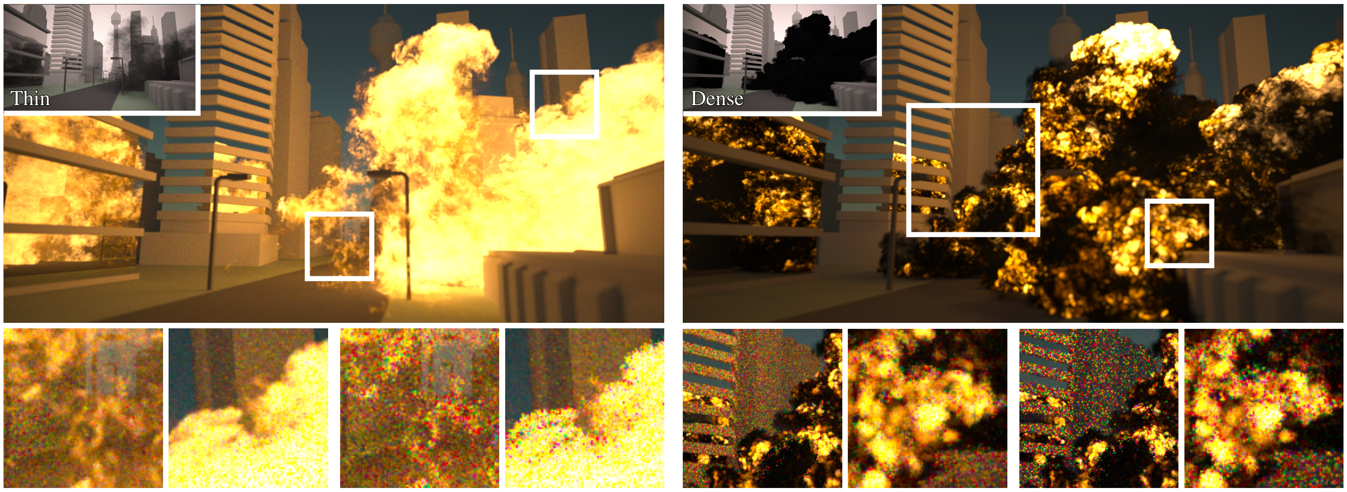
Eurographics Symposium on Rendering 2017
P. Sander and M. Zwicker
(Guest Editors)

Volume 36 (2017), Number 4

Line Integration for Rendering Heterogeneous Emissive Volumes

Florian Simon[1,2†] Johannes Hanika[1,2] Tobias Zirr[1] Carsten Dachsbacher[1]

[1]Karlsruhe Institute of Technology [2]Weta Digital Ltd.



FNEE (ours): RMSE 0.748, 48 spp NEE: RMSE 0.983, 43 spp FNEE (ours): RMSE 0.572, 196 spp NEE: RMSE 0.892, 73 spp

Figure 1: *Heterogeneous emissive volumes are a challenging form of light source. The images show an equal-time comparison of an explosion, in two versions that differ in overall density. The thin version (left, 10min) greatly benefits from a line integration estimator that accumulates emission for path segments and not only at path vertices. In addition, we propose a next event technique called forward next event estimation (FNEE) that samples the length of next event segments proportional to transmittance. This prevents long segments that do not contribute to the final image due to high extinction (right, 30min) and increases the efficiency of path tracing compared to regular next event estimation (NEE).*

Abstract

Emissive media are often challenging to render: in thin regions where only few scattering events occur the emission is poorly sampled, while sampling events for emission can be disadvantageous due to absorption in dense regions. We extend the standard path space measurement contribution to also collect emission along path segments, not only at vertices. We apply this extension to two estimators: extending paths via scattering and distance sampling, and next event estimation. In order to do so, we unify the two approaches and derive the corresponding Monte Carlo estimators to interpret next event estimation as a solid angle sampling technique. We avoid connecting paths to vertices hidden behind dense absorbing layers of smoke by also including transmittance sampling into next event estimation. We demonstrate the advantages of our line integration *approach which generates estimators with lower variance since entire segments are accounted for. Also, our novel* forward next event estimation *technique yields faster run times compared to previous next event estimation as it penetrates less deeply into dense volumes.*

CCS Concepts

•Computing methodologies → Ray tracing;

1. Introduction

Rendering realistic images typically requires the inclusion of heterogeneous participating media. This is often costly, especially with Monte Carlo rendering. In recent years, significant progress has been made in rendering scattering and absorbing media [KF12, NSJ14, KGH*14, ZRB14], however, there is considerably less work on rendering emissive heterogeneous volumes, such as fire, explosions, or flames.

In principle, one can easily account for emission by accumulating the contribution at scattering events of Monte Carlo random walks. However, for thin volumes, like flames, free path sampling accord-

[†] florian.simon@kit.edu

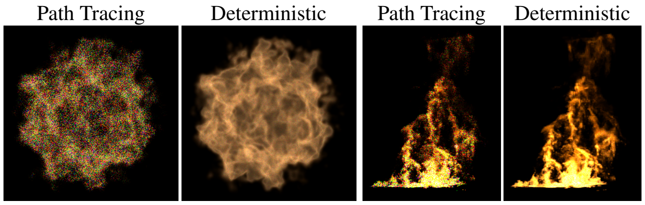


Figure 2: *Path tracing thin emissive media is challenging: path segments pass through the volume most of the time, resulting in noisy images (images: path tracing, 128 samples per pixel). Deterministically collecting all emission along a ray computes better results (images: deterministic). However, this can be inefficient if the volume has also dense parts.*

ing to transmittance will often choose a next path vertex behind the volume. Since the emission is only picked up at path vertices, it is poorly sampled and this results in noisy images (see Fig. 2). Alternatively, one can deterministically step through the whole volume and collect the emission with a quadrature rule. While this deterministic approach has no variance, it can be wasteful to access all the volume data in case the volume contains dense areas, shadowing the emission behind them.

It has been shown that the rendering efficiency can also be increased by sampling the emission directly using next event estimation (NEE) [VH13]. However, this approach neglects the transmittance between a shading point (path vertex) and the sampled emissive point.

In dense media, this can lead to many samples with low contribution, and to high cost as the volume needs to be traversed to compute the transmittance between possibly distant locations. Accessing volume data, or evaluating procedural volumes, usually dominates the render cost, and marching through the entire volume or along long segments can easily increase render times by a large factor.

In this paper, we propose a novel *forward next event estimation* (FNEE) technique which generalizes regular NEE and unifies it with both transmittance sampling and deterministic integration: first, we introduce a *line integration technique* to gather the emission along entire segments between scattering events, and show how to derive a correctly weighted measurement contribution function therefor. Second, we combine this with importance sampling of emission: we sample an emitting path vertex as in regular NEE, but then discard the vertex location and only keep the direction into which we trace a ray. The distance along this ray where emission is gathered is then determined by sampling proportional to transmittance.

Note that this is not sampling the product of emission and transmittance, but it does consider emission (leading to higher contribution) and transmittance (leading to higher efficiency because it touches less data). In conjunction with line integration, our technique can outperform regular NEE in many cases.

