or placement. Zhang et al. [44] present a method to compute the light transport by solving a convection diffusion equation. While the method supports dynamic light sources, it does not support time-varying data due to the expensive precomputation. Convolution-based methods, such as those developed by Schott et al. [36] and Patel et al. [28], do not require expensive precomputation but are instead limited in the direction and type of light sources supported. Ritschel [30] was the first to apply spherical harmonics (SH) to interactive DVR, which enables the incorporation of low frequency environmental lighting. Lindemann et al. [23] show that one can incorporate advanced material effects with SH, while Kaplanyan et al. [17] introduce light propagation volumes in combinations with SH to achieve indirect lighting. Kronander et al. [22] improve the performance of the SH computations through space partitioning data structures. Similar to this method, Zhang and Ma [45] compute and store the irradiance in a volume. Our visual importance based approach could potentially also be used to prioritize the irradiance volume calculation in the work of Zhang and Ma [45].

Progressive approaches for DVR. Monte Carlo ray-tracing has been proposed as another alternative for volumetric illumination [34, 21]. Unfortunately, the Monte Carlo ray-tracing based methods come with the computational penalty that require progressive computation to stay interactive if a noise-free rendering is desired. To cope with such issues, several researchers have presented caching techniques. Weber et al. [41] introduced a bidirectional Monte Carlo approach that caches light transport with a set of virtual lights and progressively updates them when the TF changes. Jönsson et al. [15] realized interactive volumetric illumination through photon mapping. The underpinning idea in their work, to only recompute the photons that have changed, is the same as in the work dealt with here. However, the current work and the work of Jönsson et al. [15] are substantially different in the way that the idea is realized. Jönsson et al. [15] cache the used data per ray and per photon using a binary representation. The spatial component is lost in their work and they can therefore not process time-varying data. Our approach uses min-max values instead of a binary representation, which enables us to detect more fine-grained changes. It is also data-centered, i.e. changes are stored in a grid co-located with the data, and based on importance. This means that we can handle both time-varying data as well as prioritize the updates of the photons

based on their importance. Zhang et al. [43] project the photons onto a basis function in a precomputation step, which allows them to reconstruct the out-scattered radiance in real-time during rendering. An extended comparison to these two methods is provided in Section 6.2. The first work on photon mapping was presented by Jensen and Christensen [12]. Although the method was published two decades ago there is now an increasing interest in photon mapping since the introduction of progressive photon mapping (PPM) by Hachisuka et al. [8]. Jarosz et al. [11] introduced photon beams, which are more suitable for participating media with a low extinction coefficient since a single photon beam can represent many spherical photons. Kaplanyan et al. [18] introduced a PPM technique which balances noise and bias in rendered images to minimize the overall error by analyzing the radiance estimation. Thus, they can improve the visual quality of each incremental PPM update and therefore gain acceptable results faster. For dynamic scenes, Dmitriev et al. [6] reuse unaffected photons between frames. However, their method is designed for scenes with geometry, rather than dynamic volumetric data TF changes. It is yet to be shown how to improve the performance of the PPM methods so that they can be used for interactive volume visualization. The irradiance caching method by Jarosz et al. [10] store the radiance gradient in addition to the radiance at each cache point. The radiance gradient is used to determine how sparsely the cache points can be stored. Our visual importance based metric could be used as an orthogonal approach to determining where to place the cache points.

3 BACKGROUND

This section explains the fundamentals of volumetric photon mapping. The standard formulation for a single photon density distribution is extended to deal with state transitions caused by time-varying volumetric data or visual property changes.

3.1 Volumetric Illumination Model

The fundamental building block for a volumetric illumination model is the energy particle, the photon, which is emitted, scattered or absorbed within participating media [26, 13]. Each photon describes the amount of energy it carries from a light source, and together, all photons can be used to reconstruct the flux at a given point $\mathbf{x}$ within the medium, $\Phi(\mathbf{x})$. Once emitted, the photon travels along a straight path through the medium until it is scattered or absorbed. The scattering and absorption coefficients at location $\mathbf{x}$, $\sigma_s(\mathbf{x})$ and $\sigma_a(\mathbf{x})$ form the extinction coefficient $\sigma_t(\mathbf{x}) = \sigma_s(\mathbf{x}) + \sigma_a(\mathbf{x})$ and determine how far the photon travels until it interacts with the media. The average distance it travels from its starting position $\mathbf{x}_s$ can be described by the cumulative density function:

$$F(\mathbf{x}) = 1 - T(\mathbf{x}_s, \mathbf{x}) = 1 - e^{-\int_{\mathbf{x}_s}^{\mathbf{x}} \tau(\mathbf{x}') d\mathbf{x}'}, \quad (1)$$

where $T(\mathbf{x}_s, \mathbf{x})$ denotes the transmittance between two points. The photon scatters at the interaction point $\mathbf{x}$ with a probability given by $\frac{\sigma_s(\mathbf{x})}{\sigma_t(\mathbf{x})}$, and is otherwise absorbed. Russian roulette is used in this work to determine whether it is scattered or absorbed [29]. The photon, i.e. its position, power and direction, is stored each time it interacts with the media. The flux can be utilized to compute the incident radiance L_i , going into the direction $\vec{\omega}$, by integrating the incoming light from all the directions on a sphere given by $\Omega_{4\pi}$:

$$L_i(\mathbf{x}, \vec{\omega}) = \int_{\Omega_{4\pi}} s(\mathbf{x}, \vec{\omega}', \vec{\omega}) \frac{d^2\Phi(\mathbf{x})}{d\vec{\omega}' dV} d\vec{\omega}'. \quad (2)$$

Here, $s(\mathbf{x}, \vec{\omega}', \vec{\omega})$ describes the reflective properties of the material, i.e. how much light is reflected from direction $\vec{\omega}'$ into direction $\vec{\omega}$, and is given by the phase function or bidirectional distribution function. Furthermore, $d\vec{\omega}'$ is the differential direction into which the flux flows and dV is the differential volume in which the flux is considered. The attenuated light reaching the camera at location $\mathbf{x}_c$ is computed by

