Direct Transmittance Estimation in Heterogeneous Participating Media Using Approximated Taylor Expansions

Daniel Jönsson, Joel Kronander, Jonas Unger, Thomas Schön and Magnus Wrenninge

Tweet

LiU LINKÖPING UNIVERSITY

Direct Transmittance Estimation in Heterogeneous Participating Media Using Approximated Taylor Expansions

Daniel Jönsson, Joel Kronander, Jonas Unger, *Member, IEEE,* Thomas B. Schön, *Senior Member, IEEE,* and Magnus Wrenninge

Abstract—Evaluating the transmittance between two points along a ray is a key component in solving the light transport through heterogeneous participating media and entails computing an intractable exponential of the integrated medium's extinction coefficient. While algorithms for estimating this transmittance exist, there is a lack of theoretical knowledge about their behaviour, which also prevent new theoretically sound algorithms from being developed. For this purpose, we introduce a new class of unbiased transmittance estimators based on random sampling or truncation of a Taylor expansion of the exponential function. In contrast to classical tracking algorithms, these estimators are non-analogous to the physical light transport process and directly sample the underlying extinction function without performing incremental advancement. We present several versions of the new class of estimators, based on either importance sampling or Russian roulette to provide finite unbiased estimators of the infinite Taylor series expansion. We also show that the well known ratio tracking algorithm can be seen as a special case of the new class of estimators. Lastly, we conduct performance evaluations on both the central processing unit (CPU) and the graphics processing unit (GPU), and the results demonstrate that the new algorithms outperform traditional algorithms for heterogeneous mediums.

Index Terms—Path tracing, rendering, computer graphics, scientific visualization.

✦

1 INTRODUCTION

Computing the attenuation of light through participating media, such as smoke, clouds and fire is one of the most fundamental parts of solving the radiant transport problem, i.e., how light travels in a scene. Because of its central role in rendering an image, it has been a long term goal of computer graphics to develop efficient algorithms for computing the attenuation though participating media. Similar radiant transport problems are also faced in several other application domains such as neutron transport and medical radiation dosimetry, thus highlighting the importance of radiant transport in a wide range of application areas. While much progress have been made towards robust and efficient algorithms for rendering participating media, scenes containing heterogeneous participating media are still computationally challenging to render, and can often be a bottleneck in production rendering.

The central challenge of evaluating the radiant attenuation along a path is often formulated in terms of computing the *transmittance*, describing the exponential decay of visibility along the path. In homogeneous media, where the media density is constant, evaluating the transmittance is trivial and has a closed form solution in Beer's law. However, in heterogeneous media where the density varies spatially, traditional methods often approximate the transmittance along the path by sampling the media density at a large number of points, which often leads to computationally expensive estimators.

While many of the traditional methods are guaranteed to generate correct results given enough samples, i.e. they are unbiased, there is a lack of theoretical knowledge about their behaviour such as their variance. This lack of theoretical knowledge means that it is difficult to make optimal choices and one must instead rely on experimental results.

To bridge this knowledge gap, we derive a new theoretical view on unbiased Monte Carlo transmittance estimators using a Taylor expansion of the exponential term. By combining the infinite Taylor sum, resulting from the expansion, with an unbiased Monte Carlo estimator we end up with an unbiased practical theoretical formulation for transmittance estimation. This new formulation provides a solid basis for analyzing and reasoning about Monte Carlo transmittance estimators and comes with a number of benefits. First of all, it allows us to derive theoretical variance bounds for the transmittance estimator and show that it is always better to use a higher upper bound than to average several (residual) ratio trackers along a single ray. Second, we can show that the previously proposed Ratio and Residual Ratio tracking algorithms [16] are special cases in our general formulation. Thus, our presented proofs also include these special cases which, in their original form had unknown variance bounds. Third, our theoretical foundation allow us to devise new transmittance estimation algorithms with properties suitable for different scenarios, e.g. better estimations without auxiliary costly data structures.