2. Background and Previous Work

Radiative Transport A participating medium can be fully characterized by its phase function φ along with its coefficients for scattering μ_s, absorption μ_a, emission μ_e, and extinction $\mu_t = \mu_s + \mu_a + \mu_e$. To clearly differentiate between the scattering cross section and the emitted radiance of a particle, we chose to explicitly model the source term as the product $\mu_e L_e$ of a separate coefficient for emissive particles μ_e and the emitted radiance L_e (which is then in Watts per square meter, analogous to the surface case). This will let us reason more clearly about emissive media which are thin (low μ_e) and bright (high L_e), such as candle lights.

The interaction of light with a volume is formulated with the radiative transfer equation (RTE) [Cha60, KVH84], which states that the change of radiance at a point x in direction ω is

$$\begin{aligned}(\omega \cdot \nabla)L(x,\omega) &= \mu_e(x)L_e(x) - \mu_t(x)L(x,\omega)\\ &+ \mu_s(x)\int_\Omega \varphi(x,\omega,\omega')L(x,\omega')d\omega'.\end{aligned} \tag{1}$$

In practice, the spatially varying coefficients are given in analytic form (procedurally) or as finite volume elements (voxels). If we ignore scattering (which is later handled by recursive path tracing), Eq. 1 simplifies to

$$(\omega \cdot \nabla)L(x,\omega) = \mu_e(x)L_e(x) - \mu_t(x)L(x,\omega) \tag{2}$$

which is a standard first-order differential equation with solution

$$L(x,\omega) = L(x',\omega)e^{-\int_0^t \mu_t(s)ds} + \int_0^t \mu_e(s)L_e(s)e^{-\int_0^s \mu_t(s')ds'}ds, \tag{3}$$

with reparameterized L_e, μ_e, μ_t (i.e $L_e(s) = L_e(x - s\omega)$). Here $L(x',\omega)$ is the radiance at the surface geometry which serves as boundary conditions. The *transmittance* term

$$\tau(t) = e^{-\int_0^t \mu_t(s)ds} \tag{4}$$

in the equation describes the transport of an colluded light between two points in the scene and corresponds to the loss of light along a distance t in the medium due to extinction. A much more detailed description of volume rendering and the radiative transport equation can be found in [HHS96].

Path Tracing In computer graphics, Eq. 1 is most commonly converted to integral form and solved with the Monte Carlo method, in the simplest case using *path tracing* [Kaj86].

Monte Carlo rendering requires the *sampling of distances* to the next event in the medium. Sampling such free path lengths is usually done proportionally to the transmittance and thus tends to create short path segments in dense and long path segments in thin media. This can be done implicitly by Woodcock tracking [WMHL65, RSK08] or explicitly by ray marching until the transmittance falls below a randomly chosen threshold [PH10]. Ray marching with fixed step sizes is biased which can be undesirable; instead, in our implementation we march from voxel boundary to voxel boundary, and optionally account for trilinear interpolation [SKTM11], to achieve unbiased results.

Sampling distances differently, e.g. according to the product of transmittance and emission, can be done by constructing a discrete cumulative distribution function (CDF) as in [KF12]. However, this can be costly if the volume contains fine high frequency details.

Usually some form of direct light evaluation (next event estimation, NEE, [Kaj86]), which creates endpoints of paths directly on a light source, is used to increase the efficiency of path tracing. Our forward next event estimation is similar to [MBJ*06] in that it

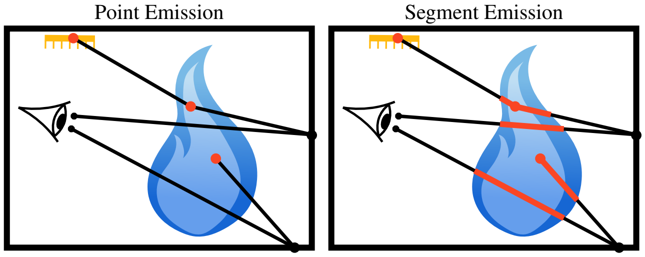


Figure 3: *Path tracing successively samples interaction points with surfaces or within participating media. The emission is evaluated only at the interaction points along the resulting transport paths (left). Therefore, for thin volumes like flames, the emission is poorly sampled since most path segments pass through the medium, yielding noisy images. Right: we propose a new Monte Carlo estimator that takes into account the emission at both the interaction points as well as along path segments.*

creates directions to a light source instead of points on it, but it still resembles next event estimation because the path is terminated at the resulting scattering point.

Rendering Participating Media Many rendering algorithms have been extended to handle participating media. This includes photon mapping [JC98], bidirectional path tracing [LW96], and Metropolis Light Transport [PKK00]. Many-lights methods [DKH*13] create emissive auxiliary lighting primitives in the scene and thus also in volumes, and in principle emissive volumes are possible. Various methods improve the integration with the help of beam estimates, e.g. [JZJ08, NNDJ12b, NNDJ12a]. These estimators have been unified with point estimators in the general framework of multiple importance sampling [KGH*14]. However, none of these methods specifically targets the rendering of emissive volumes.

Next Event Estimation in Emissive Volumes Villemin and Hery [VH13] also focus on handling emissive volumes in the context of Monte Carlo rendering. They use multiple importance sampling (MIS) [VG95] to combine emission found by chance using importance sampling of the scattering function, with emission that was directly sampled with NEE. However, sampling according to emission alone, without accounting for the transmittance between the sampled point and the shading point, can be problematic: it can result in long path segments through the volume for which the transmittance evaluation is expensive. And in case the transmittance is low, the effort is wasted as the sample does not contribute much to the illumination. Furthermore, the emission is only evaluated for the sampled point; in contrast, we propose to account for both transmittance and emission along path segments.

3. Computing Volumetric Emission

In this section, we derive the steps leading to our forward next event estimation technique. This estimator works robustly for both thin and dense media (and the spectrum in-between), samples according to emission and transmittance, and gathers emission along the next event path segments. It is based on the observation that if L_e is stored close to μ_t or density values, this will result in the same or very similar data accesses as just sampling a free distance, and thus mostly runs at the same speed (see Sec. 4 for details); we believe that similar considerations hold for procedurally generated volumes.

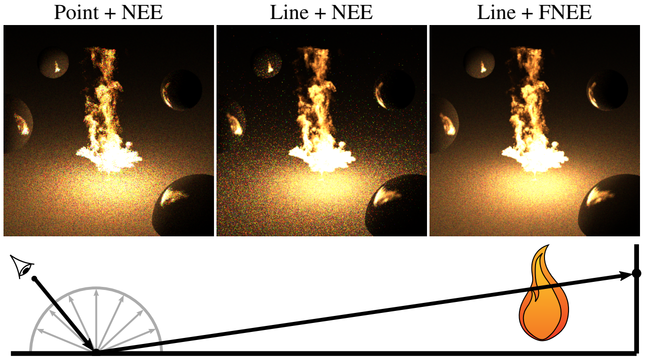


Figure 4: *Equal-sample comparison. Left: regular next event estimation (NEE) yields noisy results in the directly visible thin flame. Center: integrating emission along path segments improves the rendering of the flame. However, additional noise on the floor arises, as line integration after BSDF sampling and NEE cannot be combined using MIS. Right: our forward NEE also accounts for emission along segments and can thus be combined with line integration.*

We obtain our technique in three steps:

- In a standard Monte Carlo estimator, the volumetric emission is gathered at randomly chosen points along a ray. We introduce a *line integration* technique (Sec. 3.2) which gathers the volumetric emission along ray segments up to each chosen point (see Fig. 3). By this we include more information into the estimator, however, the emission needs to be weighted appropriately to compensate for the multiple gathering (multiple overlapping segments) and to yield an unbiased result. A similar concept is known in the field of neutron transport [SG69].
- We show how to combine the point and line integration estimators into a unified estimator which is well-suited for both thin and dense volumes (see Sec. 3.3).
- Line integration for BSDF (or phase function) sampling and NEE cannot be combined using MIS: line integration can be used for segments passing through the medium, while NEE cannot create such paths and the MIS weight would be zero for such cases (see Fig. 4). To this end, we introduce *forward NEE*: it importance samples the volume emission and converts the sample location into a direction. Along this direction, we use line integration, and can thus use MIS again. This method also leads to improved next event efficiency for dense volumes (see Fig. 5).

