Pacific Graphics 2012
C. Bregler, P. Sander, and M. Wimmer
(Guest Editors)

Volume 31 (2012), Number 7

Approximate Bias Compensation for Rendering Scenes with Heterogeneous Participating Media

Thomas Engelhardt Jan Novák Thorsten-W. Schmidt Carsten Dachsbacher

Computer Graphics Group / Karlsruhe Institute of Technology

Figure 1: *Our approximate bias compensation can be used in complex environments to recover the energy loss due to clamping the contribution of VPLs. In the Crytek Sponza the clamped volumetric and surface illumination was rendered in 39 minutes (using 118k VPLs), while the missing energy was recovered using a two-bounce ABC in only 13 minutes.*

Abstract

In this paper we present a novel method for high-quality rendering of scenes with participating media. Our technique is based on instant radiosity, which is used to approximate indirect illumination between surfaces by gathering light from a set of virtual point lights (VPLs). It has been shown that this principle can be applied to participating media as well, so that the combined single scattering contribution of VPLs within the medium yields full multiple scattering. As in the surface case, VPL methods for participating media are prone to singularities, which appear as bright "splotches" in the image. These artifacts are usually countered by clamping the VPLs' contribution, but this leads to energy loss within the short-distance light transport. Bias compensation recovers the missing energy, but previous approaches are prohibitively costly. We investigate VPL-based methods for rendering scenes with participating media, and propose a novel and efficient approximate bias compensation technique. We evaluate our technique using various test scenes, showing it to be visually indistinguishable from ground truth.

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

1. Introduction

Participating media cause a wide variety of scattering effects that contribute immensely to the realism of a scene. In landscape sceneries the appearance of haze, clouds, or smoke is caused by light that is scattered multiple times due to the interaction with small particles in the air. Unfortunately, simulating such light transport is very costly, because scattering occurs everywhere, and continues until light is absorbed or leaves the medium. A common simplification is to assume that the medium has only single scattering characteristics. This already creates impressive volumetric effects, such as light shafts piercing through tree canopies, but the simplification is only valid for participating media with low albedo (i.e. relatively high absorption). In contrast, multiple scattering is crucial for the appearance of highly scattering media.

In the context of surface shading, instant radiosity (IR) [Kel97] has proven to be very efficient for computing smooth indirect illumination. IR combines direct illumination from primary light sources and direct illumination from (secondary) virtual point lights (VPLs) to compute

© 2013 The Author(s)
Computer Graphics Forum © 2013 The Eurographics Association and Blackwell Publishing Ltd. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

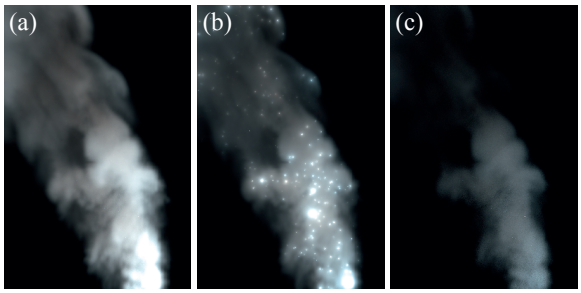

Figure 2: *(a) Unbiased rendering of heterogeneous smoke. (b) Rendering multiple scattering with VPLs causes characteristic splotches. These can be avoided by clamping, however, this removes a significant amount of energy (c).*

full surface global illumination. The same efficiency can be achieved for participating media, where the *combined single scattering* from primary light sources and VPLs yields multiple scattering [RSK08]. This has many advantages: (1) VPL methods hardly require any pre-processing, since distributing VPLs by the means of random walks is a matter of milliseconds, even for thousands of VPLs. (2) Rendering with VPLs is highly scalable, and covers the range from interactive previews [LSK*07, RGK*08] to high quality rendering with hundreds of thousands [WABG06] to millions [DKH*10, KFB10] of VPLs. (3) VPLs can rely on conventional shadow mapping to quickly resolve visibility, without the need for additional complicated structures (e.g. photon maps). These make them the ideal choice for GPU-based rendering algorithms [HPB07].

Despite these striking advantages, VPLs suffer from a very distinctive problem. The contribution of VPLs to nearby shading points is extraordinarily high, causing high-intensity peaks in the image (Fig. 2(b)). Therefore, the contribution of a VPL is normally clamped, and although this removes splotches, it suppresses the short-distance light transport and thereby darkens the image artificially. As Fig. 2(c) demonstrates, this energy loss is significant and non-uniform, making recovery non-trivial. Raab et al. [RSK08] addressed this problem by extending bias compensation (BC) [KK04] to participating media. Unfortunately, their method is prohibitively costly, ruining the elegance of IR.

In this paper we study BC in the presence of participating media and present the *approximate bias compensation* (ABC), which, unlike the original BC, increases the entire rendering cost by only a fraction of the IR rendering time. Additionally, our technique requires access to only *one* VPL at a time, being trivially progressive and GPU-friendly.

2. Previous Work

In recent years, rendering of participating media has been studied extensively. Excellent and comprehensive overviews are already given by Pegoraro [Peg09], Cerezo et al. [CPCP*05], and Premože et al. [PARN04]; we will only focus on the most relevant previous work here.

Early methods solving the radiative transfer in media [Cha60] were based on tracing rays [KVH84, RT87], and later on tracing paths [PKK00]. These techniques are general, compute unbiased solutions, and can easily handle arbitrary phase functions and heterogeneous media. In certain situations, the high dimensionality of the path space can be overcome by diffusion theory [Sta95, JMLH01], but these models imply approximations that are not suitable for media such as smoke or clouds. Recent approaches attacked the problem via caching, followed by a density estimation [JC98, JDZJ08, JZJ08, KZ11], or line space gathering [SZLG10], but their computation cost forbids interactive application. Restricting the radiative transfer to single scattering only enables real-time rendering using analytic integration [SRNN05, PSP09], sub-sampling [ED10], or hierarchical evaluation [BCRK*10], but is inadequate for scenes dominated by indirect (volumetric) illumination.

Instant radiosity [Kel97] approximates global illumination using direct lighting from a set of VPLs created using random walks. It is limited to diffuse or slightly glossy materials [KFB10] as typically no more than several hundreds or thousands of VPLs can be used at reasonable cost, the major bottleneck being visibility determination between VPLs and shading points. For this, ray casting is often used in offline algorithms, while interactive applications resort to shadow maps benefiting from their fast queries. The costly creation of shadow maps has been addressed by using imperfect shadow maps [RGK*08], or minimized via incremental updates [LSK*07]. Our method also uses VPLs and thus directly benefits from these techniques. Walter et al. [WABG06] introduced light cuts, which made rendering with hundreds of thousands of VPLs feasible and also supports volumetric effects. However, their method is limited to homogeneous participating media only and does not compensate for the bias due to clamping. Bias compensation for IR with surfaces has first been described by Kollig and Keller [KK04] and later extended to participating media by Raab et al. [RSK08] (as outlined in Sect. 3.4). In recent work Davidovic et al. [DKH*10] and Novák et al. [NED11] introduced fast approximations based on placement of new light sources, and bias compensation in screen-space, but both approaches are limited to surfaces only.

