Vision, Modeling, and Visualization (2013)
Michael Bronstein, Jean Favre, and Kai Hormann (Eds.)

Interactive Direct Volume Rendering with Many-light Methods and Transmittance Caching

Christoph Weber[†1,2] , Anton S. Kaplanyan[‡1] , Marc Stamminger[§2] and Carsten Dachsbacher[1]

[1]Karlsruhe Institute of Technology, Germany
[2]Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

a. Single scattering only, computed using ray-marching in 30 ms.

b. Multiple scattering only, with proposed many-light rendering method, 860 ms.

c. Single and multiple scattering combined with proposed method, 890 ms.

Figure 1: Global illumination for direct volume rendering of the MANIX dataset (at 800×800 resolution): (a) single scattering, computed using ray marching, only reveals some of the internal structures; (b) multiple scattering (shown isolated) computed used the proposed method; (c) full global illumination (single and multiple scattering) reveals more internal structures and improves the realism. The proposed method is based on many-light approach and transmittance caching and enables interactive rendering as well as interactive editing of transfer functions under full global illumination.

Abstract

In this paper we present an interactive global illumination method for Direct Volume Rendering (DVR) based on the many-light approach, a class of global illumination methods which gained much interest recently. We extend these methods to handle transfer function and volume density updates efficiently in order to foster ability of interactive volume exploration. Global illumination techniques accounting for all light transport phenomena are typically computationally too expensive for interactive DVR. Many-light methods represent the light transport in a volume by determining a set of virtual light sources whose direct illumination and single scattering to a view ray approximate full global illumination. Our technique reduces computation caused by transfer function changes by recomputing the contribution of these virtual lights, and rescaling or progressively updating their volumetric shadow maps and locations. We discuss these optimizations in the context of DVR and demonstrate their application to interactive rendering.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Direct Volume Rendering (DVR) methods are one of the favored approaches for exploration and presentation of volume media, such as scientific and medical data. The quality and

† christoph.weber@cs.fau.de
‡ {anton.kaplanyan, dachsbacher}@kit.edu
§ marc.stamminger@informatik.uni-erlangen.de

speed of these methods have greatly improved over the past decades. More recent work, such as Jönsson et al. [JKRY12], also explores the use of global illumination for volumetric lighting beyond single-scattering, and thus abandons the tradeoffs that have usually been made in DVR with respect to Transfer Functions (TFs) and compositing. These attempts are motivated by the idea that realistic illumination greatly contributes to the visual information present in the images and indirect lighting is an important visual cue. Realistic lighting can ease the interpretation by a human observer and reveal important global as well as local (volumetric) structures.

In this paper, we present a DVR technique with complete illumination based on the Many-Light (ML) formulation of light transport. This formulation enables efficient computation of global illumination, and gained much interest and underwent significant progress in recent years [DKH*13]. ML methods offer a unified mathematical framework for light transport reducing full transport to the computation of direct illumination and single scattering from many Virtual Point Light (VPL) sources. Their desirable property is the scalability: they are able to produce artifact-free images in a fraction of a second, and converge to the full solution given more computation time.

Although conceptually simple, light transport in participating media is costly to compute no matter which approach is used. In addition to light transport sampling, this is mainly due to the computation of transmittance along segments of transport paths. When using many-light methods, this computation is mostly required for connecting virtual lights, that is, computing their single scattering contribution to view rays. We can precompute this in form of Adaptive Volumetric Shadow Maps (AVSMs) [SVLL10], as was done by Engelhardt et al. [ENSD12]. However, a computationally expensive update is required for all VPLs when the transfer function is modified. Moreover, the locations of the VPLs themselves have to be updated. To our knowledge, our paper is the first applying many-lights rendering to DVR and makes the following contributions to greatly improve the rendering speed as well as the temporal stability:

- We enable DVR with interactive progressive rendering of global illumination (including multiple scattering), and interactive transfer function updates.
- We show how to recompute the contribution of VPLs and present a progressive update scheme for their locations and cached transmittances when the TF is modified.
- In our method the global scaling of volume densities (or TF opacity, respectively) is a special case that requires no costly updates.

2. Previous Work

In this section we review related work categorized into global illumination and interactive techniques for direct volume rendering. Both are mature fields, and thus we resort to most closely related works and point to comprehensive surveys where appropriate.