3.1. Point Integration

Our aim is to compute the incident radiance at a point x in the scene from a direction ω due to volumetric emission. Ignoring surface geometry and multiply scattered light (which is handled by the Monte Carlo path generation), it follows from Eq. 3 that we have to integrate the emitted radiance $\text{Ł}_e(t) = \mu_e(t) L_e(t)$ weighted by the corresponding transmittance $\tau(t)$ along the ray $r(t) = x + t\omega$:

$$\int_0^\infty \text{Ł}_e(t)\, \tau(t)\, dt. \quad (5)$$

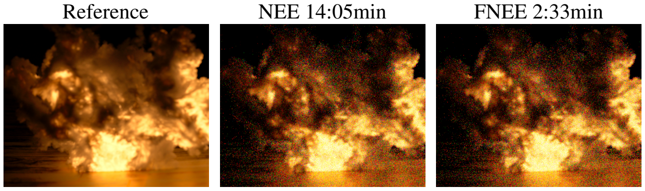


Figure 5: *Next event estimation for dense volumes can be wasteful as only nearby points contribute to a path due to high extinction. This is reflected in large time differences between regular NEE and our FNEE. Both images use 32 samples per pixel and have roughly equal quality. The volume properties are* $\sigma_t = 32$, $\sigma_s = 30$, $\sigma_e = 0.1$.

Since this integral has no analytic solution in the general case, we perform a numerical integration. Using N randomly sampled distances t_i with probability density $p(t_i)$, we can compute a Monte Carlo estimate of Eq. 5 with

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\text{Ł}_e(t_i)\,\tau(t_i)}{p(t_i)}, \tag{6}$$

which we will denote as the *point estimator*, as the emission is only evaluated at the discrete points $r(t_i)$ along the ray.

The distance sampling probability is usually chosen to be proportional to the transmittance, i.e. $p(t) \propto \tau(t)$. By this, however, regions with high emission can easily be missed, e.g. in very thin (low μ_t and μ_e, respectively) and bright flames (large L_e). Note that a typical path tracing implementation samples only one distance t_i for the next scattering event and integrates over many light paths per pixel instead. However, the arguments in the following section might be more intuitive for one ray and larger N.

3.2. Line Integration

When computing one sample t_i of the point estimator (Eq. 6), we also compute the transmittance along the segment $[0;t_i]$. The motivation for the line integration is to also consider the emission along that segment to improve the estimator, assuming that the underlying data structure allows simultaneously gathering emission at only little additional cost. However, simply replacing $\text{Ł}_e(t_i)\tau(t_i)$ by $\int_0^{t_i}\text{Ł}_e(t)\tau(t)dt$ in Eq. 6 does obviously not work, as the emission of closer segments along the ray is gathered multiple times (see also Fig. 6). To compensate for this, we introduce an emission weight function, $w_t(s)$, whose choice and derivation we discuss in this section. Note that t is the parameter which we sample for distances, and s the ray parameter along one of the segments. We need to choose $w_t(s)$ such that

$$\text{Eq. } 5 \overset{!}{=} \int_0^\infty \left(\int_0^t w_t(s)\,\text{Ł}_e(s)\,\tau(s)\,ds\right)dt. \tag{7}$$

We can now formulate two conditions to be met by the weight function: first of all, we want $w_t(s) = 0\,\forall s > t$, which means that in order to compute the emission along a ray segment up to a sampled location, i.e. $[0;t]$, we do not want to consider emission values at locations further away than t. Second, $w_t(s)$ has to reweight the contributions properly: if we fix an emissive point s on the ray (Fig. 6), the weight function needs to be *normalized over all possible sample points* t, i.e. (note $w_t(s) = 0\,\forall s > t$)

$$\int_0^\infty w_t(s)\;dt = \int_s^\infty w_t(s)\;dt = 1. \tag{8}$$

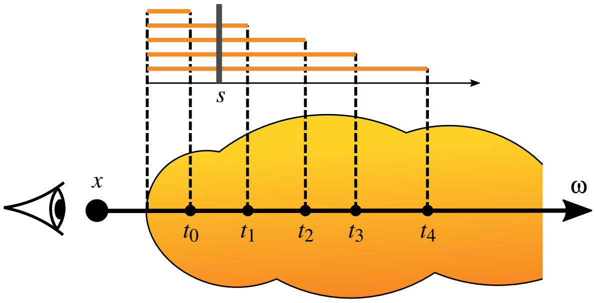


Figure 6: *Line integration computes the incident radiance due to volumetric emission along a ray* $x + t\omega$ *by randomly sampling distances* t_i *and accumulating the emission along the segments* $[0,t_i]$. *To account for multiple gathering of the emission at a point* $s \in [0,t_i]$, *we weight the contribution with a function* $w_t(s)$ *with* $\int_0^\infty w_t(s)dt = 1$.

We can verify that Eq. 7 yields the same result as Eq. 5 by first extending the interval of the inner integral ($w_t(s) = 0\,\forall s > t$), then swapping the order of integration, and reordering the terms:

$$\begin{aligned}\text{Eq. } 7 &= \int_0^\infty \left(\int_0^\infty w_t(s)\text{Ł}_e(s)\tau(s)ds\right)dt\\ &= \int_0^\infty \left(\int_0^\infty w_t(s)dt\right)\text{Ł}_e(s)\tau(s)ds\\ &= \int_0^\infty \text{Ł}_e(s)\tau(s)ds.\end{aligned} \tag{9}$$

Choice of Weight Function The above definition allows for different weight functions. Choosing $w_t(s) = \delta(s,t)$, for example, demonstrates that the line integral formulation in Eq. 7 is in fact a generalization of the point integral from Eq. 5 (of course, this choice does not change nor improve the estimator). We propose to use a weight function motivated by transmittance sampling: the transmittance $\tau(s)$ is the probability of sampling a location $t > s$ along the ray and therefore also the probability for accounting for the emission at s. In order to compensate for the multiple gathering of $L_e(s)$ during line sampling, we want $w_t(s) \propto 1/\tau(s)$ for $s \leq t$ and $w_t(s) = 0$ otherwise. In addition, we need a normalization term for $w_t(s)$ which also reflects the probability of sampling $t > s$. The probability density of sampling a distance t according to transmittance is $p(t) = \mu_t(t)\tau(t)$. Consequently, the probability of sampling a location $t > s$ and thus accounting for $L_e(s)$ again is:

$$\int_s^\infty \mu_t(t)\tau(t)dt = \tau(s) \Leftrightarrow \int_s^\infty \frac{\mu_t(t)\tau(t)}{\tau(s)}dt = 1. \tag{10}$$