3. Instant Radiosity with Participating Media

As mentioned in the introduction, instant radiosity can be used for rendering participating media with multiple scattering. Although this process is in principle outlined by Raab et al. [RSK08] and conceptually similar to bidirectional Monte Carlo methods, we are not aware of any detailed, self-contained description of VPL-based rendering for this scenario. To facilitate the discussion of our approximate bias compensation (ABC) technique later on, we summarize the basics of light transport in participating media and the VPL generation for instant radiosity in the following.

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

3.1. Light Transport in Participating Media

The volumetric rendering equation defines radiance arriving at a point $\mathbf{x}$ in direction ω as a sum of light emitted/scattered in the volume, and light emitted/reflected from surfaces:

$$L_i(\mathbf{x},\omega) = \underbrace{\int_0^{\|\mathbf{z}-\mathbf{x}\|} \tau(\mathbf{x},\mathbf{y}) L_o(\mathbf{y},\omega) ds}_{\text{volume contribution}} + \underbrace{\tau(\mathbf{x},\mathbf{z}) L_o(\mathbf{z},\omega)}_{\text{surface reflection}}, \quad (1)$$

where $\mathbf{y}=\mathbf{x}-s\omega$, $\mathbf{z}$ is an intersection of the ray $\mathbf{x}-s\omega$ with the closest surface, and $s>0$; see Fig. 3 for illustration.

The transmittance $\tau(\mathbf{x},\mathbf{y}) = e^{-\int_0^{\|\mathbf{x}-\mathbf{y}\|} \sigma_t(x-t\omega)dt}$ accounts for out-scattering and absorption along a ray segment; the extinction coefficient $\sigma_t(\mathbf{x})$ is the sum of the coefficients for absorption, $\sigma_a(\mathbf{x})$, and scattering, $\sigma_s(\mathbf{x})$. Radiance in-scattered along the ray, as well as reflected from the surface, can be expressed in a generalized light transport equation:

$$L_o(\mathbf{x},\omega) = L_e(\mathbf{x},\omega) + \int_\Omega \hat{f}(\mathbf{x},\omega,\omega') D_{\mathbf{x}}(\omega') L_i(\mathbf{x},\omega') d\omega'.$$

where ω' is the direction along which light arrives at $\mathbf{x}$, and $\hat{f}(\mathbf{x},\omega,\omega')$ defines scattering using the phase function f_p, or the BRDF f_r, as:

$$\hat{f}(\mathbf{x},\omega,\omega') = \begin{cases} f_p(\mathbf{x},\omega,\omega')\sigma_s(\mathbf{x}) & \text{if } \mathbf{x}\in\mathcal{V} \\ f_r(\mathbf{x},\omega,-\omega') & \text{if } \mathbf{x}\in\mathcal{S} \end{cases}$$

The generalized distribution function describes the probability of light travelling in direction ω' being scattered into direction ω [PH10]. Recursively expanding Eq. (1) yields the incident light L_i^n at $\mathbf{x}$ for all paths of length up to n [PKK00]:

$$L_i^n(\mathbf{x},\omega) = \sum_{k=1}^{n} \underbrace{\int..\int}_{\text{k-times}} L_e(\mathbf{x}_k,\omega_k)\mathbf{T}_{1,k}\tau(\mathbf{x}_1,\mathbf{x}) d\mu(\mathbf{x}_1)..d\mu(\mathbf{x}_k),$$

where ω_k is the direction from $\mathbf{x}_k$ to $\mathbf{x}_{k-1}$, and L_i^∞ represents the full light transport in the scene. $\mathbf{T}_{j,k}$ is the path throughput defined using the *generalized geometry term* $\hat{G}(\mathbf{x},\mathbf{y})$ and *visibility term* $\hat{V}(\mathbf{x},\mathbf{y})$ as follows:

$$\begin{aligned} \mathbf{T}_{j,k} &= \prod_{i=j}^{k-1} \hat{f}(\mathbf{x}_i,\omega_i,\omega_{i+1})\hat{G}(\mathbf{x}_i,\mathbf{x}_{i+1})\hat{V}(\mathbf{x}_i,\mathbf{x}_{i+1}) \\ \hat{G}(\mathbf{x},\mathbf{y}) &= D_{\mathbf{x}}(\mathbf{y})D_{\mathbf{y}}(\mathbf{x})/\|\mathbf{x}-\mathbf{y}\|^2 \\ \hat{V}(\mathbf{x},\mathbf{y}) &= \tau(\mathbf{x},\mathbf{y})V(\mathbf{x},\mathbf{y}), \end{aligned}$$

where $V(\mathbf{x},\mathbf{y})$ is the binary visibility function. $D_{\mathbf{x}}(\mathbf{y}) = 1$ and $d\mu(\mathbf{x}) = dV(\mathbf{x})$ when $\mathbf{x}$ is inside the medium; otherwise, in the case when light is reflected at a surface location $\mathbf{x}$, $D_{\mathbf{x}}(\mathbf{y}) = \max(0,\mathbf{n}_{\mathbf{x}}\cdot\omega_{xy})$ and $d\mu(\mathbf{x}) = dA(\mathbf{x})$. To derive an IR-based rendering algorithm we reformulate the path integral as:

$$L_i^\infty(\mathbf{x}_0,\omega_1) = L_i^2(\mathbf{x}_0,\omega_1) + \sum_{k=3}^{\infty} \underbrace{\int..\int L_e(\mathbf{x}_k,\omega_k)\mathbf{T}_{2,k}}_{\text{represented as VPLs}} \mathbf{T}_{1,2}\tau(\mathbf{x}_1,\mathbf{x}_0) d\mu(\mathbf{x}_1)..d\mu(\mathbf{x}_k). \quad (2)$$

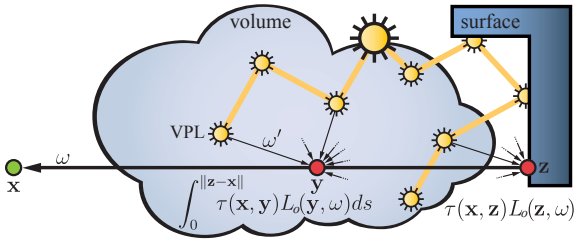

Figure 3: *Light transport in participating media: radiance arriving at a point equals the sum of the integrated, in-scattered radiance and the radiance reflected off the surface. Indirect illumination and multiple scattering is computed by gathering from VPLs.*

Keller's [Kel97] key observation is that the *path space* can be split into short paths of length less or equal to 2, and long paths. The short paths account for direct emission, direct illumination, and single scattering and can be efficiently computed e.g. using shadow mapping and ray marching. The important observation is that we can split *long paths* as well: the path prefix $(\mathbf{x}_k,\mathbf{x}_{k-1},..,\mathbf{x}_2)$ can be precomputed using a random walk and stored as VPLs. Connecting the eye path segment $(\mathbf{x}_1,\mathbf{x}_0)$ to the VPLs then approximates indirect illumination and multiple scattering (see Fig. 3).