Global Illumination. The study of physically based light transport dates back to the midst of the 20th century [Cha60] and has subsequently evolved into families of methods for both transport between surfaces and in participating media, such as unbiased Monte Carlo [Vea98, PH04], lattice-based [Cha60], photon mapping [Jen96, JC98], and many-light methods [Kel97, RDGK12, DKH*13]. This list indicates that there is a huge body of previous work in this field. As there is also a recent state-of-the-art report on many-light methods [DKH*13], on which our work builds, we limit our discussion to a brief recapitulation of the most important and closely related works. Many-light methods originate from Instant Radiosity (IR) [Kel97]. This method computes transport paths starting from the light sources of a scene, and creates virtual point lights (VPLs) at locations where light-surface interactions take place. Full global illumination is approximated by computing direct lighting from these VPLs. Raab et al. [RSK08] extended this idea to participating media where multiple scattering in the medium is approximated by single scattering from VPLs. Many-light methods are unbiased, but in their basic form suffer from singularities visible as bright splotches near VPLs. Clamping the VPLs' contributions removes these artifacts, but at the same time removes energy from the solution and introduces bias. Bias compensation techniques recover this clamped energy, but are costly to compute [RSK08]. Engelhardt et al. [ENSD12] analyze the energy loss in media and present an efficient, yet approximate, compensation. Novak et al. [NED11] approximated the compensation from (incomplete) image-space data. Another strategy is to regularize such singularities [KD13b], for example, by distributing the energy of a point light across lines [NNDJ12a, NNDJ12b], or a finite volume which we do with Virtual Spherical Lights (VSLs) [HKWB09] in our renderer.

A crucial component of many-light rendering in participating media is accumulating the contributions from many virtual lights. This is done by ray marching along the view rays and computing a virtual light's contribution, which requires accounting for the transmittance between two points. This can be accelerated using deep shadow maps [LV00], or their GPU-friendly variant, Adaptive Volumetric Shadow Maps (AVSMs) [SVLL10]. These shadow maps sample and store an approximation of the transmittance along rays originating from the light source, and thus allow for efficient transmittance queries after they have been created.

Interactive Illumination for DVR. As global illumination is challenging for interactive applications, in particular for participating media, often approximations are made to improve shading realism while maintaining interactivity for DVR. The STAR by Jönsson et al. [JSYR12] provides an excellent overview of recent progress. Popular approxima-

tions are ambient occlusion [RMSD*08] or forward scattering with texture slicing [KPH*03]. Volumetric shadows can be achieved with the aforementioned deep shadow maps and variants of this technique. Precomputation of light transport has also been used, for example, for storing visibility information using spherical harmonics [KJL*12] or handling different materials under natural lighting conditions [LR10].

Related to our work, Salama [Sal07] adopts stochastic raytracing for isosurface volume rendering on GPUs. Kroes et al. [KPB12] describe a framework for Monte Carlo ray tracing for DVR, including scattering in participating media. However, modifications to the TF always require a recomputation of the image from scratch. A recent approch of Zhang et al. [ZM13] caches the light transport in a medium using a convection diffusion equation and supports heterogeneous media as well as interactive updates. Their caching scheme covers all but the last step of the light transport, including the costly transmittance evaluation. While this method is crafted for plausible real-time volumetric rendering, we strive for an accurate visualization. Also, our method is not bound by any requirements to the medium.

Jönnson et al. [JKRY12] take a similar approach to global illumination for DVR as we do: they extend volumetric photon mapping such that photons and view-ray segments carry additional information about the parameters that affect their contribution to the lighting solution. This allows to update TFs at interactive speed. Photon mapping and many-light methods are both bidirectional Monte Carlo techniques with advantages and disadvantages. Photon mapping relies on density estimation and requires a large number of photons to be traced. Zhang et al. [ZDM13] incorporate photon mapping into a precomputed radiance transfer framework, thus providing interactive rendering at the cost of long precomputation for static media with a fixed transfer function. Many-light methods, on the other hand, require orders of magnitude less light paths (and thus virtual lights), and rendering can be very efficient, in particular with volumetric shadow mapping. However, they become less efficient with high-frequency scattering/phase functions [DKH*13]. Note that in any case, the result can be progressively refined and converges to the correct solution. As the global part of multiple scattering is often low-frequency, we believe that this approach is beneficial for interactive DVR. Note that lighting details, such as fine structures and volumetric shadows thereof are not removed from the solution: the single scattering contribution (virtual lights to view ray) accounts for this and is computed accurately. Lastly, many-light rendering is not only simpler to implement than photon mapping, but also convergences faster asymptotically [KD13a].

3. Preliminaries: Volumetric Rendering with VPLs

In order to make this paper self-contained, we briefly recapitulate volumetric light transport with virtual lights in this section. This section follows the formulation and notation of Engelhardt et al. [ENSD12], but we consider volumetric rendering only. For example, we assume the absence of explicitly defined opaque surfaces. The radiative transport in this case (please see the definitions in Table 1) is formulated as

$$L_o(\mathbf{x},\omega) = L_e(\mathbf{x},\omega) + \int_{\mathcal{S}^2} \sigma_s(\mathbf{x},\omega) f(\mathbf{x},\omega,-\omega_i) L_i(\mathbf{x},-\omega_i) d\omega_i,$$

where the phase function f describes the probability of energy being scattered from the incident direction ω_i to the direction ω, and $\sigma_s = \sigma_t - \sigma_a$ indicates the fraction of light scattered at point $\mathbf{x}$. The inscattered radiance term $L_i(\mathbf{x},\omega)$ at point $\mathbf{x}$ is computed as