Comparing Eqs. 8 and 10 shows that a valid weight function (with the aforementioned proportionality) is:

$$w_t(s) = \begin{cases}\mu_t(t)\tau(t)/\tau(s), & \text{if } s \leq t\\ 0, & \text{otherwise.}\end{cases} \tag{11}$$

If the volume is bounded at a distance T, either because there is a surface or the volume ends, the remaining weight has to be assigned to the case when the sampled distance t_i exceeds T. This

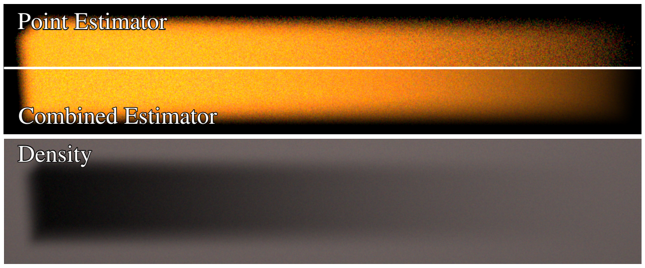


Figure 7: *A simple cube-shaped medium with decreasing density from left to right. The combined point+line estimator performs better than the point estimator alone (top row). This is best visible in the thin parts of the volume.*

is exactly the same as for the transmittance sampling probability density function (PDF).

Line Estimator The Monte Carlo estimator for Eq. 7 with a valid weight function becomes:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{\int_0^{t_i} w_{t_i}(s)\text{Ł}_e(s)\,\tau(s)\,ds}{p(t_i)}. \tag{12}$$

Using our weight function from Eq. 11 and transmittance sampling for the distances t_i, the Monte Carlo estimator can be simplified to

$$\frac{1}{N}\sum_{i=1}^{N}\int_0^{t_i}\text{Ł}_e(s)\,ds. \tag{13}$$

Discussion Eq. 13 yields the same result in the limit as the point estimator in Eq. 6, but it uses more information for a single sample by collecting the emission along the line segment. The special case of the line estimator with transmittance sampling (Eq. 13) is also known as the track length estimator in neutron transport theory [SG69]. However, Spanier and Gelbard derive only this particular Monte Carlo estimator whereas we provide a general derivation using weight functions. Even though we sample distances only according to transmittance in this paper, our general line estimator (Eq. 12) works with other distance sampling methods as well, for example sampling according to μ_s instead of μ_t for highly scattering media or equi-angular sampling [KF12]. We provide an alternative derivation of the track length estimator in the supplemental material.

3.3. Combining the Line and Point Estimators

The previously introduced line estimator is superior to the point estimator in thin media as it gathers emission even when no scattering event was sampled inside the volume. However, it does not always have lower variance than the point estimator. For example, if Ł_e and μ_t are nearly constant along the line segment, the point estimator will have little variance if transmittance sampling is used:

$$\frac{\text{Ł}_e(t_i)\,\tau(t_i)}{p(t_i)} \approx \frac{\text{Ł}_e}{\mu_t} \approx const.$$

However, the contribution of the line estimator in this case is

$$\int_0^{t_i}\text{Ł}_e(s)\,ds \approx \text{Ł}_e\,t_i,$$

where the random segment length t_i introduces variance.

For this reason, we want to combine the strengths of both estimators: we want the point estimator to have a higher contribution than the line estimator in dense regions and vice versa. To this end, we propose weighting their contributions *at a particular location in the medium* according to the transmittance along the ray up to that point. Note that this is *not* equivalent to a convex combination of Eqs. 6 and 12 because of the spatially-varying weight. We obtain the combined estimator by first splitting Eq. 5:

$$\int_0^{\infty}(1-\tau(t))\text{Ł}_e(t)\tau(t)dt + \int_0^{\infty}\tau(t)\text{Ł}_e(t)\tau(t)dt,$$

and use the point estimator for the left, and the line estimator for the right integral. We compute the latter analogously to our previous derivation with

$$\int_0^{\infty}\tau(t)\text{Ł}_e(t)\tau(t)dt = \int_0^{\infty}\int_0^{\infty}w_t(s)\tau(s)\text{Ł}_e(s)\tau(s)ds\,dt$$

and obtain the combined estimator as

$$\frac{1}{N}\sum_{i=1}^{N}\left((1-\tau(t_i))\frac{\text{Ł}_e(t_i)\tau(t_i)}{p(t_i)} + \frac{\int_0^{t_i}w_{t_i}(s)\tau(s)\text{Ł}_e(s)\tau(s)ds}{p(t_i)}\right).$$

If we sample t_i proportional to transmittance, then the combined estimator conveniently simplifies to:

$$\frac{1}{N}\sum_{i=1}^{N}\left((1-\tau(t_i))\frac{\text{Ł}_e(t_i)}{\mu_t(t_i)} + \int_0^{t_i}\text{Ł}_e(s)\,\tau(s)\,ds\right). \tag{14}$$

Note the additional spatially-varying transmittance weight inside the integral of the line estimator. Fig. 7 shows equal-sample results of the combined estimator compared to the point estimator.

Discussion The combination of the point and line estimator is similar to MIS when distances are sampled according to transmittance (the transmittance is part of the PDF in this case). However, MIS normalizes the contribution of a discrete set of estimators, whereas we normalize over a continuous domain using a weight function. We will show in Sec. 5, that our combination is, similar to MIS, not perfect. We experimented with other combinations and weight functions, but they were either much more complicated or needed information along the full ray through the volume. The introduced weight functions and combination of the point and line estimator are the most robust and efficient we found, and they proved to work well for a wide variety of volumes (see Sec. 5).

3.4. Forward Next Event Estimation

With line integration as part of the combined estimator, the emission along every path segment contributes to the image, even when there is no scatter event inside the volume. Combining such *pass-through paths* with volume NEE [VH13] using MIS is problematic. Since NEE only samples emissive points it cannot create these pass-through paths and thus their corresponding MIS-weights become zero, i.e. MIS becomes useless. For example, consider pass-through paths from BSDF-sampling as shown in Fig. 4. Here, the volume subtends only a small solid angle and the probability of hitting the volume with a random BSDF-sampled direction off the rough floor is very low and the path cannot be created by NEE. Here NEE means that an emissive point is sampled inside the volume. For NEE with other light sources see Sec. 3.5.