3.2. Generating VPLs for Instant Radiosity with Participating Media

Using a random walk, we incrementally construct N light paths (yielding M VPLs) by importance sampling the Monte Carlo estimator of the path integral. Note that in contrast to IR for surfaces, we not only sample directions at light-surface interactions, but also distances along rays, as light might be scattered in the medium. We can derive the Monte Carlo estimator for L_i^∞ by adapting it from Veach [Vea98]:

$$L_i^\infty(\mathbf{x}_0,\omega_1) \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{\infty} \frac{L_e(\mathbf{x}_k,\omega_k)}{p(\mathbf{x}_k)} \prod_{j=1}^{k} \frac{\tau(\mathbf{x}_j,\mathbf{x}_{j-1})\hat{C}_j}{p(\omega_j)p(t_j)},$$

$$\hat{C}_j = \begin{cases} 1 & \text{if } j=k \\ f_p(\mathbf{x}_j,\omega_j,\omega_{j+1})\sigma_s(\mathbf{x}_j) & \text{if } \mathbf{x}_j\in\mathcal{V} \\ f_r(\mathbf{x}_j,\omega_j,-\omega_{j+1})(\omega_j\cdot\mathbf{n}_{\mathbf{x}_j}) & \text{if } \mathbf{x}_j\in\mathcal{S} \end{cases}$$

where $p(.)$ is the probability density function (PDF) for sampling a position $\mathbf{x}_k$ on a light source, a direction ω, or distance t along a ray. In contrast to the original IR, which stores virtual point lights, we store path vertices, including the associated surface normal and BRDF (or phase function), incident radiance, and incident direction. This enables using non-diffuse materials and anisotropic phase functions. Despite this little discrepancy, we will nonetheless use the term VPL instead of saying "path vertex". The random walk begins by creating a starting path vertex on a primary light source and proceeds as follows:

1. Sample the next scattering event with direction ω_j and distance t_j along the ray $\mathbf{r}(t) = \mathbf{x}_j + t\omega_j$ to determine the

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

next path vertex $\mathbf{x}_{j-1}$. For homogeneous media analytic PDFs exist [LW96], otherwise we use Woodcock tracking [WMHT65]. Depending on the distance, $\mathbf{x}_{j-1}$ resides either in the volume or on the closest surface.
2. Create a VPL at $\mathbf{x}_{j-1}$ storing its incident radiance $L_i(\mathbf{x}_{j-1},\omega_j) = L_i(\mathbf{x}_j,\omega_{j+1})\tau(\mathbf{x}_j,\mathbf{x}_{j-1})\hat{C}_j/p(\omega_j,t_j)$, using $L_e(\mathbf{x}_k,\omega_k)/p(\mathbf{x}_k)$ instead of $L_i(\mathbf{x}_j,\omega_{j+1})$ when $\mathbf{x}_j$ resides on the light source. We also store the incident direction ω_j, and, if the VPL is created in the volume, the scattering coefficient $\sigma_s(\mathbf{x}_{j-1})$ and the phase function, or the BRDF and surface normal at $\mathbf{x}_{j-1}$ (and tangent space for anisotropic BRDFs).
3. Use Russian roulette and terminate the path, or proceed with step 1 re-weighting L_i of all further VPLs [PH10].

After finishing all random walks we divide each VPL's incident radiance by the total number of generated light paths N.

3.3. Rendering Participating Media with VPLs

At surface points, we simply gather the indirect illumination by summing up the contribution of all VPLs. Inside the volume, the VPLs' contribution must be (numerically) integrated along the eye rays using a Monte Carlo estimator:

$$L_i(\mathbf{x},\omega) \approx \sum_{s=0}^{S}\sum_{k=0}^{M} \frac{\tau(\mathbf{x},\mathbf{y}_s)\hat{f}(\mathbf{y}_s,\omega,\omega_k')\hat{G}(\mathbf{y}_s,\mathbf{v}_k)\hat{V}(\mathbf{y}_s,\mathbf{v}_k)L_k(\mathbf{v}_k,\omega_k')}{S\,p(\mathbf{y}_s)}.$$

Here, S denotes the number of samples $\mathbf{y}_s$ along the eye ray $\mathbf{x}+t\omega$, M the number of VPLs, $\mathbf{v}_k$ the position, and L_k the outgoing radiance of the k-th VPL computed by convolving its incident radiance $L_{i,k}$ with the associated scattering function $\hat{f}(\mathbf{v}_k,\omega_k',\omega_i)$. We importance sample the estimator according to the transmittance along the eye ray (i.e. either analytically or using Woodcock tracking), which significantly reduces variance. We also tried sampling according to the inverse squared distance to the VPL (present in $\hat{G}$) [KF11], but as we clamp the geometry term, this causes heavy oversampling of the bounding region, increasing the variance.

3.4. Bias Compensation

Rendering as described in Sect. 3.3 leads to unbiased solutions, but suffers from bright splotches due to a singularity in the generalized geometry term. Clamping the geometry term $\hat{G}'(\mathbf{x},\mathbf{y}) = \min(\hat{G}(\mathbf{x},\mathbf{y}),b)$ to some bound $b>0$ removes these high intensity peaks; this, however, introduces bias (see [KK04, RSK08] for a discussion how to choose b).

Kollig and Keller [KK04] recursively correct the bias for inter-surface light transport and Raab et al. [RSK08] extend this idea to participating media. Both compute a biased (clamped) solution first and then add a correction integral $L'(\mathbf{y},\omega)$ with a compensation term $\hat{B}$ to each sample $\mathbf{y}$:

$$L'(\mathbf{y},\omega) = \int_\Omega \hat{f}(\mathbf{y},\omega,\omega')\tau(\mathbf{y},\mathbf{y}')\hat{B}(\mathbf{y},\mathbf{y}')L_o(\mathbf{y}',\omega')d\omega' \quad (3)$$

$$\hat{B}(\mathbf{y},\mathbf{y}') = \frac{\max(0,\hat{G}(\mathbf{y},\mathbf{y}')-b)}{\hat{G}(\mathbf{y},\mathbf{y}')}.$$

The correction works as follows: from the shading location $\mathbf{y}$, a ray is cast into a direction chosen according to the generalized scattering function $\hat{f}(.)$ (i.e., according to the BRDF on surfaces, or to the phase function in the medium). Along this ray either scattering in the medium occurs, or the ray intersects another surface. The location of this interaction, $\mathbf{y}'$, becomes a vertex of a new path where the full illumination, including direct and (clamped) indirect lighting, must be evaluated. The recursion is automatically terminated when the compensation weight $\hat{B}(\mathbf{y},\mathbf{y}')$ becomes zero, i.e. $\mathbf{y}'$ is too far from $\mathbf{y}$ (outside the region where the clamping occurs).