Fig. 2: Our method focuses on visually de-correlated areas in the volume that are illustrated using 2D grids to efficiently advance the photon distribution. The metadata of the volume together with the transfer function are used to determine the visual impact of a region. The importance grid is traversed by the current photon distribution in order to determine the visual impact of each photon. The photons with the highest importance are updated to form the new photon distribution. The gathering process only needs to consider the difference between the old and new distribution when updating the rendering equation. The pipeline is executed on the GPU as soon as the volume or the TF changes.

attenuating the background radiance $L_0(\mathbf{x}_0, \vec{\omega}_0)$ and integrating the in-scattered and emitted radiance along the ray direction $\vec{\omega}_0$:

$$L(\mathbf{x}_c, \vec{\omega}_0) = T(\mathbf{x}_0, \mathbf{x}_c) L_0(\mathbf{x}_0, \vec{\omega}_0) + \int_{\mathbf{x}_0}^{\mathbf{x}_c} T(\mathbf{x}, \mathbf{x}_c) (\sigma_s L_i(\mathbf{x}, \vec{\omega}_0) + \sigma_a L_e(\mathbf{x}, \vec{\omega}_0)) d\mathbf{x}, \quad (3)$$

The integral in Equation 2 can be approximated by estimating the flux using the n photons within a radius r :

$$L_i(\mathbf{x}, \vec{\omega}) \approx \sum_{p=1}^n s(\mathbf{x}, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_p(\mathbf{x}_p, \vec{\omega}_p)}{\frac{4}{3}\pi r^3} W\left(\frac{\|\mathbf{x} - \mathbf{x}_p\|}{r}\right), \quad (4)$$

where Φ_p is the flux of photon p in direction $\vec{\omega}_p$ at location $\mathbf{x}_p$. $W\left(\frac{\|\mathbf{x} - \mathbf{x}_p\|}{r}\right)$ is a normalized density estimate kernel, with a bandwidth equal to r , used for smoothing. This work uses the Epanechnikov kernel, $W(d) = \frac{3}{4}(1 - d^2)$, $|d| \leq 1$, which has been shown to be optimal given the mean square error metric [7].

Here, we are interested in the distribution of photons over time, i.e. the radiant flux over time $\Phi(\mathbf{x}, t)$. In particular, we use the correlation between two time steps, t_{old} and t_{new} , to propagate the photon distribution into the next time step $\Phi(\mathbf{x}, t_{new}) = \Phi(\mathbf{x}, t_{old}) + \Delta\Phi(\mathbf{x}, t)$. Thus, the photons removed from $\Phi(\mathbf{x}, t_{old})$ together with the photons added in $\Phi(\mathbf{x}, t_{new})$ form $\Delta\Phi(\mathbf{x}, t)$. The following sections will describe how the difference distribution can be efficiently determined and used to calculate the new distribution.

4 METHOD

Our method relies on the data being correlated between time steps or between changes to the optical properties, i.e. the TF. If a correlation exists between changes in the data it also means that the photon distribution is correlated. We utilize this correlation to propagate the photon distribution from one state to the next. By working with the photon state transition we can focus the computation on a subset of the entire photon distribution, which potentially is more efficient.

However, there are caveats when working with the photon distribution transition states; it is only more efficient if the process of identifying the regions that changed and updating them is faster than updating the whole distribution. The process of identifying changed regions requires information about the data and the difference between two time steps. Furthermore, information about the changes in the photon distribution needs to be efficiently dealt with in the two stages of tracing and gathering photons, respectively. The following section will describe how we efficiently identify changes between states, what type of information about the data that is needed, and how to perform the state transition for both the tracing and gathering of photons.

4.1 Overview

The method can be divided into the four major steps depicted in Figure 2. The four steps are performed as soon as the volume data or TF

change. The first step involves finding the de-correlated areas in the volume data between the previous and current frame using pre-computed metadata mapped through the TF. The derived correlation only considers visible changes due to the incorporation of the TF in the computation, and therefore serves as a visual importance map. The visual importance of a photon is determined by integrating the importance along its path. The photons that may affect the rendering can now be determined, which allows the succeeding steps of the method to operate on the difference between the old and new distribution instead of the whole photon distribution. All photons that have a visual impact are updated, or alternatively, a percentage of the total photons can be updated in the order given by their visual importance to gain interactivity. The final step computes the radiance reaching the screen by gathering the photon difference distribution, removing the old photons that where updated, and replacing them with the new ones.

4.2 Visual Importance Map

The TF, which maps the values within a data set to optical properties, is the core component in a visualization of a volumetric data set. Typically, it is designed by the user and assigns optical properties for each feature within the data set that the user wishes to see. One of the most common types of TFs map scalar values to colors and opacities, which is the one that we will be the focus of this work.

The goal of our approach is to determine the visual impact of a change in a data value between two time-steps. A naive method would map each voxel within the two time-steps through the TF and compare the difference in color and opacity. This would provide an accurate map of the regions within the data set that will have an impact on the final rendering. The visual impact of a photon or view-ray can be extracted by traversing the difference map and collecting the total amount of change along its path. Unfortunately, this approach is too computationally expensive in terms of performing the mapping and collecting the information.

Instead, we propose creating a uniform grid where each cell covers several voxels within the original data set as illustrated in the first step in Figure 2. The larger the cell size, the less computationally expensive it becomes at the expense of accuracy in the size of the features it can handle. In this work we use a cell size of 8^3 voxels. For each cell, we store the minimum and maximum values of the voxels it covers. By mapping the values between the minimum and maximum through the TF it is possible to efficiently retrieve the visual impact of the whole region that it represents. Unfortunately, this does not deliver any information on the change between two time-steps since the minimum and maximum values can be equivalent while the values in between change. In order to gain this information we use a second uniform grid of the same size, which describes the mean absolute difference within the cell between the two consecutive time steps. The minimum, maximum and mean absolute difference for each cell are computed once in a pre-processing step.

An alternative approach would be to store the density distribution within each cell and compare them between time-steps. However, this

would require more computation and memory while being less robust since the density distribution may be the same even if the data values change.