$$L_i(\mathbf{x},\omega) = \int_{t_{min}}^{t_{max}} T(\mathbf{x},\mathbf{x}+t\omega) L_o(\mathbf{x}+t\omega,-\omega) dt, \quad (1)$$

where the integration is performed along the ray $\mathbf{x}+t\omega$ in the range $[t_{min};t_{max}]$ (bounded by the medium). The term $T(\mathbf{x},\mathbf{y})$ is the *transmittance* and defines the fraction of radiance that remains after outscattering and absorption on a path between two points $\mathbf{x}$ and $\mathbf{y}$. It is defined as

$$T(\mathbf{x},\mathbf{y}) = e^{-\int_0^{|\mathbf{x}-\mathbf{y}|} \sigma_t(\mathbf{x}+t\omega) dt}. \quad (2)$$

Eq. 1 is recursive with respect to the radiance $L_i(\mathbf{x},\omega)$ and can be expanded for all paths up to the length n as

$$L_i^n(\mathbf{x},\omega) = T_n \sum_{k=1}^{n-1} \underbrace{\int .. \int}_{k \text{ times}} T_k \, f_k \, d\mu_k .. d\mu_{k-1} \, L_e, \quad (3)$$

with $T_k = T(\mathbf{x}_k,\mathbf{x}_{k-1})$ and $f_k = \sigma_s(\mathbf{x}_k) f(\mathbf{x}_k,\omega_k,\omega_{k-1})$. As depicted in Fig. 2, the radiance from the light source (vertex $\mathbf{x}_0$) is transported to the camera (vertex $\mathbf{x}_n$) via scattering at points $\mathbf{x}_k$ inside the medium. Based on Eq. 3 we regroup the

Table 1: Notation used throughout this paper.

Symbol	Description	Units
$L(\mathbf{x},\omega)$	Radiance at position $\mathbf{x}$ towards direction ω	$[Wm^{-2}sr^{-1}]$
L_i, L_o, L_e	Incident, outgoing and emitted radiance	$[Wm^{-2}sr^{-1}]$
$f(\mathbf{x},\omega,\omega')$	Phase function	$[sr^{-1}]$
$\sigma_s,\sigma_a,\sigma_t$	Scattering, absorption and extinction	$[m^{-1}]$
$\tau(\mathbf{x},\mathbf{y})$	Optical thickness between $\mathbf{x}$ and $\mathbf{y}$	–
$T(\mathbf{x},\mathbf{y})$	Transmittance between $\mathbf{x}$ and $\mathbf{y}$	–
(d_i,t_i)	Depth and transmittance cached in AVSM	–
$\hat{S}$	Density scale, a multiplier for σ_t	–
$\bar{T}$, $\bar{S}$	Average transmittance and global scale	–
$p(t)$	Pdf for sampling distance along the ray	–
r	Radius of VSL	[m]
$d\mu$	Differential volume measure	–
$\mathcal{S}^2$	Unit sphere domain in 3-dimensional space	–
$\mathcal{U}(a,b)$	Uniform distribution in the range (a,b)	–

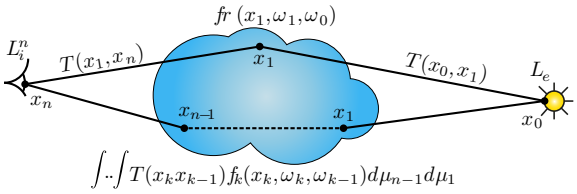

Figure 2: Two different light transport paths. The top-most illustrates single scattering, while the bottom path illustrates multiple scattering with $n-1$ scattering events in between.

radiance computation as

$$L_i^n(\mathbf{x},\omega) = T_n \sum_{k=1}^{n-2} T_{k-1} \underbrace{\int .. \int T_k \, f_k \, d\mu_k .. d\mu_{k-1} L_e}_{\text{stored as VPLs}} .$$

This equation allows us to store scattering vertices $\mathbf{x}_1..\mathbf{x}_{n-1}$ inside the medium as virtual lights (see Fig. 2) and reuse them for volumetric rendering using many-light methods.

3.1. Distributing Virtual Lights

Computing a set of N virtual lights is equivalent to subsampling the global part of light transport. The distribution of VPLs is computed using local importance sampling of the corresponding phase functions at every vertex of the path. The algorithm consists of the following steps.

1. Starting from the light source, sample an origin and a direction and cast the initial ray.
2. Sample the length of the path segment depending on the transmittance of the medium along that ray (as in Alg. 1). If the resulting position is in the medium, place a VPL. Otherwise terminate the path.
3. Using Russian roulette based on the remaining path throughput, either terminate the path (if $T_{\text{path}} < \mathcal{U}(0,1)$) or reweight its contribution by $1/(1-T_{\text{path}})_+$ and continue
4. Importance-sample the phase function at the resulting position to get a new direction and return to step 2.