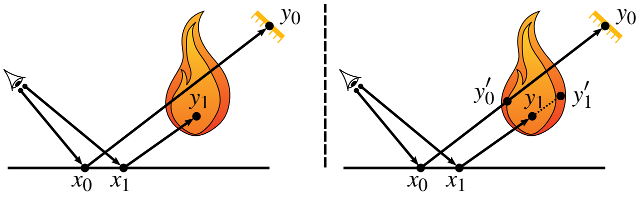


Figure 8: *Left: path tracing with regular NEE constructs paths in two ways: sampling y via outgoing directions from x and distance by transmittance, or by sampling a point on a light source. Right: FNEE provides a third way. At first, a vertex y′ inside the volume is sampled according to emission. Then, a distance towards y′ is sampled by transmittance. This process may pass through the volume and result in a vertex y on a geometric light source. Such paths will have to be weighted against NEE on surfaces.*

To overcome this problem we introduce *forward next event estimation* (FNEE). The key idea is to reinterpret the point sampling process of volume NEE as a solid angle direction sampling. For this, we discard a sampled point but keep the direction to it. Thereafter, we sample a distance according to transmittance along that direction, and use the combined estimator from Eq. 14 (as we do for phase function or BSDF scattering). This technique has three advantages:

- We can again use the combined estimator to gather emission along segments.
- It can sample pass-through paths and has meaningful MIS weights for every contributing path.
- Sampling distances according to transmittance avoids deep penetration of dense volumes and thus improves the efficiency of direct light computation significantly.

The main challenge is that multiple points in the volume result in the same direction ω from a shading point x. To compute the solid angle measure probability of the resulting direction, $p_\sigma(\omega)$, we need to integrate the probability over all points, i.e. compute

$$p_\sigma(\omega) = \int_0^\infty p_x(t)\, t^2\, dt, \tag{15}$$

where $p_x(t)$ is the vertex volume sampling probability of a point $P(t) = x + t\omega$, and t^2 is the transformation Jacobian from volume to solid angle measure.

This integration is computationally expensive as it requires stepping from x in direction ω through the entire volume to accumulate the point sampling probabilities $p_x(t)$. Obviously, this would not be cheaper than gathering the entire emission directly. However, we can perform point sampling and PDF integration at a low-resolution version of the volume without noticeable differences in the rendering. Even though fine emission details could be missed – which would have a significant impact on regular NEE – line integration has a high chance to pick up the high frequency details again. Note that the distance sampling and line integration are still performed on the full volume resolution and therefore the estimator is unbiased. We evaluate this sampling in Fig. 11.

Probabilistic Transmittance for NEE FNEE is faster for dense volumes because the additional transmittance sampling prevents unnecessarily deep penetration. The same speed benefit can be acquired for NEE with an unbiased probabilistic transmittance approximation. For this, we sample a distance along the ray to the sampled NEE point according to transmittance and if the distance sampling stops before the NEE point, the transmittance will be zero, otherwise one. This can be viewed as a form of Russian roulette (RR). We found however, that this approach can actually increase variance resulting in higher *equal-time* RMSE (see Fig. 9) compared to regular NEE. FNEE on the other hand, uses line integration which increases the likelihood of useful contributions and always outperformed NEE with and without RR in our experiments. All comparisons apart from Fig. 9 are with regular NEE since it is the published state of the art at this time [VH13]. For a comparison where RR is beneficial compared to regular NEE (but not compared to FNEE) please refer to the supplemental material.

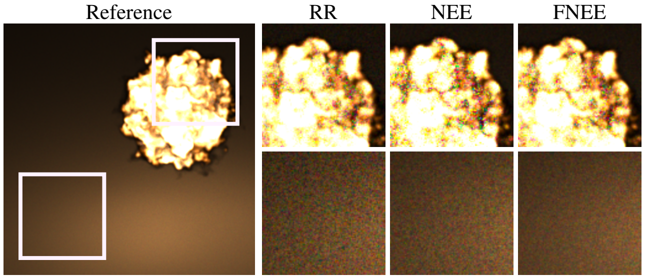


Figure 9: *Equal-time comparison of NEE with probabilistic transmittance sampling (RR, 487spp, left), regular NEE (232spp, middle) and our FNEE (394spp, right). The quality of all techniques is roughly the same in the directly visible flame (top row), however, the noise in the RR version is higher compared to NEE and even more so compared to FNEE in the illuminated parts (bottom row). All renders took 3 minutes. The RMSE over the cropped view in the bottom row are (from left to right):* 0.87, 0.65, 0.49.

3.5. Other Light Sources

FNEE can create path segments that pass through the volume and potentially hit other light sources, e.g. area lights or environmental illumination (see Fig. 8). Such paths can be constructed by FNEE in the volume or NEE of the light source. We model this in the MIS framework by stochastically choosing surface NEE or volume FNEE via the one-sample MIS model [VG95]. If there are multiple emissive volumes in a scene, they can all contribute to the combined PDF of a path. The easiest way to account for this is to treat them as a single aggregate volume.

4. Implementation

In this section, we provide details on the implementation of forward next event estimation and line integration in a path tracing framework, and begin with details of the volumetric data structure.

Volumetric Data Structure We use a voxel back end to represent spatially varying volume properties; in particular, we use a 4D storage similar to [Wre16]. The basic structure is a hierarchical grid with an $8\times8\times8$ voxel branching factor, similar to the one used in OpenVDB [Mus13]. For computing the images in this paper, we stored density d and temperature in every voxel. We use a constant phase function and reconstruct the spatially varying scattering parameters as $\mu_* = \sigma_* \cdot d$, where $\sigma_{s,a,e}$ are user parameters which can be changed at run time. A shader is used to compute the emission L_e from the temperature, in most of our results by implementing a black body emitter.

We construct a full binary tree for every grid to form a light hierarchy. This enables us to sample a voxel randomly according to $d \cdot L_e$ (as in [VH13]). Ultimately, we want to sample by $\mu_e L_e = \sigma_e \cdot d \cdot L_e$, but the scalar user constant σ_e is factored out as it only affects normalization.

The concepts derived in this paper are largely independent of the exact data structure used. We also implemented the technique in a production renderer using a kd-tree instead of hierarchical grids.

Free Path Sampling To compute the distance to the next scattering event of a path, we traverse the volume along a ray and accumulate the transmittance on the leaf level of the voxel hierarchy until it falls below a randomly chosen threshold [SKTM11]; this is possible as the transmittance is normalized.

Traversing the voxels makes the evaluation of the segment emission integral in Eq. 14 straightforward. For our results, we assumed homogeneous media inside voxels, i.e. constant μ_e, L_e, μ_t, μ_s, which allows to analytically compute

$$\int_0^T \mu_e L_e \tau(s) ds = \mu_e L_e \int_0^T e^{-\mu_t s} ds = \frac{\mu_e}{\mu_t} L_e (1 - e^{-\mu_t T}), \qquad (16)$$

where T is the length of the ray segment inside the voxel. Pseudocode of the free path sampling and the combined estimator can be found in Alg. 1. Computing segment emission along with the transmittance between two points works analogously.

We also tested Woodcock tracking in combination with a coarse grid storing the majorant $\mu_t^{\max}$, however, for all volumes in our scenes it resulted in more memory accesses and longer run times. Note that in cases where Woodcock tracking jumps over thin homogeneous regions (i.e. using fewer memory accesses), it also skips potentially important emission.

Point sampling for NEE To sample a light vertex for next event estimation, we use 3D sample warping on a full binary tree around the voxels of each grid. The target distribution is proportional to $\mu_e L_e$ averaged over wavelength. The intermediate levels sample the averages over the time domain, intermediate levels coinciding with the inner nodes of the hierarchical grid, however, also resolve the probabilities in time.