Our method can handle both volume data changes, i.e. time-varying data, as well as TF changes. We will first outline the case of changing TF followed by the case of time-varying data.

4.2.1 Visualization Parameter Changes

When the TF changes and the volume data remains constant, it is only necessary to consider the minimum and maximum values within a cell. The maximum change between the old and new TF within the minimum and maximum value is stored in the visual importance map. We use the distance in Lab color space for the colors in order to retrieve a perceptual metric. The color distance is normalized to fall within the same range as the opacity, $[0, 1]$, and then added to the difference in opacity to form the final visual correlation value ψ_{TF} for a cell with minimum, V_{min} , and maximum, V_{max} values:

$$\psi_{TF} = \max_{v \in [V_{min}, V_{max}]} \left(\frac{\left\| \frac{d}{dt} \vec{C}(v, t) \right\|}{C_0} \right) + \max_{v \in [V_{min}, V_{max}]} \left(\left| \frac{d}{dt} \alpha(v, t) \right| \right). \quad (5)$$

Here, $\vec{C}(v, t)$ is the TF color component for scalar value v at time t , $C_0 = \|(100, 500, 400)\|$ is the normalization factor for the Lab color and $\alpha(v, t)$ is the TF opacity for scalar value v at time t . Note that colors with zero opacity are ignored and that the time parameter refers to when the user changes the visualization parameter. The equation above will ensure that the visual importance for a cell will be high for the data value ranges where the user is making large changes to the TF and otherwise low or zero.

A range of other metrics can be integrated into the pipeline, such as multiplying the two terms and thus placing higher priority on large changes than on small ones. However, the user is not often changing both the color and opacity at the same time. Multiplying the two terms would in this case result in a zero importance, which does not reflect the visual change in the rendering. Applying addition to the two terms ensures that individual color and opacity changes produce non-zero importance.

4.2.2 Volume Data Changes

Time-varying data changes can use a similar metric to the visual property changes, but instead of considering the partial derivative in time we consider the partial derivative in scalar value for the TF. In other words, we determine the visual impact of a varying scalar value by calculating how much the visual parameter changes within scalar's value range. The change in volume data is taken into account using the mean absolute difference between two time steps of the scalar field within the cell, $E[|V(t_i) - V(t_{i-1})|]$:

$$\psi_V = E[|V(t_i) - V(t_{i-1})|] \left(\max_{v \in [V_{min}, V_{max}]} \left(\frac{\left\| \frac{d}{dv} \vec{C}(v, t) \right\|}{C_0} \right) + \max_{v \in [V_{min}, V_{max}]} \left(\left| \frac{d}{dv} \alpha(v, t) \right| \right) \right). \quad (6)$$

Note the change of partial derivative to scalar value, $\frac{d}{dv}$, for the visual properties compared to time in Equation 5. A cell with zero difference in volume data, or a TF with constant parameters in $[V_{min}, V_{max}]$, will result in a weight of zero since they will not have a visual impact on the rendering. Cells with highly varying data or TF will receive higher weights, while regions with small changes in data or TF will receive a lower weight. Assuming that linear interpolation is used to reconstruct a time-step the minimum and maximum values can be interpolated between discrete time steps, while the mean absolute difference is the same in-between two discrete time-steps. Note that $[V_{min}, V_{max}]$ for time-varying data is the minimum and maximum of the same cell in the current and previous time-step. The weight $\psi = \psi_{TF} + \psi_V$ is computed and stored for each cell in parallel.

4.3 Photon Prioritization

The visual importance map is used as the underlying function when deciding if a photon needs to be updated, and in what order. The function is integrated from the start position of the photon, $\mathbf{x}_s$, to the end of its path, $\int_0^1 \psi(\mathbf{x}_s + s\vec{\omega}_p) ds$. The integrated importance is accumulated across frames such that photons with many small errors will be eventually updated. The importance of a photon is reset as soon as it has been updated. All photons are sorted according to their importance using indices instead of the their underlying data to reduce computation and memory overhead. The sorted indices are used as a queue from which the photon updating phase will take its work from.

4.4 Advancing Photon Distribution State

The maximum number of photons permitted during one iteration are taken from the prioritized queue. A sorting, based on spatial location, is performed before the photons are updated, i.e. re-traced. We discovered that updating the photon in the importance based order degraded the performance by up to an order of magnitude compared to the spatially sorted photons. The spatial sorting is therefore essential in achieving good performance on GPUs. The photon difference distribution, i.e. the indices to the updated photons and their updated data, is sent to the photon gathering stage for radiance estimation.

4.5 Photon Gathering

Once the photons have been traced into the scene we can estimate the incoming radiance towards the camera by computing the photon density along the view rays, confer Equation 4. In essence, we must gather all the photons within a given radius of the view ray. This can be done using a brute force method, that for a given set of points along the view ray, intersects all photons in the scene and discards those that are further than the given radius. However, the brute force method is too computationally expensive for interactive purposes and other methods have therefore been explored [15, 43]. Unfortunately, they have not been able to reach interactive frame-rates for time-varying data.

We suggest an alternative approach that reduces both the photon gathering and the ray-casting time. Revisiting Equation 4, we can see that for each step along the view-ray it is necessary to collect all the photons within the photon radius and perform shading. We propose dividing the equation into two steps, one where the density estimation is performed, and stored and one where the shading is done, such that the expensive photon density estimation can be reused for multiple steps along the view-ray. The density estimation is performed once when the photons change for each position $\mathbf{x}_i$ in a grid:

$$\Phi(\mathbf{x}_i) \approx \sum_{p=1}^n \frac{\Phi_p(\mathbf{x}_p, \vec{\omega}_p)}{\frac{4}{3}\pi r^3} W(\|\mathbf{x}_i - \mathbf{x}_p\|/r). \quad (7)$$