To this end, our main contributions are:

1) A new theoretical view of unbiased transmittance estimation algorithms based on a truncated Taylor expansion of the exponential.
2) Derivations of explicit variance bounds for the new class of estimators (including previous work on Ratio and Residual

- *D. Jönsson, J. Kronander, J. Unger are with Linköping University, Sweden. E-mail: joelkronander@gmail.com, {daniel.jonsson, jonas.unger}@liu.se.*
- *T. B. Schön is with Uppsala University, Sweden. E-mail: thomas.schon@it.uu.se.*
- *M. Wrenninge is with Pixar Animation Studios, USA. E-mail: magnus.wrenninge@gmail.com.*

Ratio tracking).

3) New types of Residual Ratio Tracking algorithms (3 and 4) that do not rely on additional costly data structures, yet outperforms previous Residual Ratio Tracking in heterogeneous media. .

4) A thorough parameter analysis and comparison between the transmittance estimators using both the CPU and the GPU.

2 TRANSMITTANCE ESTIMATION AND FREE-FLIGHT SAMPLING

For producing photorealistic renderings of participating media such as water, smoke, fire etc, it is necessary to consider light transport models accounting for the effects of how light interacts with the media in the scene. In computer graphics, and in many other fields of science such as neutron transport [21] and medical physics [1], the media is modeled as a large number of microscopic scattering particles interacting with the light. In computer graphics we are interested in simulating the interactions between particles in the media and photons with relatively low energy (visible light). This allows us to model interactions using two type of events, either a photon is absorbed (e.g. converted to heat) or it collides and scatters in another direction. In other fields, considering photons with higher energy, such as radiation dosimetry [1], more complex collision events, such as *Compton scattering* and *pair production*, have to be considered as well [19].

The optical properties of a participating medium are commonly described by its *absorption coefficient*, σ_a, and *scattering coefficient*, σ_s, which are both defined in inverse length (i.e. m^{-1}) units. They define the number of *e-foldings* occurring as a quantity of light passes through a unit length of the medium, meaning that light is attenuated by a factor of e over one unit of length for a medium with $\sigma_a = 1$. The inverse property, *mean free path* is easier to intuit: it specifies the mean distance between collision events for photons traveling through the medium. Media where $\sigma_a(x)$ and $\sigma_s(x)$ are constant for all x are referred to as *homogeneous*, otherwise, if the coefficients vary spatially, the media is *heterogeneous*. The sum of $\sigma_a(x)$ and $\sigma_s(x)$ define the *extinction coefficient*, σ_t, and the ratio σ_s/σ_t is referred to as the *albedo* of the medium.

The fraction of light that passes along a ray in a medium up to a distance d is given by the *transmittance* along the ray:

$$T(d) = \exp\left(-\int_0^d \sigma(t)dt\right), \qquad (1)$$

where $\sigma(t)$ is extinction coefficient and the integral in the exponent, $\int_0^d \sigma(t)dt$, is the *optical depth* (or *optical thickness*). In homogeneous media with a constant extinction coefficient σ the transmittance simplifies to $T(d) = \exp(-d\sigma)$. However, in heterogeneous media the transmittance is generally intractable. In practice, different forms of estimators are used to approximate the true transmittance instead of exact analytical solutions. Deriving efficient and robust transmittance estimators is one of the main challenges in simulating light/neutron transport in participating media, and is the focus of this paper.

A closely related problem to transmittance estimation is that of *free-flight sampling*, aiming at randomly sampling distances between consecutive interactions in the media. Specifically, the free-flight distance, s, between two consecutive interaction points in the media is given by the probability density:

$$p(s) = \sigma(s)\exp\left(-\int_0^s \sigma(t)dt\right) \qquad (2)$$

For homogeneous media this pdf can be sampled using the standard inversion method [18]. However, for heterogeneous media the inversion method is only tractable for special cases where analytical solutions to the optical depth are known [3].

2.1 Previous work

Below we categorize and survey existing techniques for estimating the transmittance in heterogeneous participating media.

Estimating the optical depth A simple approach to estimate the transmittance is to evaluate the integral in Equation 1, the optical depth, using standard numerical quadrature. This approach is often referred to as *ray marching* and implemented by traversing the ray using a fixed step size while assuming a constant $\sigma(t)$ between the queried points along the ray. However, even if an unbiased estimate is used to approximate the optical depth the exponential required to evaluate the transmittance Equation 1 makes the resulting transmittance estimator biased. A large number of samples can be used to more accurately approximate the optical depth, but this tends to produce inefficient estimators in practice.