Forward NEE directional PDF We compute the PDF (in solid angle measure) for the direction sampling of FNEE by evaluating the integral in Eq. 15 for homogeneous voxels as shown in Alg. 2. While free path sampling (also during FNEE) and point sampling for regular NEE are always performed on the highest resolution of our voxel data structure, we use a coarser representation for FNEE direction sampling for efficiency reasons. This simply changes the list of voxels to traverse in Alg. 2. The term $(t_1^3 - t_0^3)/3$ is the analytic solution of the integral in Eq. 15 for homogeneous volumes after pulling out the constant PDF $p_x(t)$.

Algorithm 1 Free Path Sampling with Segment Emission

```
procedure SAMPLEFREEPATH($o, \omega, \xi$)
    $T \leftarrow 1$ // transmittance
    $L_e^c \leftarrow 0$ // combined estimate
    for all voxels along $o + t\omega$ do
        $\{t_0, t_1\} \leftarrow$ distance of voxel entry/exit point
        $\{\mu_t, \mu_e L_e\} \leftarrow$ extinction/emission of voxel
        $d \leftarrow t_1 - t_0$
        $T' \leftarrow T \cdot e^{-\mu_t d}$
        if Scatter($\xi, T, T'$) then
            $t_s \leftarrow$ distance to scatter point in voxel
            $d \leftarrow t_s - t_0$
            $L_e^c \leftarrow L_e^c + T \cdot L_e \frac{\mu_e}{\mu_t} \cdot (1 - e^{-\mu_t d})$
            $T \leftarrow T \cdot e^{-\mu_t d}$
            $L_e^c \leftarrow L_e^c + (1 - T) L_e \frac{\mu_e}{\mu_t}$
            return $\{t_0 + d, L_e^c\}$
        end if
        $L_e^c \leftarrow L_e^c + T \cdot L_e \frac{\mu_e}{\mu_t} \cdot (1 - e^{-\mu_t d})$
        $T \leftarrow T'$
    end for
    return $\{\infty, L_e^c\}$
end procedure
```

Algorithm 2 Compute direction PDF

```
procedure ACCUMPDF($o, \omega$)
    $p \leftarrow 0$
    for all voxels along $o + t\omega$ do
        $\{t_0, t_1\} \leftarrow$ distance of voxel entry/exit point
        $p_v \leftarrow$ probability of choosing voxel with point sampling
        $p \leftarrow p + (t_1^3 - t_0^3)/3 \cdot p_v$
    end for
    return $p$
end procedure
```

5. Results

All images in this paper were generated with a custom Monte Carlo renderer written in C using spectral rendering. The renderer used a single wavelength per path, but line integration and FNEE would also work with multiple wavelengths or RGB rendering. The images are inherently dark, to keep the fine detail in the emissive volumes. However, that impairs the quality for printing and we recommend to view the images on a computer screen. The renderings in all images and equal-time measurements where done using a machine with a quad-core Intel i7-3770 and 16GB RAM, except for Fig. 13, which was rendered on a 128-core machine with 4TB of main memory.

We use six volumes in the paper with varying complexity (see Tab. 1). All volumes use isotropic scattering (i.e. a mean cosine of zero). Except for the second scene in Fig. 12, which uses a custom shader to mimic bioluminescence, all volumes are black body emitters.

5.1. Evaluation of the Individual Estimators

To evaluate the different estimators (point, line and combined) we rendered the fireball volume for varying densities across four orders of magnitude in Fig. 10. The emission is scaled to produce

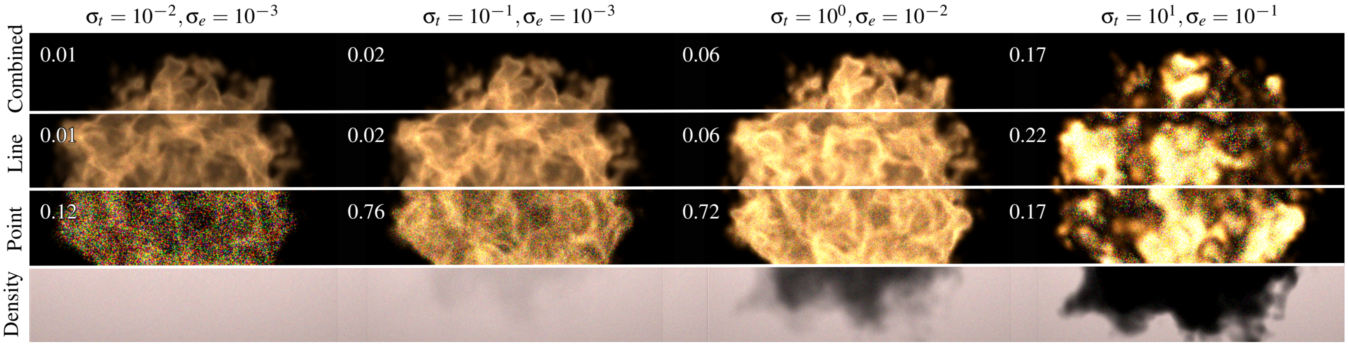

$\sigma_t = 10^{-2}, \sigma_e = 10^{-3}$
$\sigma_t = 10^{-1}, \sigma_e = 10^{-3}$
$\sigma_t = 10^{0}, \sigma_e = 10^{-2}$
$\sigma_t = 10^{1}, \sigma_e = 10^{-1}$
Combined
0.01
0.02
0.06
0.17
Line
0.01
0.02
0.06
0.22
Point
0.12
0.76
0.72
0.17
Density

Figure 10: *The fireball from Fig. 2 rendered with the point (Sec. 3.1), line (Sec. 3.2) and the combined point+line estimator (Sec. 3.3) for varying densities. The emission is scaled to maintain roughly the same brightness. The density values are scaled from left to right, $\sigma_s = 0$ for all cases. The the RMSE after 128 samples per pixel is given for each image. Compared to our integration method, which is robust for several orders of magnitude of density values, the point integration becomes worse the thinner the medium gets.*

Volume	Size	#Voxels	Figure
wedge	10MB	$\approx 2 \cdot 10^6$	7
flame	28MB	$\approx 3 \cdot 10^6$	2, 4, 12
swirl	150MB	$\approx 21 \cdot 10^6$	12
fireball	390MB	$\approx 39 \cdot 10^6$	2, 10, 9
explosion	3.5GB	$\approx 349 \cdot 10^6$	1, 5, 11
sun	30GB	$\approx 6903 \cdot 10^6$	13

Table 1: *Size of the volumetric datasets used in this paper.*

roughly the same total brightness in each case. We can see that the line and combined estimators gain advantage over the traditional point estimator for decreasing thickness of the volume. However, for very dense volumes the point estimator is better than the line estimator since the segment length introduces additional variance. The combined estimator performs better than the line estimator in this case, although it does not reach the quality of the point estimator for dense volumes. Please refer to the supplemental material for a more detailed analysis of the different estimators.

5.2. Forward Next Event Estimation

Comparison to NEE In Fig. 4 a simple test scene shows the problem arising when line integration is used with regular NEE (middle image). The high variance paths created by forward BSDF sampling cannot be combined with next event estimation using MIS in this case, as NEE cannot create pass-through paths which contribute due to segment emission. On the other hand, forward next event estimation is able to create pass-through paths. Combining BSDF sampling with FNEE using MIS results in an overall superior result, also compared to point integration with NEE (right image).