We keep spawning new paths until we reach the desired total number of VPLs. For a VPL at the path vertex $\mathbf{x}_k$ coming from the path edge $\overline{\mathbf{x}_{k-1},\mathbf{x}_k}$, we store the incident radiance L_i^k and direction ω_i^{k-1} that is always sampled at the previous step $k-1$,

$$L_i^k = \frac{T(\mathbf{x}_k,\mathbf{x}_{k-1})}{p(t_{k-1})} L_o^{k-1}, \text{ with}$$

$$L_o^{k-1} = \frac{1}{p(\omega_k)} \begin{cases} L_e & \text{if } \mathbf{x}_{k-1} \text{ is on the light} \\ L_i^{k-1} f_{k-1} & \text{else, in the medium} \end{cases},$$

and the incident radiance $L_i^j = L_i(\mathbf{x}_j, \omega_{j-1})$.

3.2. Sampling Path Segment Distance

The random walk samples directions according to the importance sampling pdf $p(\omega_j)$ of the phase function f_j. Along path segments, we also need to sample the distance the light particle travels through the medium before the next scattering event occurs. The ideal probability density function (pdf) for sampling this distance is proportional to the transmittance along this ray [WMHL65]. Therefore, we construct the sampling pdf $p(t)$ by performing a stochastic simulation of the mean path length using Alg. 1. Instead of sampling the transmittance, we sample the optical thickness $T = e^{\S\tau}$. Note, when changing the extinction properties σ_t, this alters the transmittance and thus also the sampling pdf $p(t)$. As a consequence all virtual lights should be redistributed (see Sect. 4 for handling of this case).

Algorithm 1 Sampling path segment distance. An additional factor $\S$ is a uniform scaler of the medium density.

```
function SAMPLEDISTANCE(x_start, ω)
    ξ ← −ln(U(0,1)/§)        ▷ τ of avg. transmittance
    τ ← 0; x ← x_start
    while τ < ξ do           ▷ until particle is absorbed
        τ ← τ + §σ(x)dt      ▷ accumulate thickness τ
        x ← x + ω dt         ▷ do a step along the ray
    end while
    return x                 ▷ return sampled position
end function
```

3.3. Rendering with Virtual Lights

During the image rendering, we gather the contribution of all VPLs and illuminate every shading point with

$$L(\mathbf{x},\omega) = \frac{1}{NS} \sum_{t=0}^{S} T(\mathbf{x},\mathbf{x}_t) \sum_{j=0}^{N} f_t T(\mathbf{x}_t,\mathbf{x}_j) \frac{f_j L_i(\mathbf{x}_j,-\omega_j)}{\left|\mathbf{x}_t-\mathbf{x}_j\right|^2}, \quad (4)$$

where S denotes the number of samples $\{l_t\}$ along the view ray $\mathbf{x}_t = \mathbf{x} - l_t\omega$, uniformly distributed along it in the medium; N is the number of VPLs, and j is the index of the VPL; the two phase functions are defined as $f_t = \sigma_s(\mathbf{x}_t) f(\mathbf{x}_t, \omega, \overline{\mathbf{x}_t\mathbf{x}_j})$ and $f_j = \sigma_s(\mathbf{x}_j) f(\mathbf{x}_j, \overline{\mathbf{x}_j\mathbf{x}_t}, \omega_j)$.

Rendering with Virtual Spherical Lights. The inverse distance term between the shading point and the VPL introduces a singularity visible as a bright splotch if a shading point is close to the VPL. These artifacts disappear very slowly with increasing numbers of VPLs. To mitigate these singularities, we apply a spatial regularization to our virtual lights by spreading the emitted energy across a finite spherical volume around the center of the VPL, also called Virtual Spherical Lights (VSLs) [HKWB09]. Now, instead of connecting to the center of a VPL, we need to integrate over the solid angle Ω_j of the VSL's sphere, visible from a shading point $\mathbf{x}_t$. This solid angle is computed from a user-defined radius of the VSL r_j and the distance $\left|\mathbf{x}_t-\mathbf{x}_j\right|$ between the

VSL and the shading point. We can then approximate Eq. 4 using VSLs instead of virtual point lights as

$$L(\mathbf{x},\omega) \approx \frac{1}{NS}\sum_{t=0}^{S} T(\mathbf{x},\mathbf{x}_t)\sum_{j=0}^{N} f_t T(\mathbf{x}_t,\mathbf{x}_j)\frac{f_j L_i(\mathbf{x}_j,-\omega_j)}{\pi r_j^2 p(\Omega_j)}, \quad (5)$$

where $p(\Omega_j)$ is the pdf for sampling the cone spanned by the VSL. Note that we avoid an actual integration over the visible spherical cap by considering only one sample per VSL. Otherwise, the integration would require costly ray marching for transmittance evaluation.