4. Approximate Bias Compensation

Bias compensation according to Raab et al. [RSK08] is very costly as it requires recursive ray tracing, access to all VPLs at every compensation vertex, and in the worst case degenerates to (bidirectional) path tracing [DKH*10].

In this section we analyze the energy recovery due to bias compensation and derive several optimizations, sampling strategies, and simplifying assumptions by analyzing Eq. (3). Then we derive a more efficient, approximate bias compensation technique, that yields results indistinguishable from ground truth. We illustrate and motivate the proposed simplifications for ABC using plots of the brightness along the scanline highlighted in Fig. 4, and renderings with different settings to assess the image quality (Figs. 5, 6, and 8).

4.1. Limiting Recursion Depth

Bias compensation is a recursive process, as clamping of a VPL's contribution occurs at new path vertices as well. However, the compensation integral convolves the gathered radiance with the generalized scattering function $\hat{f}$, thus removing high frequencies. More importantly, the contribution of the compensation drops exponentially with the recursion depth due to the $\hat{f}$ term. Fig. 4 shows that one compensation step and then clamping the VPLs' contribution to the new path vertices recovers most of the energy and already mimics the behavior of the ground truth curve quite well. It can also be seen that two steps (and then clamping) is visually almost indistinguishable from the ground truth solution.

4.2. Path Generation and Locally Homogeneous Media

To create new path vertices Raab et al. [RSK08] choose a random direction ω and sample a distance along it using Woodcock tracking [WMHT65]. If the new vertex happens to be outside the clamping region it will have zero contribution. This in turn leads to high variance in the compensation estimator. The variance can be reduced using more compensation vertices, however, this implies a substantial overhead, as there is no possibility in *heterogeneous* participating media to sample new vertices according to the transmittance within a given distance. We avoid this issue by introducing the assumption that the medium is *locally homogeneous* around the compensation point.

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

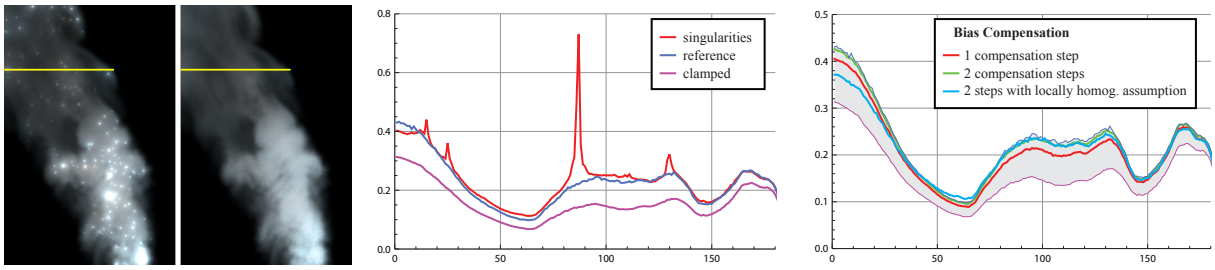

Figure 4: *Left: we use the highlighted scanline to illustrate our observations. Center: plot of the pixels' radiance when computing the image with un-clamped VPLs (exhibiting singularities, red), clamped VPLs (purple), and ground truth (full compensation, blue). Right: radiance plotted for ground truth and clamping (top and bottom boundary of the band), with 1 (red), 2 (green), or 2 locally homogeneous (blue) additional compensation steps.*

Figure 5: *Bias compensation with accurate transmittance and with locally homogeneous assumption.*

One key ingredient to our ABC is that we generate the new path vertices always inside the bounding region where clamping occurs. The radius d of the spherical region can be computed from the bound b as $d = 1/\sqrt{b}$. Assuming a *locally homogeneous* medium with extinction coefficient $\overline{\sigma}_t$, the probability density function for sampling a distance t (for the new vertex $\mathbf{y}'$) from $\mathbf{y}$ within the bounding region reads:

$$p(t) = \frac{\overline{\sigma}_t e^{-\overline{\sigma}_t t}}{1 - e^{-\overline{\sigma}_t d}}. \tag{4}$$

Using the inversion method we compute the distance as:

$$t = -\frac{\ln(1 - \xi(1 - e^{-\overline{\sigma}_t d}))}{\overline{\sigma}_t}, \tag{5}$$

with a uniform random number $\xi \in [0;1)$. For $\overline{\sigma}_t$ we use the average extinction coefficient within the bounding region, which can be efficiently obtained from a downsampled version of the medium, i.e. if stored in a 3D texture. Note that if there is a surface intersection occurring closer to $\mathbf{y}$ than t, this intersection becomes the new compensation vertex.

The assumption of local homogeneity does not compromise the results. In fact, it only affects the placement of the new vertex $\mathbf{y}'$, and the computation remains unbiased as long as we correctly evaluate the transmittance $\tau(\mathbf{y}, \mathbf{y}')$. Nevertheless, we take the assumption one step further, and also approximate the transmittance $\overline{\tau}(\mathbf{y}, \mathbf{y}') = \exp(-\overline{\sigma}_t \|\mathbf{y} - \mathbf{y}'\|)$, effectively avoiding costly evaluation (e.g. by Woodcock tracking [JNT*11]). In all our test scenes this yielded results visually indistinguishable from ground truth (see Fig. 5), although the assumption can theoretically fail at locations with strongly varying extinction. Note that these locations could be easily detected if necessary using the gradient of $\sigma_t(\mathbf{y})$.

4.3. Integration Strategies

In this section we analyze different sampling strategies for bias compensation. Each of the strategies focuses the computation on different parts of the compensation. In Fig. 6 we compare the results at roughly equal rendering time. A corresponding plot of the RMSE induced by the different strategies is shown in Fig. 7.

1-to-N: Raab et al. [RSK08] propose to connect each compensation vertex to all VPLs. This has two implications. First, because of free path sampling along the compensation ray, the vertex may be placed outside the bounding region, and compensation is entirely discarded. This approach can lead to high variance, since the compensation integral (Eq. (3)) is severely under-sampled (Fig. 6, 1-to-N). Second, each compensation step requires access to *all* VPLs and their shadow maps, making the approach GPU-unfriendly.