Free-flight sampling estimators To sample free-flight distances in heterogeneous media an algorithm related to rejection sampling known as *delta tracking* can be used. This method is also sometimes referred to as Woodcock tracking, null collision tracking or pseudo tracking and was developed independently in neutron transport [25] and plasma physics [20]. The method depends on knowing an upper bound of the extinction coefficient along the ray, also known as the majorant extinction, $\sigma_u > \sigma_t$. The central idea is to introduce virtual, or fictitious, particles into the media corresponding to a extinction coefficient of σ_v along the ray. The density of virtual particles introduced along the ray is varied so that the total extinction coefficient from both real and virtual particles is constant, and equal to the majorant extinction, i.e. $\sigma_u = \sigma_t + \sigma_v$. The algorithm then advances along the ray by sampling free-flight distances in accordance to the majorant extinction, and decides randomly at each visited point whether the collision is with a real or virtual particle. Another free-flight distance is generated if the collision is with a virtual particle, otherwise the sampled distance is returned. A formal proof that this algorithm samples from the desired distribution can be found in [5]. The efficiency of this algorithm directly depends on how tight the upper bound of the extinction coefficient is. Several methods have been proposed to provide better bounds for both voxel-based and procedurally generated participating media using localized upper bounds. For example using super voxel grids, where a separate upper bound is stored in each super voxel [16], [23] or other spatial subdivision data-structures such as octrees and kd-trees [26] or Wrenninge's volume aggregates [7].

Free-flight sampling estimators can also be used for generating an unbiased estimate of the transmittance along a ray. By first generating M free-flight distances $s_i, i \in [1, M]$, the transmittance is estimated as the ratio of binary samples corresponding to $1 : s_i > d$ to $0 : s_i < d$, i.e. the average of a series of binary random variables. However, to provide low variance estimates it is often necessary to average a large number of binary samples, which can lead to long rendering times.

Rao-Blackwellized free-flight estimators To reduce the variance of delta tracking Novák et al. [16] proposed ratio tracking that can be seen as a Rao-Blackwellized version of delta tracking where the ray is traversed until the end, and the actual probability that the delta tracking procedure would reach all the way to the end is used as an estimate of the transmittance. They also considered using a control variate type of approach where a known, pre-computed, constant extinction can be subtracted from the extinction along the ray to further reduce the variance. An equivalent algorithm has also independently been derived in the neutron transport literature [15]. The residual ratio tracking algorithm is summarized in algorithm 1. See Novák et al. [16] for more details. Kutz et al. [12] used the same approach, but reformulated in a probabilistic framework and with a focus on reducing the cost of taking samples instead of the variance. Kutz et al. [12] also do not derive theoretical variance bounds as done in this work. Szirmay-Kalos et al. [22] also utilized the control variate, but removed the restriction of requiring an upper bound in their Rao-Blackwellized formulation. Similar to our work, Szirmay-Kalos et al. [22] derive variance bounds in their probabilistic mathematical formulation.

Piecewise integration with varying extinction majorants In production rendering, the extinction coefficient can vary widely throughout a scene. For example, both thin (but heterogeneous) fog and thick smoke can be present in the same image. Yue et al. [26] has showed that tracking-based estimators can be applied to sub-regions in a piecewise fashion while remaining unbiased. Recently, Villemin and Wrenninge [24] showed that the piecewise varying-majorant problem can be reinterpreted as a piecewise constant-majorant one. As a consequence, we only consider single-interval integration in this paper, knowing that it will generalize to more complex cases using the aforementioned techniques.

Our approach We propose to compute the transmittance using a new class of algorithms using an unbiased truncation of the Taylor expansion of the exponential function. Thus, unlike the majority of previous work basing their formulation on the introduction of virtual particles, we instead utilize standard Importance or Russian roulette Monte Carlo estimates of the infinite Taylor expansion series. In a recent independent work, Georgiev et al. [9] propose a new set of integral formulations for transmittance estimators, including a power series formulation. While some of the underlying ideas are similar to our work, we approach the problem from a different mathematical formulation allowing us to provide additional insights and mathematical proofs. We also show how these new mathematical formulations are related to the computational statistics literature, while Georgiev et al. 's mathematical approach is more similar to the classical transport theory literature. We propose several new transmittance estimators, and general proofs, not found in Georgiev et al. 's work, for example Algorithm 3, and 4. Using our new mathematical formulation and insights, we also provide several guidelines for parameter settings of different algorithms not found in previous work. In addition, we provide new experimental insights including algorithm behaviour related to CPU and GPU usage. Since our mathematical formulation allows the transmittance estimation to be seen from a new viewpoint not studied before in either computer graphics, transport theory or other related fields, we believe that it also has the potential for opening up the area for new ways of approaching the problem.