3.4. Transmittance Caching

Most of the render time with VLs in Eq. 5 is the evaluation of transmittance between all shading points $\mathbf{x}_t$ and all virtual lights $\mathbf{x}_j$, as it requires the transmittance computed with ray marching through the medium. Adaptive Volumetric Shadow Maps (AVSMs) [SVLL10] store m pairs (t_i, d_i) of transmittance and depth values in every texel of a cube shadow map. By generating one AVSM per VL, this allows for a piecewise linear approximation of the transmittance during lighting computation. In our implementation we use AVSMs with a resolution of 48^2 pixels for each cubemap face (in many-light rendering low-resolution shadow maps are typically sufficient [RGK*08]), and use $m = 8..12$ depth samples to approximate the transmittance towards each light (Fig. 3). We further increase the quality by performing trilinear interpolation of the cached transmittances (bilinear on the cubemap face and linear along the depth). All the renderings in this paper employ trilinear interpolation.

4. Editing the Transfer Function

Although caching transmittance with AVSMs greatly accelerates the rendering, interactive editing of the transfer function requires a recomputation of all VLs including their AVSMs. This makes responsive editing of the TF difficult. Not only the recomputation is expensive, but the redistribution would result in a very different set of VLs and this – since VLs subsample the full light transport – results in sudden changes of the global illumination approximation (unless a very large number of VLs is generated, which is usually impractical for interactive rendering, see Fig. 4).

Instead, we propose to update VLs progressively to achieve faster response, more predictable results, and better temporal stability. We update VLs in two phases: First, the contributions of all VLs are updated, while their positions and AVSMs remain unchanged. This is very cheap as it only requires reevaluating phase functions and transmittance for a few VLs, and results in a first immediate response. Second, outdated VLs are progressively removed and new VLs are generated by tracing new light subpaths considering the new TF until all VLs are updated.

The necessity of deleting and redistributing outdated VLs is demonstrated when adding tissue in MANIX scene as in Fig. 6. VLs that lie in affected places might no longer contribute to the final image as the medium can absorb the entire path contribution before reaching the first VL (since $\sigma_s = 0$).

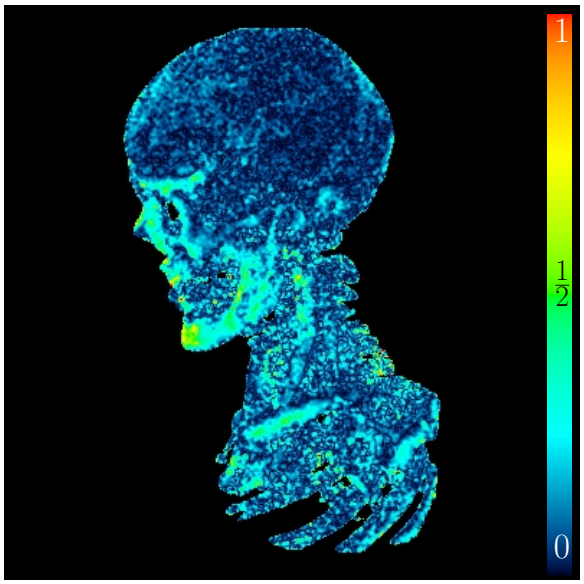

Figure 3: The relative error introduces by caching transmittance using AVSMs; error is normalized by pixel intensity and evaluated on MANIX dataset. AVSM resolution of 40^2 pixels per cubemap face with $m = 8$ depth samples each.

As a special interaction mode we also consider changing the transfer function by a global scaling factor. We show how VLs can be distributed in a way that is useful for an entire interval of scaling factors, so that no complete redistribution is necessary and immediate response becomes possible. We describe this option in Sect. 5 and focus on a more general approach first. Note that in this case we do not remove tissue but scale its extinction properties uniformly across the medium.

4.1. Updating VL Contributions

If the transfer function is modified, the positions of all VLs remain unchanged, only their contribution is updated according to the new transfer function. To this end, we update the contribution of each VL by retracing the light subpath

$$L^k_{i,\text{new}} = \frac{T_{\text{new}}(\mathbf{x}_k,\mathbf{x}_{k-1})}{p_{\text{new}}(t_{k-1})} L^{k-1}_{e,\text{new}}. \quad (6)$$

T_{new} and p_{new} are computed using ray marching along the established paths. This operation is cheap, but the VLs are no longer optimally distributed, resulting in higher variance. Moreover, at this point the AVSMs for each VL still contain the old transfer function. Such transport paths account for the correct throughput except for their last segment. This is

Figure 4: Different sets of VLs (each containing 200 lights) can result in very different renderings (left, center). The difference image on the right is computed as a relative difference of pixel intensities as $|I_{\text{left}} - I_{\text{center}}| / I_{\text{left}}$.

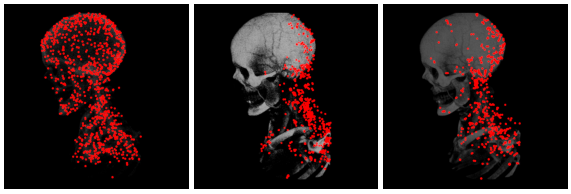

Figure 5: VL distributions influenced by different global medium density. Left to right: (1) sparse medium; (2) dense medium; (3) distribution according to the average transmittance (Eq. 8). The density of the medium is proportional to its reflectance (Ş scales σ_t).

a compromise to achieve immediate feedback at the price of approximate results.