Computing the Solid Angle PDF FNEE requires computing the solid angle PDF for a sampled direction (Eq. 15). As detailed in Sec. 4 we compute the PDF by integrating along a ray through a coarser version of the volume. In our implementation we can simply choose one of the hierarchical grid levels. Next we evaluate the impact of using coarser levels.

For this test, we computed the irradiance for two locations p_0 and p_1, illustrated in Fig. 11, outside a volume where next event estimation is the most useful. Interestingly, the equal-sample comparison of the first row show that the choice of coarseness does not have a large impact on the sample quality in this test, even though the solid angle subtended by the volume differs greatly at these two locations. The line integration is tolerant to slightly more inaccurate initial point sampling, and compensates for that by gathering the emission at high resolution along path segments.

The middle and bottom rows in Fig. 11 show equal-time comparisons. FNEE-0 has a larger RMSE because it is much more expensive than FNEE-1 and FNEE-2. All renders use the second finest grid (FNEE-1) for forward next event estimation except for the explosion in Figs. 1 and 5 where FNEE-2 is used. This shows that even though the explosion dataset has 349M voxels, we can use a 64^3-version of the volume to perform forward next event estimation.

Volume Density The density of a volume also influences the efficiency of next event estimation. If the volume is dense, then it is likely that emission at a sampled point is irrelevant because of high extinction in the medium. The bottom row of Fig. 11 shows the RMSE after 100ms plotted for varying densities. We can see that FNEE gains advantage over NEE for increasing σ_t.

5.3. Test Scenes

We performed equal-time renders for different scenes and compared next event estimation using point integration and forward next event estimation using line integration. In all our test cases, FNEE delivers results with significantly less residual noise than NEE.

In Fig. 1 we rendered a city containing the explosion dataset with two different values for σ_t. The scattering parameters are $\sigma_t = 0.2$, $\sigma_s = 0.01$, $\sigma_e = 0.1$ for the thin version (left, 10min), and $\sigma_t = 40$, $\sigma_s = 0.1$, $\sigma_e = 64$ for the dense version (right, 30min). The overall density of the resulting volumes is shown in insets where the volume is rendered with $\sigma_e = 0$. The image resolution is 1024×576.

The thin version greatly benefits from line integration as can be seen in the insets showing directly visible parts of the volume. FNEE is much more efficient than NEE in the dense version: FNEE rendered more than twice as many samples as NEE in the same time frame producing a much better result than NEE.

Fig. 12 shows equal-time comparisons between NEE and FNEE

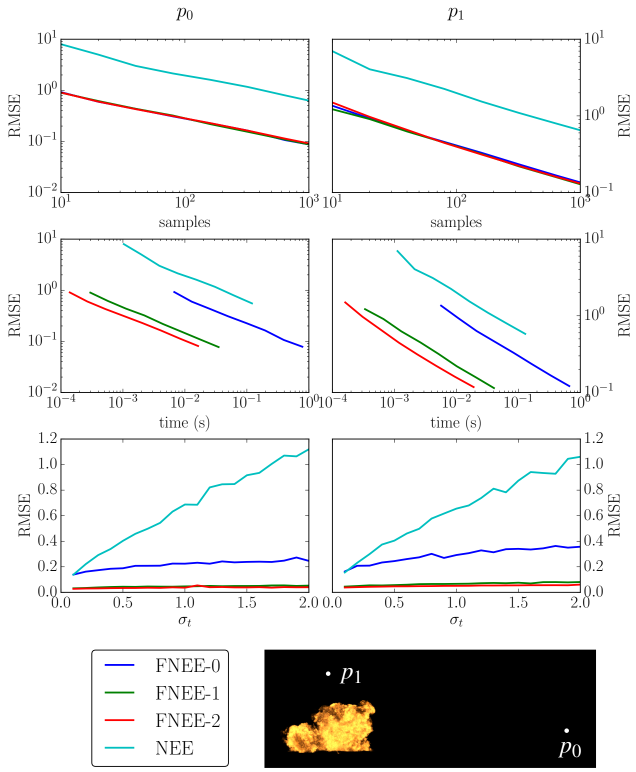


Figure 11: *The irradiance is computed for two locations p_0 and p_1 outside the volume, where next event estimation is the most useful. The RMSE is plotted for an equal amount of samples (top) and time (middle). The notation FNEE-i means that the grid i levels above the leaf level is used for FNEE (FNEE-0 is the finest). The bottom row shows the RMSE after 100ms plotted for varying densities showing the advantage of FNEE for dense volumes. Note how inefficiently NEE behaves for increasingly dense media. The top and middle row correspond to $\sigma_t = 1$.*

in two scenes (resolution 512 × 512). The upper scene contains a thin flame in a simple closed box. The flame is reflected by the mirroring back wall as well as the glossy objects. The renders are equal-time (15min). The insets show that FNEE outperforms NEE not only in parts where the flame is directly visible, but also in indirectly lit parts of the scene. In this scene FNEE is slower than NEE (FNEE: 47spp, NEE: 70spp) because the volume is thin and FNEE passes through the volume most of the time. Still, the RMSE compared to a converged reference is greatly reduced (FNEE: RMSE 0.13, NEE: RMSE 0.21). The bottom scene shows a frame of a fluid simulation where a rotating object mixes two initially separated layers of fluid with different densities. The volumetric emission in the fluid is proportional to the velocity, imitating bioluminescence. The renders took 10min and FNEE again has lower remaining noise (RMSE 0.05) compared to NEE (RMSE: 0.11) even with lower sample count (FNEE: 134spp, NEE: 174spp).

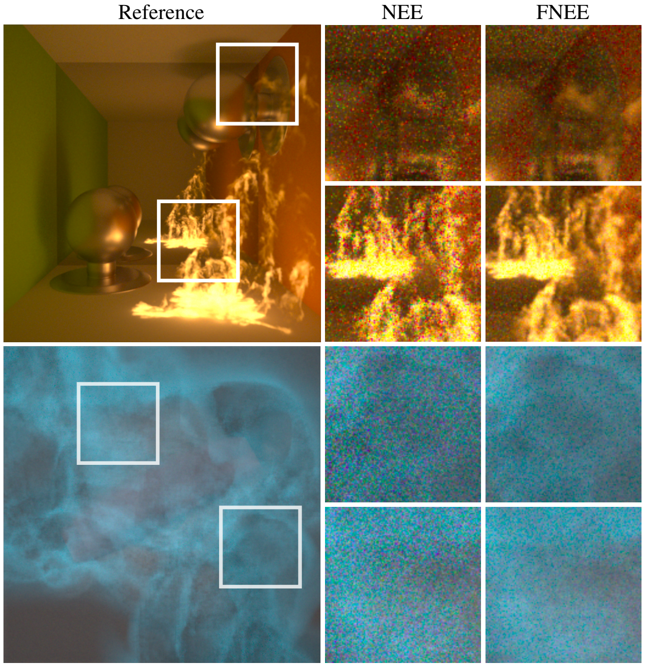


Figure 12: *Two equal-time comparisons of NEE with point integration (left) to our FNEE with line integration (right). Even though FNEE results in less samples per pixel the images contain less residual noise.*