N-to-1: We found that noise can be significantly reduced by creating more compensation vertices, but connecting each of them to one VPL only. Therefore we process all VPLs sequentially, connecting each to one compensation vertex in turn. Unfortunately, this increases the number of rays that need to be traced, thereby increasing the run-time. However, as shown in the equal-time comparison in Fig. 6, the N-to-1 approach still exhibits lower variance than [RSK08].

1-to-1: By always creating only a single compensation vertex connected to the VPL that is currently evaluated along the primary ray, we significantly reduce the cost of the compensation, and thus we can take more samples along the eye ray and further reduce the noise due to transmittance. This approach is also GPU-friendly, because VPLs can be processed independently: we can compute the contribution of a single VPL to all pixels (including BC), accumulating the results progressively over time. Consequently, we need to store only a single shadow map in memory at a time.

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

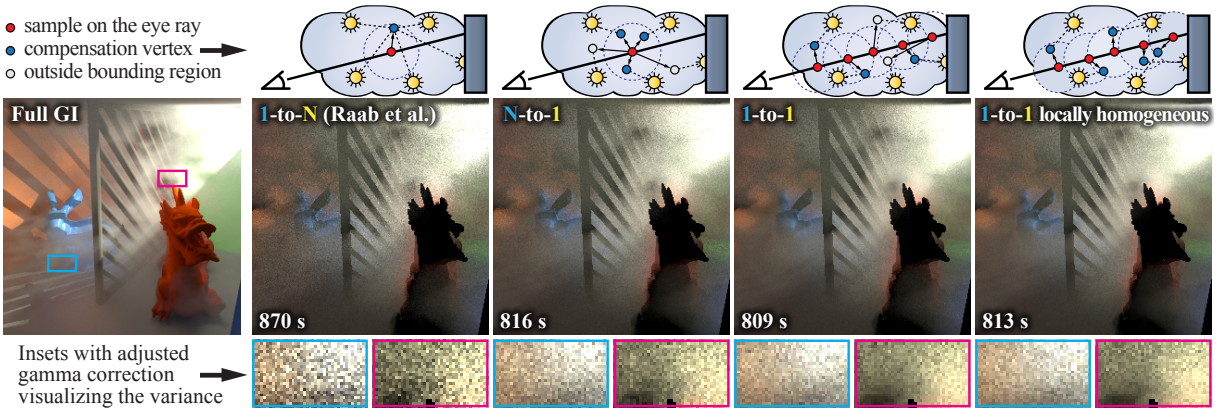

Figure 6: *Different compensation strategies: Raab et al. connect all VPLs to a single compensation vertex (1-to-N). Lower variance can be achieved by generating more vertices while connecting each of them to a single VPL (N-to-1). A similar, but GPU-friendly approach is to generate only one vertex connected to a single VPL (1-to-1). By assuming a locally homogeneous medium we avoid expensive evaluation of the transmittance and ensure that vertices are always created within the bounding region, thus minimizing divergent code paths on the GPU. Rendering time is controlled by adjusting the number of samples along the eye ray: 3, 1, 115, and 78 for 1-to-N, N-to-1, 1-to-1, and 1-to-1 locally homogeneous, respectively. (Please note that 1-to-1 discards roughly half of the compensation vertices compared to the loc. hom. approach.)*

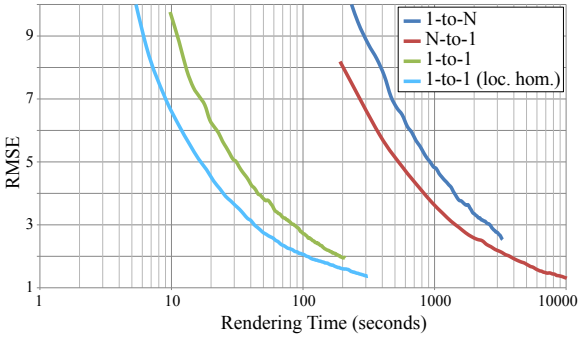

Figure 7: *RMSE plot for different compensation strategies (see Sec. 4.3) used in Figure 6, computed against a reference solution (converged 1-to-N strategy proposed in [RSK08]). The 1-to-1 strategy clearly outperforms other strategies.*

1-to-1 locally homogeneous: The 1-to-1 approach discards many compensation rays, because of free path sampling. Ideally we would like to create all compensation vertices within the bounding region. This can be assured using our locally homogeneous assumption. Additionally, it ensures that parallel execution paths do not become divergent, which is favorable for GPU implementation.

4.4. Omitting Visibility and Local Visibility

Computing visibility between a point $\mathbf{y}$ that requires compensation, and a new path vertex $\mathbf{y}'$ is crucial for overall performance. Obviously, the path $\mathbf{y}$ to $\mathbf{y}'$ can only be occluded when $\mathbf{y}$ is close to a surface, but not in "free space".

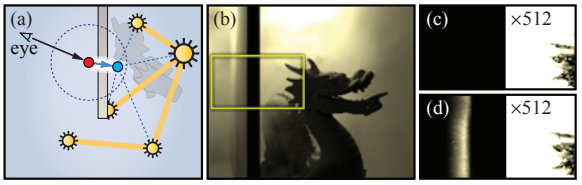

Figure 8: *Compared to the accurate computation (c), ignoring visibility to new path vertices (Sect. 4) causes artifacts (d) only revealed with tremendous scaling (×512).*

Fig. 8a illustrates the case when not computing the visibility causes artifacts. To asses how often these artifacts appear, and their influence on the resulting images, we set up a series of experiments. Interestingly, it was not easy to produce visible artifacts at all. This can be explained by considering the circumstances that have to coincide to cause them (Fig. 8a): (1) $\mathbf{y}$ must be close to a thin opaque object, and (2) the medium must not be too dense otherwise sampling a distance through the opaque object is unlikely. Note that if $\mathbf{y}$ and $\mathbf{y}'$ are further apart (yet within radius d), the quadratic decrease of the compensation term with the distance also suppresses light bleeding. Fig. 8b shows one of our test scenes; artifacts become visible only after scaling the brightness by at least 2 orders of magnitude. Note that a somewhat similar assumption (ignoring visibility on short distances) has also been used for global illumination on surfaces [AFOH05, DKH*10].