4.2. Progressive Regeneration of VLs

After updating the VLs contributions, we progressively remove old invalidated VLs in the order of their creation and replace them by generating new light subpaths according to the updated transfer function. To get faster and smoother visual feedback, we only update a fixed small fraction of light subpaths every frame. Recomputing the new VL positions is cheap, but it also requires the recomputation of their AVSMs, which is a more costly step. We apply two rules

1. The AVSMs of the repositioned VLs are regenerated anew, that is, depth and transmittance values are computed. This leads to longer response times (Table 2) while providing the best results.
2. The AVSMs of the remaining VLs are left unchanged.

The outdated AVSMs cause noticeable rendering errors (Fig. 6). Also note that any changes to σ_t affect all AVSMs, independently of the VLs contributions. Thus, modifying σ_t anywhere in the volume requires a complete update of all AVSMs.

5. Global Scaling of Transfer Function

In this section we consider a particular modification of the TF, namely, a *global* scaling of a medium density (TF's opacity). Scaling the density can lead to a significant impact on the distribution of VLs due to the distance sampling (Fig. 5). However, we will now derive a way to distribute VLs such that we can keep their positions fixed, yet still obtain an acceptable result for all scaling factors. We now express the transmittance from Eq. 2 as

$$T_{\text{Ş}}(\mathbf{x}_j, \mathbf{x}_{j-1}) = \exp\left(-\int_0^t \text{Ş}\sigma_t(\mathbf{x}_{j-1} + \omega s)ds\right), \quad (7)$$

where $t = |\mathbf{x}_j - \mathbf{x}_{j-1}|$, and ω is the direction from $\mathbf{x}_{j-1}$ to $\mathbf{x}_j$. In this case, the ideal sampling pdf would be the scaled transmittance. However, as we uniformly scale the medium density (potentially from a very thin to a very thick medium) – while keeping the VLs' positions fixed – we should consider the impact of such a scaling on the distribution of VLs. Very thin media creates almost uniform distributions of VLs, whereas dense media cause the VLs to be clustered closer to the light sources (Fig. 5). The distribution that minimizes the error for both extremes requires an average sampling pdf. We derive this pdf from the mean transmittance $\bar{T}_{\bar{\text{Ş}}}$, which itself results from an average scaling parameter. We use the arithmetic mean

$$\bar{T}_{\bar{\text{Ş}}} = \frac{1}{2}(T^{\text{Ş}_{\min}} + T^{\text{Ş}_{\max}}), \quad (8)$$

where the values $\text{Ş}_{\min}$ and $\text{Ş}_{\max}$ limit the range of possible scales. The reasoning behind sampling the length of path segments according to such a mean value is to obtain a VL distribution as independent as possible of the interactive scaling and usable for the entire range of possible scaling values. We extend the second step of our VL distribution scheme accordingly.

1. We trace a ray in direction ω_{k-1} (sampled at the previous VL or on the light source).
2. We accumulate the transmittance for the entire length of the ray. Note that we do not consider completely opaque or infinite media.
3. Next, we compute the mean transmittance using Eq. 8.
4. Using the resulting transmittance $\bar{T}$, we evaluate the mean scaling parameter $\bar{\text{Ş}}$ for sampling the new VL position $\mathbf{x}_j$ as $\bar{\text{Ş}} = \ln(\bar{T})/-\tau$, where $\tau = \int_0^t \sigma_t(s)ds$ is the *optical thickness* of the original medium along the current ray.

We can now define the termination criterion for sampling the length using the introduced average density scale (Alg. 1).

5.1. Updating VLs

After each modification to the scaling parameter, we update the contribution of all VLs as in Eq. 6. However, in this case we do not need to change $p(t)$ and as the scaling is independent of σ_t (Eq. 7), we can update the transmittance without

Table 2: Timings in ms for multiple scattering depending on the number of VLs (image resolution 800^2). We compare Ray Marching (RM) with caching of transmittance (point sampling). Bilinear interpolation is almost twice slower. Additional timing for pre-caching of transmittance with AVSM resolution 40^2 texels $\times 6$ faces $\times$ 8 samples.

# VLs	RM	AVSM		
		Creation	Point sampling	Bilinear
100	720	100	90	160
500	3600	500	460	860
1000	7220	990	870	1510
2000	14600	2300	1850	3080

ray marching as

$$T_{\text{new}} = T_{\text{old}}^{\frac{s_{\text{new}}}{s_{\text{old}}}}.$$

We can also easily modify the color of the light or the medium, or the ratio between scattering σ_s and absorption σ_a, thus making a medium more or less absorbent. Note, that the described uniform scaling and sampling with an average transmittance does not require a redistribution but always provides a suboptimal estimation. However, when allowing for a progressive update, it is commendable to distribute according the actual transmittance of the medium.