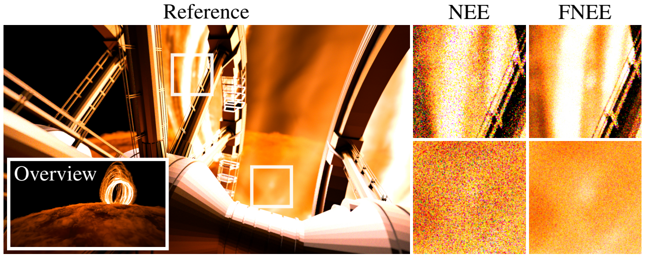


Figure 13: *Scalability test on a 30GB volume. These images show equal sample counts (128spp) with an RMSE of 52.6 for NEE and 36.7 for FNEE. The time for NEE was 7.5 and for FNEE 11.2 core hours, i.e. the time one core would need to create the image. Most of the light comes from the relatively thin loop – a case which works reasonably well with regular NEE. Note that this is a difficult scenario for FNEE as it creates long segments through the thin medium, i.e. it has a higher cost per sample than NEE. However, this is still amortized by collecting the emission along the path segment: the RMSEs for equal-time (60 core hours) are 14.9 FNEE (704spp) and 18.6 NEE (1024spp). For larger versions of these images please refer to the supplemental material.*

In Fig. 13, we rendered a 30GB scene to evaluate the cost of FNEE for larger volumes. As expected, the computational cost increases (PDF accumulation, longer segments in thin regions), but FNEE still pays off for equal-time renders. A version with bigger images can be found in the supplemental material.

6. Limitations

In this paper, we specifically addressed the challenges with rendering emissive volumes. Next event estimation in thin scattering fog, for example, is a different problem and is usually addressed with equiangular sampling [KF12] or by creating a CDF along a ray. The latter approach may also prove useful for emission, e.g. if very bright lights are hidden behind thick layers of absorbing smoke.

While our techniques would transparently work with Kelemen Metropolis [KSKAC02], it is not as straightforward for path space methods derived from Metropolis Light Transport [VG97]. This might be an interesting direction for future work.

We did not try using segment emission in bidirectional path tracing but it might be interesting to investigate this in future work. For deterministic connections, the segment emission can be computed in the same way as we already do for NEE paths to other light sources. However, MIS may need to be adjusted to account for the fact that paths sharing a common prefix close to the sensor can result in the same contributions (similar to when picking up point emission along all path vertices, not just the end points).

7. Conclusion

We presented a line integration estimator that accumulates emission for path segments, not only at path vertices. This now allows Monte Carlo light paths to contribute to the image in cases when no scattering event inside the volume has been sampled. This greatly improves Monte Carlo rendering in thin parts of emissive media.

We also proposed forward next event estimation, which combines sampling points according to emission and segment length according to transmittance. This prevents long segments that do not contribute to the final image in volumes with high extinction.

Whereas line integration works best for thin emissive volumes, FNEE is particularly beneficial for dense media with high extinction. Used together, they provide robust and efficient rendering of a wide range of emissive heterogeneous volumes, and we showed that resulting images can have greatly reduced Monte Carlo noise.

References

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

[DKH*13] DACHSBACHER C., KŘIVÁNEK J., HAŠAN M., ARBREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum 33*, 1 (2013), 88–104. 3

[HHS96] HEGE H.-C., HÖLLERER T., STALLING D.: *Volume Rendering: Mathematical Models and Algorithmic Aspects*. Tech. rep., ZIB (Konrad-Zuse-Zentrum), Berlin, 1996. 2

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. SIGGRAPH* (1998), pp. 311–320. 3

[JZJ08] JAROSZ W., ZWICKER M., JENSEN H. W.: The beam radiance estimate for volumetric photon mapping. *Computer Graphics Forum (Proceedings of Eurographics) 27*, 2 (2008), 557–566. 3

[Kaj86] KAJIYA J. T.: The rendering equation. *Computer Graphics (Proc. SIGGRAPH) 20*, 4 (1986), 143–150. 2

[KF12] KULLA C., FAJARDO M.: Importance sampling techniques for path tracing in participating media. In *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)* (2012). 1, 2, 5, 10

[KGH*14] KŘIVÁNEK J., GEORGIEV I., HACHISUKA T., VÉVODA P., ŠIK M., NOWROUZEZAHRAI D., JAROSZ W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Trans. on Graphics (Proc. SIGGRAPH) 33*, 4 (2014). 1, 3

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum 21*, 3 (2002), 531–540. 10

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

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Proc. Eurographics Workshop on Rendering* (1996), pp. 91–100. 3

[MBJ*06] MORLEY R. K., BOULOS S., JOHNSON J., EDWARDS D., SHIRLEY P., ASHIKHMIN M., PREMOŽE S.: Image synthesis using adjoint photons. In *Proceedings of Graphics Interface 2006* (2006), pp. 179–186. 2

[Mus13] MUSETH K.: VDB: High-resolution sparse volumes with dynamic topology. *ACM Trans. on Graphics 32*, 3 (2013). 6

[NNDJ12a] NOVÁK J., NOWROUZEZAHRAI D., DACHSBACHER C., JAROSZ W.: Progressive virtual beam lights. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering) 31*, 4 (2012), 1407–1413. 3

[NNDJ12b] NOVÁK J., NOWROUZEZAHRAI D., DACHSBACHER C., JAROSZ W.: Virtual ray lights for rendering scenes with participating media. *ACM Trans. on Graphics (Proc. SIGGRAPH) 31*, 4 (2012), 60:1–60:11. 3

[NSJ14] NOVÁK J., SELLE A., JAROSZ W.: Residual ratio tracking for estimating attenuation in participating media. *Trans. on Graphics (Proc. of SIGGRAPH Asia) 33*, 6 (2014). 1

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory to Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., 2010. 2

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

[SG69] SPANIER J., GELBARD E. M.: *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, 1969. 3, 5

[SKTM11] SZIRMAY-KALOS L., TÓTH B., MAGDICS M.: Free path sampling in high resolution inhomogeneous participating media. In *Computer Graphics Forum* (2011), pp. 85–97. 2, 7

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *Proc. SIGGRAPH* (1995), pp. 419–428. 3, 6

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proc. SIGGRAPH* (1997), pp. 65–76. 10

[VH13] VILLEMIN R., HERY C.: Practical illumination from flames. *Journal of Computer Graphics Techniques (JCGT) 2*, 2 (2013), 142–155. 2, 3, 5, 6, 7

[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. Conf. Applications of Computing Methods to Reactor Problems* (1965), p. 2. 2

[Wre16] WRENNINGE M.: Efficient rendering of volumetric motion blur using temporally unstructured volumes. *Journal of Computer Graphics Techniques (JCGT) 5*, 1 (2016), 1–34. 6

[ZRB14] ZHAO S., RAMAMOORTHI R., BALA K.: High-order similarity relations in radiative transfer. *ACM Trans. on Graphics (Proc. SIGGRAPH) 33*, 4 (2014), 104:1–104:12. 1