The equation above can be extended to handle time-varying data and visual parameter changes by including time and the photon difference distribution. We divide the photon difference distribution $\Delta\Phi_p(\mathbf{x}, t)$ into two parts; one containing the old photons from the previous time step that was affected by the change, defined as $\Delta\Phi_p(\mathbf{x}_p, \vec{\omega}_p, t_{old})$, and one containing the updated photons at the new time step, defined as $\Delta\Phi_p(\mathbf{x}_p, \vec{\omega}_p, t_{new})$. The new flux can now be computed by removing the old photons affected by the change and adding the new ones:

$$\Phi(\mathbf{x}_i, t_{new}) \approx \Phi(\mathbf{x}_i, t_{old}) + \sum_{p=1}^n \frac{\Delta\Phi_p(\mathbf{x}_p, \vec{\omega}_p, t_{new})}{\frac{4}{3}\pi r^3} W(\|\mathbf{x}_i - \mathbf{x}_p\|/r) - \sum_{p=1}^n \frac{\Delta\Phi_p(\mathbf{x}_p, \vec{\omega}_p, t_{old})}{\frac{4}{3}\pi r^3} W(\|\mathbf{x}_i - \mathbf{x}_p\|/r). \quad (8)$$

Due to the two operations, it is only beneficial to perform the computation when the change correspond to half of the contributing photons. The flux at a given point can be reconstructed using linear interpolation within the grid, which is much less costly than computing Equation 7. Unfortunately, this means that the directional component of the

Fig. 3: (a) and (b) display renderings of contrast enhanced vessels and bone from a CT scan of a head, while (c) shows the structural similarity between them. (d) The optical properties have been changed to introduce a transparent layer of skin. Similar to Jönsson et al. [15], our method can detect the invariance in the photon distribution from (a) and recompute the variant photons to propagate the distribution into the state resulting in (d). The presented approximation can improve the performance by almost an order of magnitude and therefore allow interactive exploration of time-varying data.

photons is lost. However, we can regain an approximation of the direction during ray-casting by performing shading per-light source and weighing the contribution by the relative power of the light sources:

$$L_i(\mathbf{x}, \vec{\omega}) \approx \sum_{l=1}^{L_n} s(\mathbf{x}, \vec{\omega}_l, \vec{\omega}) \Phi(\mathbf{x}) W_l, \quad (9)$$

where L_n is the number of light sources in the scene, $\vec{\omega}_l$ is the direction from the light source position and W_l is the normalized weight proportional to the power of the light sources in the scene. The approximation relies on the fact that high frequency effects are retained using normal direction calculated at the sample point. It is also possible to include the shading in Equation 7, making it view-dependent, and recompute it as soon as the camera changes. However, our experiments showed that high frequency shading effects are lost unless an unfeasible high resolution grid is used. In Section 6.2, we will show that our new approach enables photon mapping to be used for camera changes as well as time-varying volume data. Further details on how to implement the gathering approach are given at the end of Section 5.

The method of Jönsson et al. [15] will be used for accurate renderings, when no changes are occurring, while our approximation is used during change.

5 IMPLEMENTATION

The implementation of the method has been realized using a mixture of OpenGL and OpenCL. Sourcecode can be found at GitHub [14]. All the steps outlined in the overview (Figure 2) are performed on the GPU while the metadata precomputation is performed on CPU. The last step, the ray-casting, uses an OpenGL implementation to share code with other algorithms that sample a flux volume, while the other steps are implemented using OpenCL.

The visual importance grid is computed in parallel by evaluating Equations 5 and 6 for each cell. An importance grid cell that evaluates below a threshold of 10^{-4} is set to zero. The two equations are solved by utilizing the fact that the TF implementation is based on a piecewise linear representation for both color and opacity. For Equation 5, the derivative can thus be calculated by using the difference between the two piecewise linear curves in time. In practice, the derivative only needs to be evaluated at control-points that changed between two time steps. Thus, each importance grid cell evaluates the difference between the few control-points that changed between the two TFs, lying between $[V_{min} V_{max}]$, and selects the maximum. A similar approach is used for Equation 6, with the difference that time is exchanged for the intensity value. An alternative approach to is to use pre-computed

lookup tables as performed in the work by Šoltészová et al.[39]. However, precision is lost when using lookup tables, which would lead to unnecessary recomputation of photons, and it does not scale well with higher dimensional TFs.

The integration of the visual importance for each photon is discretized by traversing the grid using a 3D digital differential analyzer (3D-DDA) [2, 5]. Here, the final integrated photon importance is represented using an integer such that they can be sorted efficiently on the GPU using the radix-sorting algorithm [27]. The total number of photons that need to be updated are counted using parallel reduction. If the number of photons requiring updating are more than the allowed computational load per frame, we continue to update photons iteratively until new input arrives. Each iteration performs a spatial sorting before tracing the photons. The sorting is based on the photon location hashed through a uniform grid with the same cell size as the visual importance grid. The importance of a photon is reset to zero as soon as the photon has been recomputed.

The indices of the recomputed photons together with their corresponding data are sent to the photon gathering stage. The gathering stage keeps a copy of the previous photon state, which it uses to remove the contribution of the updated photon before adding the contribution of the recomputed photon. We evaluated two different approaches to computing the grid resulting from Equation 7, gathering and splatting. The gathering approach goes through each voxel and collects the flux using a chaining hash table [1] data structure to find the nearby photons. The splatting approach goes through each photon with non-zero contribution and adds its contribution to the voxels within the radius using atomic operations. The two approaches were evaluated on a flux-volume of half the original volume size, which we also use for the examples in this work. We concluded that the splatting approach was faster than building the chaining hash table and gathering the photons. We therefore use the splatting approach for all the following measurements. When available, the ray-casting algorithm uses the min-max grid to perform empty space skipping using 3D-DDA traversal.

6 RESULTS AND EVALUATION

In this section, we will describe the quality and performance of our method using a range of time-varying and static data sets provided in Table 1. The time-varying data sets are used to show how the method performs with changing data, while the static data set shows how the method performs when changing the TF. All the tests were performed on a computer with an Intel Core i5 3.2 GHz processor, 8 GB random access memory and an Nvidia Geforce 980 GPU.