5.2. Editing the Phase Function

For modeling anisotropic scattering we use the Heyney-Greenstein (HG) [HG41] phase function throughout our system. Analogous to the mean transmittance, we can interpret isotropic scattering as the mean phase function obtained from the HG-parameter interval. As scattering tends towards isotropic propagation after several bounces anyway, we use this assumption to generate VL positions that provide a reasonably good approximation for varying scattering behavior. Note, that many-light methods, in particular when using only hundreds to thousands of VLs, are restricted to phase functions with moderate directionality [ENSD12].

6. Results

By replacing ray marching with AVSM sampling we can shorten the render time significantly. The timings in this section were measured with an NVIDIA® GTX Titan. Table 2 shows the performance gain of transmittance caching compared to ray marching. We also provide the creation time of the visibility cache. Note that even large sets of 2000 VLs produce visibly different renderings for identical scenes (Fig. 4). Convergence requires several orders of magnitude more VLs, impractically many for interactive visualization. The timings were taken from a subsampled MANIX dataset with an extend of 256^3. The rendering of larger datasets would benefit even more from the transmittance caching, though the cache creation would become slower.

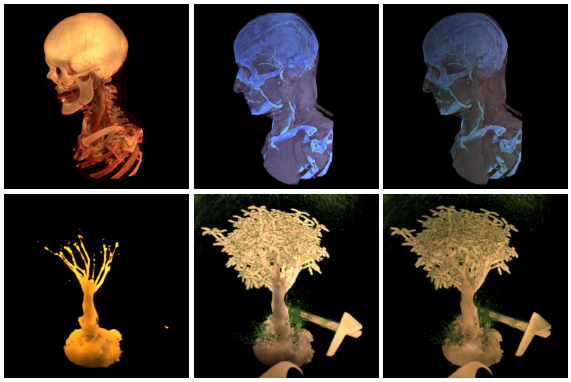

Figure 6: Two scenes: MANIX (top) and BONSAI (bottom). Columns, left to right: (1) initial VL distribution; (2) example of transfer function editing (added muscles, changed color) without redistribution of VLs and without AVSM update; (3) after progressive redistribution of VLs. Note that (2) provides an immediate update but the AVSMs provide wrong transmittance values. Option (3) shows the correct result after the convergence, that is, after the complete redistribution of VLs and recreation of AVSMs. For detailed images please zoom into the electronic version.

Fig. 6 shows interactive TF edits, like an addition of soft tissues. The center column yields the immediate update of VL contributions without any correction of positions or the resampling of AVSMs. The right column shows the medium after a progressive resampling. Most prominent is the increased brightness in the center images which stems from the wrong transmittance values. The initial medium has a smaller extend, so regions that appeared after editing are not captured by the AVSMs. Only a complete recreation of AVSMs can handle this change.

7. Conclusions and Limitations

We have presented a many-light technique for volume rendering which is tailored for interactive editing of transfer functions. The many-light approach, by approximating light transport with a set of virtual lights, enables us to cache a large portion of global illumination and to achieve interactivity. The time-intensive transfer function editing was alleviated by providing an immediate update of the VLs contribution and progressively redistributing them. We also provided a quick update of cached VLs that enables real-time previews with global scaling of the medium density.

As we aim at interactive visualization, we use a relatively small number of VLs. We also avoid the singularities of VPLs using regularization. Temporal coherence is improved by progressively updating the VLs positions and refreshing only their contribution. Our method works best if the transfer function is changed slowly, for example, by slightly adjusting the curve in a GUI. Strong changes, for example, switching to a completely different transfer function, cause visi-

ble flickering until the VLs are distributed and the AVSMs are redrawn. Assuming 1 ms per AVSM and 20 updates per frame, a rendering with 100 VLs converge after 900 ms (Fig. 6). The frame rate can vary slightly depending on the number of AVSMs to update. Repositioning the light source does not affect the AVSMs and requires only the retracing of light path and the update of contribution. Drastic changes are easily handled by our progressive update method.

References

[Cha60] CHANDRASEKHAR S.: *Radiative transfer*. Dover, 1960. 2

[DKH*13] DACHSBACHER C., KRIVÁNEK J., HASAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. In *Eurographics - State of the Art Reports* (2013). 2, 3

[ENSD12] ENGELHARDT T., NOVÁK J., SCHMIDT T.-W., DACHSBACHER C.: Approximate bias compensation for rendering scenes with heterogeneous participating media. *Pacific Graphics 31*, 4 (2012), 61–64. 2, 3, 7

[HG41] HENYEY L. G., GREENSTEIN J. L.: Diffuse radiation in the galaxy. *Astrophysical Journal 93* (1941), 70–83. 7

[HKWB09] HAŠAN M., KŘIVÁNEK J., WALTER B., BALA K.: Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 28*, 5 (2009), 143:1–143:6. 2, 4

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH* (1998), pp. 311–320. 2

[Jen96] JENSEN H. W.: Global illumination using photon maps. In *Proc. of Eurographics Workshop* (1996), pp. 21–30. 2