5. Implementation Details

We integrated our approximate bias compensation technique into a custom offline renderer to evaluate our assumptions

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

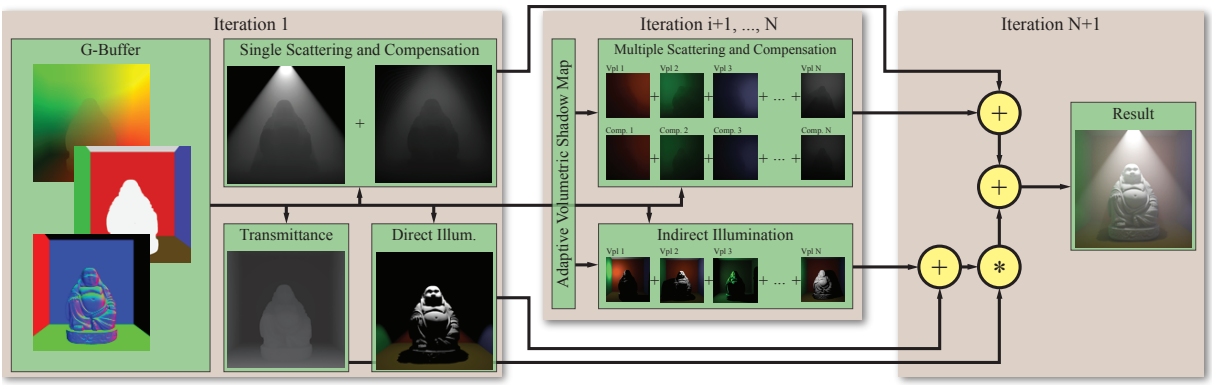

Figure 9: *Data flow in our progressive GPU renderer. In the first iteration we compute the geometry buffer, single scattering with bias compensation from primary light sources, transmittance, and direct illumination. In each subsequent iteration we compute multiple scattering, indirect illumination, and bias compensation due to a single VPL, and add it to the final result.*

(Figs. 4, 5, 6, 8) For further acceleration we implemented a progressive GPU renderer using Direct3D 11 (Figs. 1, 9, 11, 12, 13). Random walks for creating VPLs are always carried out on the CPU using ray tracing. For acceleration, we use a *k*D-tree built with the surface area heuristic.

The CPU implementation of our method is shown as pseudo code in Appendix A. In this section, we restrict ourselves to the peculiarities of the GPU implementation, which is outlined in Fig. 9. We split the computations into evaluating contributions from primary light sources and from secondary light sources, i.e. VPLs. First we render a geometry buffer filled with all relevant information such as BRDFs, positions, and normals. Afterwards we evaluate single scattering and direct illumination with visibility computed using shadow maps (with resolutions of 512^2 up to 4096^2). Transmittance towards the light sources is evaluated analytically in case of homogeneous media and numerically in heterogeneous media using ray marching (the offline renderer uses slower but unbiased Woodcock tracking).

VPL Lighting. After the contributions from primary light sources have been computed, our renderer iterates over all VPLs and accumulates their contribution, one per iteration. Apart from conventional shadow maps, we also compute a variant of adaptive volumetric shadow maps [SVLL10]. Since lookups into regular AVSMs involve a costly search for nearby transmittance samples, we resample the AVSM at fixed intervals. Looking up the transmittance then boils down to a single access to an array of fixed size. Finally, we accumulate the contribution of a VPL to the surfaces and the volume. The latter we evaluate using adaptive ray marching, i.e. the number of samples depends on the ray segment that intersects the volume.

Bias compensation. We split the compensation integral into two terms: one for compensating from the primary light sources, and the second gathering compensation energy from VPLs. This allows us to evaluate compensation from the primary lights directly in the single scattering shader, which is executed only once in the first iteration. For that we generate a buffer that contains a set of random directions and additional random numbers used to sample a distance along the compensation ray. Our assumption of locally homogeneous medium and neglecting visibility allow us to sample the new path vertices *always* within the bounding region. This, in contrast to the original BC [KK04], avoids branching and divergent execution paths, substantially accelerating the GPU implementation. Bias compensation gathering illumination from VPLs is handled in a similar spirit: for each VPL and each sample point along the ray we create a single vertex within the bounding region and compute its contribution to the correction term.

6. Results

We evaluated our method using several test scenes with homogeneous and heterogeneous participating media. All timings were recorded using an Intel Core i7 6-core system with 3.2GHz and a GeForce GTX 580 GPU.

Our algorithm can be used for rendering moderately com-

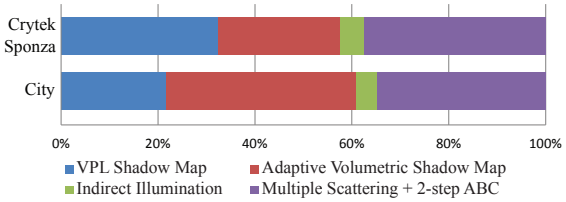

Figure 10: *Shading time per VPL for the Crytek Sponza and City scenes: most of the shading time is used for constructing the shadow maps (about 60%), while indirect (surface) illumination is relatively fast, and multiple scattering with 2-step ABC requires only 35 – 38% of the entire shading cost.*

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

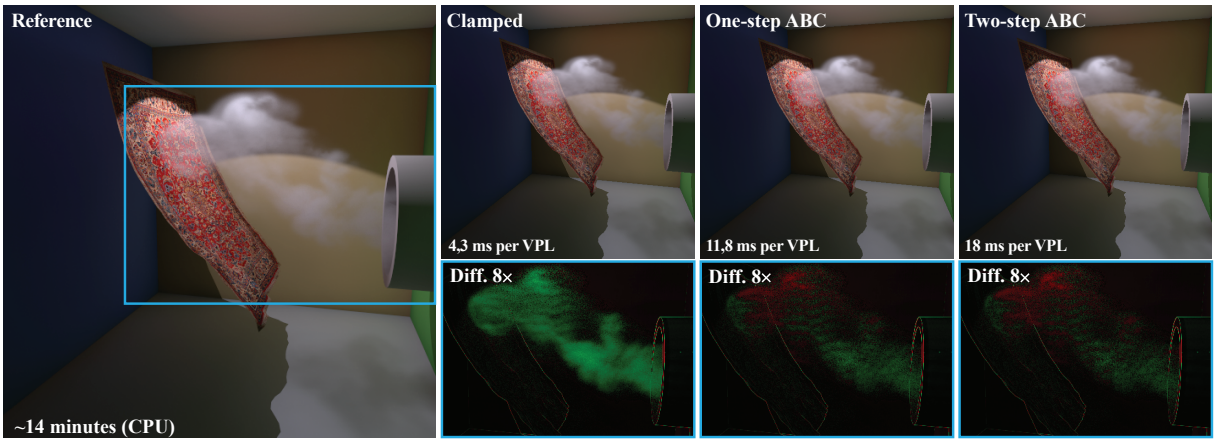

Figure 11: *Room with a heterogeneous smoke ($\sigma_s = 0.9$, $\sigma_a = 0.001$) rendered with 6800 VPLs. Clamping removes a remarkable amount of energy, which is almost completely recovered using just one ABC step. The insets visualize lost energy (green), and overcompensation (red). Differences on the edges are due to different sampling of primary visibility on the CPU and GPU.*