ALGORITHM 1: Residual ratio tracking (Novák et al. [16])

Input: Upper bound $\bar{\sigma} > \sigma(t)$, Control extinction σ_c

Set $t = 0$
Set $T_c = \exp(-d\sigma_c)$
Set $T_r = 1$
while *true* **do**
    Sample $r \sim \exp(r|-\bar{\sigma}d)$
    Set $t = t + r$
    if $t > d$ **then**
        break
    end
    $T_r = T_r(1 - (\sigma(t) - \sigma_c)/\bar{\sigma})$
end
Return $T_c T_r$

---

3 EXPRESSING TRANSMITTANCE USING TAYLOR EXPANSION

A key idea of this paper is the formulation of the transmittance using a Taylor expansion. Using this formulation, we then explore different methods to form Monte Carlo estimators of the Transmittance. Here, we introduce the basic mathematical formulation used as a basis to derive the different algorithms.

Inspired by previous work [16], we note that, given a constant, $\sigma_u < \infty$, we can formulate the transmittance as

$$
\begin{aligned}
T(d) &= \exp\left(-\int_0^d \sigma(t)dt\right) \\
&= \exp\left(-\int_0^d \sigma(t) + \sigma_u - \sigma_u dt\right) \\
&= \exp(-d\sigma_u)\exp\left(\int_0^d \sigma_u - \sigma(t)dt\right)
\end{aligned}
\tag{3}
$$

where σ_u could for example be chosen to be an upper bound of $\sigma(t)$. However, as we will discuss later, other choices are also possible.

By considering the Taylor expansion of the second exponential in Equation 3, the transmittance can be expressed as

$$
\begin{aligned}
T(d) &= \exp(-d\sigma_u)\exp(\tau(d)) \\
&= \exp(-d\sigma_u)\sum_{k=0}^{\infty}\frac{(\tau(d))^n}{n!}
\end{aligned}
\tag{4}
$$

where $\tau(d) = \int_0^d \sigma_u - \sigma(t)dt$.

Now assume we have access to an unbiased estimator of $\tau(d)$ given by $\hat{\tau}(d)$, i.e. $E[\hat{\tau}(d)] = \tau(d)$. In this paper we will consider $\hat{\tau}(d)$ to be a Monte Carlo integral estimator with a single sample,

$$
\hat{\tau} = d(\sigma_u - \sigma(\varepsilon)) \qquad \varepsilon \sim \mathscr{U}[0,d]
$$

where $\mathscr{U}[0,d]$ denotes the uniform distribution between 0 and d. We can then substitute all occurrences of $\tau(d)$ in Equation 4 by i.i.d estimators $\hat{\tau}_j(d)$, to form the estimator:

$$
\widehat{T(d)} = \exp(-d\sigma_u)\sum_{k=0}^{\infty}\frac{d^k}{k!}\prod_{j=1}^{k}\left(\sigma_u - \sigma(\varepsilon_j)\right)
\tag{5}
$$

where ε_j are i.i.d. uniform variables, i.e. $\varepsilon_j \sim \mathscr{U}[0,d]$. It is easy to show that this estimator is unbiased, i.e. $E[\widehat{T(d)}] = T(d)$.

4 FINITE UNBIASED EVALUATION OF INFINITE SERIES

A problem with the estimator given by Equation 5 is that it consists of an infinite number of terms, making a direct evaluation infeasible. However, we can approximate the infinite series with an unbiased estimator using only a finite number of terms. To this end, we will propose two different approaches.