Fig. 4: Photographic light source setup on a data set of a fetus from a pre-natal ultrasound examination. The head of the fetus can be seen in the top, the arm in the middle, and the legs in the lower part of the images. The left rendering uses a single key light while the right rendering also includes a fill light. The fill light softens the hard shadows created by the key light and makes it easier to see features in the shadowed areas. Data courtesy of Context Vision AB.

Fig. 5: A fused MRI and fMRI data set displayed using our method with single (left) and multiple scattering (right). The material representing the fMRI signal is assigned to scatter incoming photons, which makes it easier to discover and distinguish the fMRI signal.

6.1 Results

We start by showing a comparison to previous state-of-the-art within interactive volumetric photon mapping and then move on to show a range of different illumination effects that can be achieved using the photon mapping technique. Then, the results of a visual quality assessment for time-varying data based on updating a fixed percentage of the photons each frame is shown. Lastly, the performance of the photon importance, tracing and gathering stages are provided. All the results were created with a one-dimensional TF, which can be viewed in the video in the supplementary material.

6.2 Comparison With Previous Work

Previous work on interactive volumetric photon mapping has dealt with static data [43] and changing optical properties [15]. For comparison, we used the Golden lady data set (Figure 3) with the same setup as [15] (Table 1), i.e. 1.5 samples per voxel, 2.1 Million photons and nine ray-segments. Whereas they stored and used the history of the data for each photon and view-ray to determine which ones to update, we instead make use of the visual importance grid. Two gathering approaches were evaluated, one presented in [15], which uses a hash table approach to find the nearby photons during ray-casting, as well as our approximation based detailed in Section 4.5. For the first gathering approach, our method performs slightly worse: 1.85 versus 2.4 times speedup when performing the change seen in Figure 3(a)-(d), i.e. going from a TF displaying vessels and bone to having a thin transparent layer of skin. Nevertheless, these results are expected since their way of querying invalid photons is more efficient. Note, however, that their method cannot handle time-varying data as their information is decoupled from the spatial changes in the data. The second, approximative gathering approach, receives a performance speedup of

8.0 times while producing the visual result seen in Figure 3(b). The performance gain allows us to retrieve interactive performance during exploration using the camera and is the key enabler for interactive exploration of time-varying volumetric data using photon mapping.

Compared with the work of Zhang et al. [43], which perform interactive camera exploration, our approximative gathering approach requires less memory, since a single value is used instead of many coefficients for the basis functions. The construction time of their grid with basis coefficients is orders of magnitude slower, and requires seconds to minutes of computation time compared with a few to a hundred milliseconds in our approach as evident in Table 1.

6.2.1 Prenatal ultrasound

Figure 4 displays a rendering of a 4D prenatal ultrasound data set using our method with a photographic light setup. The first light source, placed above to the right of the camera, is used as key light while the second light source, placed below to the left of the camera, is used as fill light. Both renderings make use of a back light. The fill light softens the hard shadows created by the key light and therefore creates a more aesthetically pleasing rendering for the future parents. While many interactive volume illumination techniques are limited by the number or type of light sources supported [38, 36, 28], the photon mapping technique is not restricted in this manner.

6.2.2 Brain activity

The brain activity data set is a combination of two scans that use magnetic resonance imaging (MRI) and functional MRI (fMRI), where the structural components are given by the MRI data and the brain activity information is given by the fMRI data, respectively. The task for the subject was to read sentences which had one missing word and then think about a word that could be inserted while being in the fMRI scanner. The fMRI signal is shown using a probabilistic animation [25], which means that the most active regions are displayed longer. Furthermore, brain activity below half of the signal strength is thresholded to remove active areas that are likely to be unrelated to the task. The two data sets have been fused together into a single data set, such that the lower bits are used to represent the MRI signal and the upper bits are used for the fMRI signal. Figure 5 displays the structural brain information obtained from the MRI scan using a brown material, while the fMRI is represented using a yellow-orange material. Our method is capable of scattering light multiple times, which is demonstrated in Figure 5 (right). It can be seen that, by making the fMRI material scatter photons, it is possible to further enhance the perception of the fMRI signal. While a few existing interactive methods are capable of producing multiple scattering for volume data, they do so using large approximations [32] or are limited to forward scattering [20].

6.2.3 Beating heart

The heart data set represent one heart beat cycle that has been scanned using computed tomography. A contrast agent has been injected into the patient to highlight the blood flow in the heart, which allows the physician to investigate the function of the heart. The heart beat sequence, seen in the middle row of Figure 6, starts with the heart being in a relaxed state, then proceeding into a contracted state and then returning back to the relaxed state. The data set contains a considerable amount of noise, which causes the contrast agent to frequently appear and disappear as well as change color during the entire sequence. It is therefore particularly challenging to detect and utilize the invariance between time-steps within the visualization. The next section will demonstrate how well the method handles this type of data where essentially all voxels change value between time-steps.

6.3 Visual Quality

In order to assess how well the method uses the correlation between time-steps we set a maximum number of photons that were permitted to be recomputed during one frame. As comparison with the visual importance based method we assign equal importance to each photon and use an update pattern that changes each frame. Each photon will be recomputed every other frame when updating 50% of the photons,

Fig. 6: An overview of the time-varying behavior of the data sets used for evaluation at equally spaced time-steps. (Top) Ultrasound scan of a fetus where features frequently appear and disappear over time. (Middle) Computed tomography scan of a heart beat with a significant amount of noise. The heart starts in a relaxed state, proceeds into a contracted state and then returns back to the relaxed state. (Bottom) Fusion of a MRI scan of the brain (brown tissue) and a functional MRI scan (yellow-orange) depicting the oxygenated areas when performing a language task.

Table 1: List of data sets used. P_n denotes the number of photons used. The performance numbers have been obtained with a photon recomputation setting that does not generate any visual errors.