Figure 12: *Equal-time comparison for volumetric illumination. Left: reference CPU solution with full bias compensation. Middle: beam radiance estimate (BRE) [JZJ08] with 1 million volume photons. Right: GPU-accelerated ABC with 7887 VPLs and 2 approximate compensation steps.*

Figure 13: *The Buddha in a homogeneous medium ($\sigma_s = 0.075$, $\sigma_a = 0.001$) with varying g parameter of the HG phase function. From left to right, images are rendered with backward, isotropic, and forward scattering medium.*

plex scenes with image-based lighting, such as those in Fig. 1. Both the Crytek Sponza scene (262k triangles, 118k VPLs) and the City scene (823k triangles, 108k VPLs) were rendered with a 2-step ABC. The shading cost depends on the geometric complexity of the scene and the resolution of the 3D texture storing the heterogeneous participating medium. A detailed analysis of the per-VPL GPU shading cost is shown in Fig. 10.

Fig. 11 shows a visual comparison of our approximate bias compensation algorithm to a reference image computed with full bias compensation [KK04] in the participating medium and 6800 VPLs in the entire scene. As can be seen, the approximate bias compensation recovers most of the lost energy already after the first compensation step.

In Fig. 12 we compare our algorithm to photon mapping using the beam radiance estimate (BRE) [JZJ08] for images with 1024^2 pixels. In contrast to the BRE, IR with our compensation technique can be trivially parallelized, thus we used the GPU implementation in the comparison. The indirect illumination is represented with 7887 VPLs for our technique and 1 million of volumetric photons for the BRE. Shooting photons and building the search data structure took 25 seconds, rendering using beam queries additional 110 seconds (135 seconds in total). The rendering time of our VPL technique with a 2-step ABC is 125 seconds. Photon mapping still shows the typical artifacts that arise from an insufficient number of photons in the volume, while ABC is nearly indistinguishable from the reference solution, computed using the method of Raab et al. [RSK08] in 31 hours.

As shown in Fig. 13, our algorithm supports anisotropic media. We rendered the scene with 4320 VPLs and a 2-step ABC varying the g parameter of the Henyey-Greenstein phase function. The average shading time per VPL is 16 ms.

7. Discussion and Future Work

In this section we describe findings from experimenting with our method, which we believe are important to assess its strengths and limitations, and use in complex scenes.

IR and Participating Media. Similar to IR for surfaces we *sub-sample the path space*. The result images are typically

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

close to ground truth, because single scattering and transmittance are computed at high precision using ray marching, which preserves crisp features. The quality of the "global" light transport heavily depends on the number of VPLs. We observe that more VPLs are required for dense and heterogeneous than for thin or homogeneous media. For surfaces, the link between the scene geometry and materials and the number of required VPLs has been extensively studied by Křivánek et al. [KFB10]. Deriving similar dependency on the parameters of the medium is an interesting future work.

Complex Scenes. Scenes with large extent benefit from bidirectional VPL generation [SIP06] to create enough VPLs that actually contribute to the image. In our implementation we use a variant of Georgiev and Slusallek's [GS10] method: we create a larger number of paths and keep only those VPLs that contribute the most to the image. This is evaluated by simply considering whether they are close to the camera. This simple approach is surprisingly well-suited for scenes with participating media, such as those in Fig. 1.

Phase Functions. Our method supports *anisotropic phase functions*, but strong backward or forward scattering causes problems similar to glossy materials with IR. This is because sub-sampling the path space assumes smooth illumination, which is only valid for isotropic and moderately anisotropic scattering. To achieve artifact-free renderings of highly anisotropic media a large number of VPLs is required.

Animated Scenes. Temporal changes to primary light sources or scene geometry inherently change the distribution of VPLs in each frame. This becomes noticeable as flickering if an insufficient number of VPLs is used. Since our method directly depends on the VPL distribution, this effect is carried over. To reduce the temporal artifacts, a sufficient number of VPLs is necessary (several thousand for small scenes). To further improve temporal coherence, the VPLs can be distributed using the same sequence of random numbers in every frame.

8. Conclusions

We presented a novel method for rendering global illumination including multiple scattering in heterogeneous media, which is based on instant radiosity, thus requiring no precomputation. Key to our method is the approximate bias compensation technique that enables rendering images close to ground truth. While the visual impact of all our approximations is indistinguishable from the original BC, our technique is more efficient and amenable to GPU acceleration.

Acknowledgements

We thank Christoph Weber for proof-reading drafts of the paper and the reviewers for helpful comments. Carsten Dachsbacher acknowledges funding from the Intel Visual Computing Institute, Saarbrücken, Germany.

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.

Appendix A: ABC Pseudo Code

The following pseudo code outlines the computation of compensation energy at an arbitrary point. For brevity we do not include surface compensation, which is thoroughly described in [KK04]. We focus on the new N-to-1 and 1-to-1 strategies.

```
Color compensate(Point p, VPL vpl, int rec) {

  if(isSurface(p))
    return compensateSurface(p,vpl,rec);

  Direction w = samplePhaseFunction(p);
  Surface   s = infinity;
  Distance  d = 0;
  Ray     ray = Ray(p,w);

  // Limiting recursion depth (Section 4.1)
  if(rec > 3)
    return 0;

  // Ignore Visibility (Section 4.4)
  if(!ignoreVisibility)
    s = intersect(ray);

  // Local homogeneous assumption (Section 4.2)
  d       = sampleDistance(ray, isLocallyHomogeneous);
  Color T = transmittance(ray, d, isLocallyHomogeneous);

  // Phase Function
  Color P = phaseFunction(p,w);

  Point next_p = p + d * w;
  if(isOutsideBoundingRegion(next_p,p))
     return 0;

  // Compensation weight (Raab et al.)
  Real G = 1;
  if(isSurface(next_p,s))
    G = dot(NormalAt(next_p),-w);

  G /= (d*d);
  G  = max(0,G-bound) / G;

  if(1-to-1) {
     // 1-to-1 (Section 4.3)
     return G * P * T *
       (evalVplAt(next_p) +
        compensate(next_p, vpl, rec+1))
       / (pdf(d) * pdf(w));
  } else  {
    // N-to-1 (Section 4.3)
    Color C = none;
    foreach(VPL v)
      c += evalVplAt(next_p, v)
        + compensate(next_p, v, rec+1);

    return G * P * T * C /
          (pdf(d) * pdf(w) * N_vpls);
  }
}
```

References

[AFOH05] ARIKAN O., FORSYTH D. A., O'BRIEN M., HANRAHAN P.: Fast and detailed approximate global illumination by irradiance decomposition. In *ACM Transactions on Graphics (Proc. of SIGGRAPH)* (2005), pp. 1108–1114. 6