[JKRY12] JÖNSSON D., KRONANDER J., ROPINSKI T., YNNERMAN A.: Historygrams: Enabling interactive global illumination in direct volume rendering using photon mapping. *IEEE Trans. on Visualization and Computer Graphics 18*, 12 (2012), 2364–2371. 2, 3

[JSYR12] JÖNSSON D., SUNDÉN E., YNNERMAN A., ROPINSKI T.: Interactive volume rendering with volumetric illumination. In *Eurographics STAR program* (2012). 2

[KD13a] KAPLANYAN A. S., DACHSBACHER C.: Adaptive progressive photon mapping. *ACM Trans. on Graphics 32*, 2 (2013). 3

[KD13b] KAPLANYAN A. S., DACHSBACHER C.: Path space regularization for holistic and robust light transport. *Computer Graphics Forum (Proc. of Eurographics) 32*, 2 (2013). 2

[Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH* (1997), pp. 49–56. 2

[KJL*12] KRONANDER J., JÖNSSON D., LÖW J., LJUNG P., YNNERMAN A., UNGER J.: Efficient visibility encoding for dynamic illumination in direct volume rendering. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 18*, 3 (2012), 447–462. 3

[KPB12] KROES T., POST F. H., BOTHA C. P.: Exposure render: An interactive photo-realistic volume rendering framework. *PLoS ONE 7*, 7 (2012). 3

[KPH*03] KNISS J., PREMOZE S., HANSEN C., SHIRLEY P., MCPHERSON A.: A model for volume lighting and modeling. *IEEE Trans. on Visualization and Computer Graphics 9*, 2 (2003), 150–162. 3

[LR10] LINDEMANN F., ROPINSKI T.: Advanced light material interaction for direct volume rendering. In *IEEE/EG International Symposium on Volume Graphics* (2010), pp. 101–108. 3

[LV00] LOKOVIC T., VEACH E.: Deep shadow maps. In *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (2000), pp. 385–392. 2

[NED11] NOVÁK J., ENGELHARDT T., DACHSBACHER C.: Screen-space bias compensation for interactive high-quality global illumination with virtual point lights. In *Symposium on Interactive 3D Graphics and Games* (2011), pp. 119–124. 2

[NNDJ12a] NOVÁK J., NOWROUZEZAHRAI D., DACHSBACHER C., JAROSZ W.: Progressive virtual beam lights. *Computer Graphics Forum (Proc. of Eurographics) 31*, 4 (2012), 1407–1413. 2

[NNDJ12b] NOVÁK J., NOWROUZEZAHRAI D., DACHSBACHER C., JAROSZ W.: Virtual ray lights for rendering scenes with participating media. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 31*, 4 (2012), 60:1–60:11. 2

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory To Implementation*. Morgan Kaufmann series in interactive 3D technology. Elsevier Science, 2004. 2

[RDGK12] RITSCHEL T., DACHSBACHER C., GROSCH T., KAUTZ J.: The state of the art in interactive global illumination. *Computer Graphics Forum 31*, 1 (2012), 160–188. 2

[RGK*08] RITSCHEL T., GROSCH T., KIM M. H., SEIDEL H.-P., DACHSBACHER C., KAUTZ J.: Imperfect shadow maps for efficient computation of indirect illumination. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 27*, 5 (2008), 129:1–129:8. 5

[RMSD*08] ROPINSKI T., MEYER-SPRADOW J., DIEPENBROCK S., MENSMANN J., HINRICHS K. H.: Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Computer Graphics Forum (Proc. of Eurographics) 27*, 2 (2008), 567–576. 3

[RSK08] RAAB M., SEIBERT D., KELLER A.: Unbiased global illumination with participating media. In *Monte Carlo and Quasi-Monte Carlo Methods 2006* (2008), pp. 591–606. 2

[Sal07] SALAMA C. R.: GPU-based Monte Carlo volume raycasting. In *Pacific Graphics* (2007), pp. 411–414. 3

[SVLL10] SALVI M., VIDIMČE K., LAURITZEN A., LEFOHN A.: Adaptive volumetric shadow maps. In *Computer Graphics Forum (Proc. of Eurographics)* (2010), pp. 1289–1296. 2, 5

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1998. AAI9837162. 2

[WMHL65] WOODCOCK E., MURPHY T., HEMMINGS P., LONGWORTH S.: Techniques used in the gem code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Proc. Appl. of Computing Methods to Reactor Problems* (1965), p. 557. 4

[ZDM13] ZHANG Y., DONG Z., MA K.-L.: Real-time volume rendering in dynamic lighting environments using precomputed photon mapping. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 19*, 8 (2013), 1317–1330. 3

[ZM13] ZHANG Y., MA K.-L.: Fast global illumination for interactive volume visualization. In *Proc. of SIGGRAPH Symposium on Interactive 3D Graphics and Games* (2013), pp. 55–62. 3