Data set (resolution)	P_n (M)	Samples /voxel	Tracing (ms)	Splatting (ms)	Raycasting (ms)
Fetus, fig 6 (top) 518 × 327 × 510 × 29	3.2	1.0	47	53	16
Heart, fig 6 (middle) 512 × 512 × 640 × 20	1.6	2.0	49	12	48
Brain, fig 6 (bottom) 306 × 378 × 318 × 10	4.1	1.5	9	35	29
Golden lady, fig 3 512 × 512 × 625	2.1	1.5	65	14	40

four out of ten in each frame when updating 40%, and so on. The rendered images of both methods are compared to a reference where all photons have been updated.

The mean structural similarity (MSSIM) index is used as the error metric, which is a widely known method for image quality assessment [40]. Contrary to absolute error metrics, such as mean square error or peak signal to noise ratio, the MSSIM metric uses a perception-based model and is therefore more suitable to use for image quality assessment. The index values are in the range $[-1, 1]$, where -1 is the worst and 1 is equivalent to the original image.

The visual importance based method can achieve virtually lossless quality (MSSIM > 0.995) when updating 50% of the photons for all the examples included in this paper. The main reason for this is that 40 – 98% of the photons can be identified as invariant between frames and therefore do not have a visual impact on the rendering. These photons are often passing through space within the data set that has constant visual properties, which for example occurs when the extinction coefficient is set to zero. The number of photons invariant to change is naturally dependent on the data set, the optical properties and the light source setup. For example, the ultrasound fetus data set contains features such as a moving head that appear and disappear throughout the time sequence. These changes are difficult to cope with without considering visual importance as seen in Figure 7. Reducing the updated percentage of photons to below 20% causes a significant degradation in quality due to the large changes in the data and few updated photons. More advanced importance metrics could potentially improve the visual quality.

The heart data set, on the other hand, has a high degree of temporal noise, which causes color shifts in addition to the appearing and disappearing features. These types of changes affect all of the visualized features and therefore, it is harder to determine which ones are visually important. In particular, if the user is interested in blood flow

in the ventricles (seen in the middle of the rendering in Figure 8(b)) the mean absolute difference metric will assign higher importance to the noisy surrounding structures. Although it can be seen in Figure 8 that the visual importance approach produces better image quality, the noise misleads the visual importance based metric for lower photon recomputation thresholds.

The brain activity data set contains a majority of invariant photons between time steps due to the use of a contextual MRI brain as seen in Figure 9, where only 10% of the photons are updated. The amount of photons changing between time-steps for this sequence varies between 1 – 15%. While the use of equal importance produces visible errors, the visual importance method is able to detect the changing areas affected by both single and multiple scattering, and thereby maintain a visually error-free rendering.

6.4 Performance

In this section, we focus on the performance of time-varying data since we have already shown that the method can gain almost an order of magnitude in comparison with previous works for static data described in section 6.2. The baseline for the measurements was set to the time it takes to perform photon tracing and gathering to a flux volume for all photons. The average photon computation times were 108 ms, 61 ms and 129 ms, with a photon radius of 4.0, 2.4 and 4.0 voxels, for the fetus, heart and brain, respectively.

The performance for the visual importance based approach, including the overhead involved in determining which photons to update, was evaluated for a range of visual quality settings given by the maximum percentage of photons to update each frame. The highest percentage of photons to update was set to give a visually error free image and then reduced by ten percent at the time. The resulting speedup compared to the baseline is given for each data set in Figure 10. At first, one would expect that the performance gain would be roughly equivalent to the percentage of photons that are recomputed. However, deeper analysis revealed that the photons passing through empty space do not take as long to trace, and more importantly, less time to gather to the flux volume. This means that the performance gain when having lossless quality for the fetus and heart data set ranges from six to seventeen percent rather than the initially expected one hundred percent. The performance gain for these two data sets increases by roughly ten percent for each ten percent photon recomputation decrease until it decreases below half of the initial error free percentage threshold, whereupon the performance increases more rapidly. The reason for the performance gain at the lower recomputation thresholds is that the gathering to flux volume can begin to benefit from adding and removing photons. The brain activity data set has a higher correlation between frames and therefore gains higher performance speedups as seen in Figure 10(c).

(a) Image quality per frame when updating 40% of the photons using the visual importance based method (dotted line) and equal importance (solid line).

(b) Ref., frame 37. (c) Equal importance. (d) Visual importance.

Fig. 7: (Top) The visual importance based method produces consistently better image quality compared to using equal importance when updating 40% of the photons for the ultrasound fetus data set. (Middle) Visual comparison of the reference and the two methods at frame 37. (Bottom) Color mapped SSIM error of (c) and (d).

The computational overhead of computing the importance grid, gathering the importance for each photon, and sorting them according to importance is provided in Figure 11. By extrapolating down to zero photons it can be seen that there is a small initial overhead, that stems from the computation of the importance grid. The computation time varies linearly with the number of photons, but is usually only a few percentages of the time it takes to be traced and gather all of the photons.

6.4.1 Memory consumption

Each photon requires 32 bytes, 12 for position, 12 for the flux (red/green/blue) and 8 for the normalized direction in spherical coordinates. Additional memory is required to maintain the different states of the photons, temporary variables for sorting, and indices to changed photons. The additional memory increases the memory requirement by roughly four times per photon. The photon distribution volume consumes four bytes per voxel, i.e. floating point precision, when used.

Higher resolution data requires smaller, and thus more, photons to accurately capture detailed features within the data. The flux volume resolution needs to scale linearly with the input data resolution if details are to be retained. The photons benefit from a sparse representation, i.e. they are only stored at locations where features are visible, and therefore scale better with larger data sets compared with the flux volume. Thus, the main memory bottleneck when scaling to higher resolution data is the flux volume.

6.4.2 Metadata memory consumption and calculation

The minimum, maximum and the mean difference between two volumes are computed once and stored along with the volume data. Two bytes are used for the minimum and maximum values, while four bytes

(a) Image quality per frame when updating 40% of the photons using the visual importance based method (dotted line) and equal importance (solid line).

(b) Ref., frame 24. (c) Equal importance. (d) Visual importance.