[BCRK*10] BARAN I., CHEN J., RAGAN-KELLEY J., DURAND F., LEHTINEN J.: A hierarchical volumetric shadow algorithm for single scattering. In *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* (2010), pp. 178:1–178:10. 2

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

[CPCP*05] CEREZO E., PEREZ-CAZORLA F., PUEYO X., SERON F., SILLION F.: A survey on participating media rendering techniques. *The Visual Computer 21* (2005), 303–328. 2

[DKH*10] DAVIDOVIČ T., KŘIVÁNEK J., HAŠAN M., SLUSALLEK P., BALA K.: Combining Global and Local Virtual Lights for Detailed Glossy Illumination. In *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* (2010). 2, 4, 6

[ED10] ENGELHARDT T., DACHSBACHER C.: Epipolar sampling for shadows and crepuscular rays in participating media with single scattering. In *Proc. of ACM I3D* (2010), pp. 119–125. 2

[GS10] GEORGIEV I., SLUSALLEK P.: Simple and robust iterative importance sampling of virtual point lights. In *Eurographics 2010 short papers* (2010). 9

[HPB07] HAŠAN M., PELLACINI F., BALA K.: Matrix Row-Column Sampling for the Many-Light Problem. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 26* (2007). 2

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient Simulation of Light Transport in Scences with Participating Media using Photon Maps. In *SIGGRAPH '98* (1998), pp. 311–320. 2

[JDZJ08] JAROSZ W., DONNER C., ZWICKER M., JENSEN H. W.: Radiance caching for participating media. *ACM Transactions on Graphics 27*, 1 (2008), 7:1–7:11. 2

[JMLH01] JENSEN H. W., MARSCHNER S. R., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *SIGGRAPH '01* (2001), pp. 511–518. 2

[JNT*11] JAROSZ W., NOWROUZEZAHRAI D., THOMAS R., SLOAN P.-P., ZWICKER M.: Progressive Photon Beams. *ACM Transactions on Graphics 30*, 6 (2011), 181:1–181:12. 5

[JZJ08] JAROSZ W., ZWICKER M., JENSEN H. W.: The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum 27*, 2 (2008), 557–566. 2, 8

[Kel97] KELLER A.: Instant radiosity. In *SIGGRAPH '97* (1997), pp. 49–56. 1, 2, 3

[KF11] KULLA C., FAJARDO M.: Importance Sampling of Area Lights in Participating Media. In *ACM SIGGRAPH 2011 Talks* (2011), SIGGRAPH '11, pp. 55:1–55:1. 4

[KFB10] KŘIVÁNEK J., FERWERDA J. A., BALA K.: Effects of global illumination approximations on material appearance. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 29*, 4 (2010), 1–10. 2, 9

[KK04] KOLLIG T., KELLER A.: Illumination in the presence of weak singularities. *Monte Carlo and Quasi-Monte Carlo methods* (2004), 245–257. 2, 4, 7, 8, 9

[KVH84] KAJIYA J. T., VON HERZEN B. P.: Ray tracing volume densities. *Computer Graphics (Proc. of SIGGRAPH) 18*, 3 (1984), 165–174. 2

[KZ11] KNAUS C., ZWICKER M.: Progressive Photon Mapping: A Probabilistic Approach. *ACM Transactions on Graphics 30* (May 2011), 25:1–25:13. 2

[LSK*07] LAINE S., SARANSAARI H., KONTKANEN J., LEHTINEN J., AILA T.: Incremental instant radiosity for real-time indirect illumination. In *Proc. of Eurographics Symposium on Rendering* (2007), pp. 277–286. 2

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Proc. of the Eurographics Workshop on Rendering* (1996), pp. 91–100. 4

[NED11] NOVÁK J., ENGELHARDT T., DACHSBACHER C.: Screen-space bias compensation for interactive high-quality global illumination with virtual point lights. In *Proc. of ACM I3D* (2011), pp. 119–124. 2

[PARN04] PREMOŽE S., ASHIKHMON M., RAMAMOORTHI R., NAYAR S.: Practical rendering of multiple scattering effects in participating media. In *Proc. of Eurographics Symposium on Rendering* (2004). 2

[Peg09] PEGORARO V.: *Efficient Physically-Based Simulation of Light Transport in Participating Media*. PhD thesis, School of Computing, University of Utah, 2009. 2

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers, 2010. 3, 4

[PKK00] PAULY M., KOLLIG T., KELLER A.: Metropolis light transport for participating media. In *Proc. of the Eurographics Workshop on Rendering* (2000), pp. 11–22. 2, 3

[PSP09] PEGORARO V., SCHOTT M., PARKER S. G.: An analytical approach to single scattering for anisotropic media and light distributions. In *Proc. of Graphics Interface* (2009), pp. 71–77. 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. In *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* (2008), pp. 129:1–129:8. 2

[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, 4, 5, 6, 8

[RT87] RUSHMEIER H. E., TORRANCE K. E.: The zonal method for calculating light intensities in the presence of a participating medium. *Computer Graphics (Proc. of SIGGRAPH) 21*, 4 (1987), 293–302. 2

[SIP06] SEGOVIA B., IEHL J.-C., Pï£¡ROCHE B.: Bidirectional Instant Radiosity. In *Proc. of the Eurographics Workshop on Rendering* (2006). 9

[SRNN05] SUN B., RAMAMOORTHI R., NARASIMHAN S. G., NAYAR S. K.: A practical analytic single scattering model for real time rendering. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 24*, 3 (jul 2005), 1040–1049. 2

[Sta95] STAM J.: Multiple scattering as a diffusion process. In *Proc. of the Eurographics Workshop on Rendering* (1995), pp. 41–50. 2

[SVLL10] SALVI M., VIDIMČE K., LAURITZEN A., LEFOHN A.: Adaptive volumetric shadow maps. In *Eurographics Symposium on Rendering* (June 2010), pp. 1289–1296. 7

[SZLG10] SUN X., ZHOU K., LIN S., GUO B.: Line space gathering for single scattering in large scenes. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 29*, 4 (July 2010), 54:1–54:8. 2

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1998. 3

[WABG06] WALTER B., ARBREE A., BALA K., GREENBERG D. P.: Multidimensional lightcuts. *ACM Transactions on Graphics 25*, 3 (July 2006), 1081–1088. 2

[WMHT65] WOODCOCK E., MURPHY T., HEMMINGS P., T.C. L.: Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. In *Applications of Computing Methods to Reactor Problems* (1965), Argonne National Laboratory, pp. 557–579. 4

© 2013 The Author(s)
© 2013 The Eurographics Association and Blackwell Publishing Ltd.