Fig. 8: (Top) The visual importance based method produces consistently better or equal image quality compared to using equal importance when updating 40% of the photons for the heart data set. (Middle) Visual comparison of the reference and the two methods at frame 24. (Bottom) Color mapped SSIM error of (c) and (d).

are used for the mean volume difference. The memory consumption is given by the cell size used for the grid relative to the size of the data set. A cell size of eight will result in a memory consumption of eight bytes times $1/8^3 \approx 1.5\%$ the number of voxels in the data set. As an example, the heart beat consumes 3.3 Gb, while the metadata require 51 Mb. The pre-computation time for two time-steps is in the order of hundreds of milliseconds using a CPU implementation. The metadata memory requirement is almost negligible compared to the volume data set memory consumption. The main benefit of increasing the cell size is not the consumed memory, but rather the speed improvement for traversing the grid and computing the visual importance. Increasing the cell size by two theoretically reduces the traversal time by half and can be used to reduce overhead for higher resolution data.

The visual importance approach can be extended to handle multidimensional TFs by storing additional necessary metadata and extending the importance metric accordingly.

7 CONCLUSIONS, LIMITATIONS AND FUTURE WORK

In this work we have approached the challenging problem of enabling advanced global illumination of spatio-temporal volumetric data. Our overall goal is to maintain the highest image quality possible while updating as few photons as possible. The utility of our method is demonstrated using real-life data sets and we have shown how interactive rendering of time-varying data is obtained together with illumination involving multiple scattering effects in participating media. The underpinning idea is based on the correlation of photon maps during updates of the data in a time series or during parameter changes such as TF updates or camera moves. A central part of the method is the generation of visual importance maps used to identify regions in which photons need to be retraced. The results show that applying the presented visual importance metric produces significantly better results compared

(a) Image quality per frame when updating 10% of the photons using the visual importance based method (dotted line) and equal importance (solid line).

(b) Ref., frame 31. (c) Equal importance. (d) Visual importance.

(e) Color map. (f) SSIM, equal importance. (g) SSIM, visual importance.

Fig. 9: (Top) The brain activity data set has a large proportion of constant features due to the static context. The visual importance based method identifies all of the variant photons and does not produce any error, while the equal importance method produces visible errors. (Middle) Visual comparison of the reference and the two methods at frame 31. (Bottom) Color mapped SSIM error of (c) and (d).

to removing photons using a uniform pattern. It was also shown that the introduced overhead is less than the performance gain. It should be noted that in extreme cases where all the data values change and are visualized at the same time belong to one of the cases where the method only adds computational overhead. The applied visual metric is a simple estimate of importance, which is easy to implement, but as described in the results section, has difficulties in handling noisy data sets such as a beating heart. Nevertheless, this work has opened up the possibility of using more advanced importance metrics in interactive settings.

Another important component of our method is the reduction of the computational overhead associated with photon gathering, by representing the photon distribution in a grid co-registered with the original data grid. We have shown that this method for approximating photon gathering gained almost an order of magnitude in performance compared to previous work when it came both to construction and gathering. However, it should be noted that this is an approximation and there are cases when the error is more apparent. Examples of such cases are use of multiple scattering or area light sources, which require individual photon directions to be correctly shaded.

Future work will involve using more advanced metrics for visual importance mapping as well as implementing our method in streaming paradigms for large scale out-of-core data.

ACKNOWLEDGMENTS

We thank all reviewers for their valuable comments which helped to improve this paper. Special thanks go to Erik Sundén for help with the related works and Konrad Schönborn for proofreading the manuscript. This work was supported by grants from the Swedish e-Science Research Centre (SeRC), the Swedish Research Council (VR) grant

(a) Performance improvement per frame for the fetus data set

(b) Performance improvement per frame for the heart data set

(c) Performance improvement per frame for the brain activity data set

Fig. 10: Speedups when recomputing a maximum percentage of photons using the visual importance metric. A speedup of one denotes the time taken to recompute all photons. The solid line in each plot represents the speedup obtained without loss of image quality. Photons passing empty space does not add much overhead since the speedup would be larger for the solid lines in (a) and (b). Furthermore, (a) and (b) shows that the speedup increases substantially when reaching below half of the error free recomputation threshold (i.e. below 25%), which is where the splatting computation begins to benefit from working with the difference distribution.

Fig. 11: Measurements of the time required to determine which photons should be updated, and in which order, for an increasing number of photons. The overhead compared to tracing all the photons and gathering them into a flux volume is 4.5%, 3.8% and 2.5% for the fetus, heart and brain data sets, respectively.

2015-05462 and the Knut and Alice Wallenberg Foundation (KAW) grant 2013-0076. The presented concepts have been realized using the Inviwo open source visualization framework (www.inviwo.org).

REFERENCES

- [1] D. A. Alcantara. *Efficient Hash Table on the GPU*. PhD thesis, University of California, Davis, California, USA, 2011.

- [2] J. Amanatides, A. Woo, et al. A fast voxel traversal algorithm for ray tracing. In *Eurographics*, volume 87, page 10, 1987.
- [3] M. Ament, F. Sadlo, C. Dachsbacher, and D. Weiskopf. Low-Pass Filtered Volumetric Shadows. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2437–2446, 2014.
- [4] J. Chen, B. Wang, and J.-H. Yong. Improved stochastic progressive photon mapping with metropolis sampling. In *Computer Graphics Forum*, volume 30, pages 1205–1213. Wiley Online Library, 2011.
- [5] J. G. Cleary and G. Wyvill. Analysis of an algorithm for fast ray tracing using uniform space subdivision. *The Visual Computer*, 4(2):65–83, 1988.
- [6] K. Dmitriev, S. Brabec, K. Myszkowski, and H.-P. Seidel. Interactive global illumination using selective photon tracing. In *Proc. EGSR*, pages 25–36, 2002.
- [7] V. A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.
- [8] T. Hachisuka, S. Ogaki, and H. W. Jensen. Progressive photon mapping. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2008)*, pages 130:1–130:8, 2008.
- [9] F. Hernell, P. Ljung, and A. Ynnerman. Local Ambient Occlusion in Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 16(4):548–559, 2010.
- [10] W. Jarosz, C. Donner, M. Zwicker, and H. W. Jensen. Radiance caching for participating media. *ACM Transactions on Graphics (TOG)*, 27(1):7, 2008.
- [11] W. Jarosz, M. Zwicker, and H. W. Jensen. The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2):557–566, Apr. 2008.
- [12] H. W. Jensen. Global illumination using photon maps. In *Rendering Techniques*, pages 21–30. Springer-Verlag, 1996.
- [13] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of SIGGRAPH 1998, SIGGRAPH '98*, pages 311–320. ACM, 1998.
- [14] D. Jönsson. Source code. <https://github.com/ResearchDaniel/Correlated-Photon-Mapping-for-Interactive-Global-Illumination-of-Time-Varying-Volumetric-Data>, 2016 (accessed June 15, 2016).
- [15] D. Jönsson, J. Kronander, T. Ropinski, and A. Ynnerman. Historygrams: Enabling Interactive Global Illumination in Direct Volume Rendering using Photon Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2364–2371, 2012.
- [16] D. Jönsson, E. Sundén, A. Ynnerman, and T. Ropinski. A Survey of Volumetric Illumination Techniques for Interactive Volume Rendering. *Computer Graphics Forum*, 33(1):27–51, 2014.
- [17] A. Kaplanyan and C. Dachsbacher. Cascaded Light Propagation Volumes for Real-time Indirect Illumination. In *Proc. of SIGGRAPH Symp. on Interactive 3D Graphics and Games*, pages 99–107, 2010.
- [18] A. S. Kaplanyan and C. Dachsbacher. Adaptive progressive photon mapping. *ACM Transactions on Graphics*, 32(2):16:1–16:13, Apr. 2013.
- [19] A. Keller and I. Wald. *Efficient importance sampling techniques for the photon map*. Universität Kaiserslautern. Fachbereich Informatik, 2000.
- [20] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson. A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):150–162, 2003.
- [21] T. Kroes, F. H. Post, and C. P. Botha. Exposure Render: An Interactive Photo-Realistic Volume Rendering Framework. *PLoS ONE*, 7(7), 2012.
- [22] J. Kronander, D. Jönsson, J. Löw, P. Ljung, A. Ynnerman, and J. Unger. Efficient visibility encoding for dynamic illumination in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):447–462, 2012.
- [23] F. Lindemann and T. Ropinski. Advanced light material interaction for direct volume rendering. In *IEEE/EG Int. Symp. on Volume Graphics*, pages 101–108, 2010.
- [24] F. Lindemann and T. Ropinski. About the Influence of Illumination Models on Image Comprehension in Direct Volume Rendering. *IEEE TVCG (Vis Proceedings)*, 17(12):1922–1931, 2011.
- [25] C. Lundström, P. Ljung, A. Persson, and A. Ynnerman. Uncertainty visualization in medical volume rendering using probabilistic animation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1648–1655, Nov. 2007.
- [26] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.
- [27] D. Merrill and A. Grimshaw. High Performance and Scalable Radix Sorting: A case study of implementing dynamic parallelism for GPU computing. *Parallel Processing Letters*, 21(02):245–272, 2011.
- [28] D. Patel, V. Šoltészová, J. M. Nordbotten, and S. Bruckner. Instant convolution shadows for volumetric detail mapping. *ACM Transactions on Graphics (TOG)*, 32(5):154, 2013.
- [29] M. Pharr and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2004.
- [30] T. Ritschel. Fast GPU-based Visibility Computation for Natural Illumination of Volume Data Sets. In *Eurographics*, pages 17–20, 2007.
- [31] T. Ritschel, T. Grosch, C. Dachsbacher, and J. Kautz. State of the art in interactive global illumination. *Computer Graphics Forum*, 31(31):160–188, 2012.
- [32] T. Ropinski, C. Döring, and C. Rezk Salama. Advanced Volume Illumination with Unconstrained Light Source Positioning. *IEEE Computer Graphics and Applications*, 2010.
- [33] T. Ropinski, J. Meyer-Spradow, S. Diepenbrock, J. Mensmann, and K. Hinrichs. Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Proceedings of Eurographics 2008)*, 27(2):567–576, 2008.
- [34] C. Salama. Gpu-based monte-carlo volume raycasting. In *Pacific Conference on Computer Graphics and Applications*, pages 411–414, 2007.
- [35] P. Schlegel, M. Makhinya, and R. Pajarola. Extinction-based Shading and Illumination in GPU Volume Ray-Casting. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1795–1802, 2011.
- [36] M. Schott, V. Pegoraro, C. Hansen, K. Boulanger, and K. Bouatouch. A directional occlusion shading model for interactive direct volume rendering. *Computer Graphics Forum (Proceedings of EG/IEEE Symposium on Visualization 2009)*, 28(3):855–862, 2009.
- [37] E. Sundén and T. Ropinski. Efficient Volume Illumination with Multiple Light Sources through Selective Light Updates. In *Proceedings of the 2015 IEEE Pacific Visualization Symposium*, pages 231–238, 2015.
- [38] E. Sundén, A. Ynnerman, and T. Ropinski. Image plane sweep volume illumination. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2125–2134, 2011.
- [39] V. Šoltészová, Å. Birkeland, S. Stoppel, I. Viola, and S. Bruckner. Output-sensitive filtering of streaming volume data. In *Computer Graphics Forum*. Wiley Online Library, 2016.
- [40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.
- [41] 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.
- [42] M. Weiss and T. Grosch. Stochastic progressive photon mapping for dynamic scenes. *Computer Graphics Forum*, 31(2pt3):719–726, 2012.
- [43] Y. Zhang, Z. Dong, and K.-L. Ma. Real-time volume rendering in dynamic lighting environments using precomputed photon mapping. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1317–1330, Aug. 2013.
- [44] Y. Zhang and K.-L. Ma. Fast Global Illumination for Interactive Volume Visualization. In *Proc. of SIGGRAPH Symp. on Interactive 3D Graphics and Games*, pages 55–62, 2013.
- [45] Y. Zhang and K.-L. Ma. Decoupled Shading for Real-time Heterogeneous Volume Illumination. *Computer Graphics Forum*, 2016.