First, in subsection 4.1, we derive a class of estimators based on forming an unbiased estimate of the transmittance by sampling a single term in the Taylor expansion using importance sampling. This approach is inspired by so called Poisson Estimators developed in the statistics literature [2]. Three different types of algorithms are proposed based on this Poisson formulation. The first type only require an upper bound of the extinction, the second type also require a *precomputed* control extinction but can lead to reduced variance estimators, while the third type instead utilizes *online computation* of control extinction computation, trading a small amount of bias for reduced variance. We also derive explicit variance results of these estimators, enabling insights into the different trade-offs between them, and guidelines in the selection of optimal parameters for different problem settings. We also show that the previously proposed Ratio Tracking and Residual Ratio Tracking algorithms [16] can be seen as special cases of this class of estimators.

The second class of algorithms we consider are based on using Russian Roulette [4], [13] to truncate the infinite series. This class of estimators have different theoretical properties compared to the first class of estimators as discussed in subsection 4.2. For this class, we present an algorithm adapting samples according to the underlying extinction. This algorithm belongs to the first type, i.e. only requiring an upper bound of the extinction. We also discuss how this class of estimators is related to the previously proposed delta tracking algorithm [16], and how they differ.

Lastly, we briefly discuss the benefits of partitioning the transmittance integral into shorter segments.

4.1 Single Term Estimators

Single term estimators are based on importance sampling over the terms in the Taylor expansion. Consider first sampling an auxiliary variable $k \in [0...\infty]$ with probability $q(k)$. Using importance sampling, to approximate the sum in $T(d)$ (Equation 5), an unbiased estimator of $T(d)$ can then be formed by

$$\widehat{T}_s(d) = \exp(-d\sigma_u) \frac{d^k}{q(k)k!} \prod_{j=1}^{k} (\sigma_u - \sigma(\varepsilon_j)), \qquad (6)$$

where ε_j are i.i.d. uniform variables, i.e. $\varepsilon_j \sim \mathcal{U}[0,d]$. The variance of this estimator is given by

$$Var[\widehat{T}_s(d)] = \exp(-2d\sigma_u) \sum_{k=0}^{\infty} \left(\frac{d^k}{q(k)(k!)^2} \left(\int_0^d (\sigma_u - \sigma(t))^2 dt \right)^k \right) - T(d)^2$$

See appendix A for mathematical details and proofs.

The choice of the importance distribution $q(k)$ can have a big effect on the variance of the estimator. By minimizing the variance with respect to $q(k)$, one can show that the optimal distribution is a Poisson distribution $Po(\lambda^*)$ with intensity

$$\lambda^* = \left(d \int_0^d (\sigma_u - \sigma(t))^2 dt \right)^{1/2}. \qquad (7)$$

Again, see appendix A for mathematical details and proofs. Given that we now know the optimal choice of $q(k)$, we can set out

to design practical algorithms imitating importance distribution close to the known optimal Poisson distribution. In the following subsections, we describe four different variants of such algorithms, and discuss how their parameters can be tuned. The basic algorithm that the other algorithms are based on is the Poisson Estimator, that we describe in detail first. Note that each algorithm has different trade-offs that we describe in detail below.

4.1.1 Poisson Estimator

While the optimal intensity λ^* depends on an intractable integral, a conservative approach is to use $\lambda_U = d\sigma_u$. This choice is motivated by the fact that it is an upper bound, i.e. $\lambda_U \geq \lambda^*$, which ensures that the importance distribution has longer tails than the target distribution, a desirable property of importance sampling estimators [17]. Using a Poisson distribution with intensity λ_U as a proposal distribution for k in Equation 6 results in the following estimator of the transmittance, $T(d)$,

$$\boxed{\widehat{T}_p = \prod_{j=1}^{k} \left(1 - \frac{\sigma(\varepsilon_j)}{\sigma_u} \right) \quad k \sim Poisson(k|\lambda_U),} \qquad (8)$$

where ε_j are i.i.d. uniform variables, i.e. $\varepsilon_j \sim \mathcal{U}[0,d]$. This, *Poisson estimator* is summarized in algorithm 2 with $\sigma_c = 0$.

By applying Equation 7, we can find an explicit expression for the variance of the Poisson tracking estimator also including ratio tracking:

$$Var[\widehat{T}_p] = T(d)^2 \left(\exp \left(\int_0^d \frac{\sigma(t)^2}{\sigma_u} dt \right) - 1 \right) \qquad (9)$$

Note that the proposed Poisson estimator is more general than the Ratio Tracking algorithm as any algorithm for sampling Poisson distributed interaction points along the ray could be used, see appendix A.1 for proof. This opens up for completely different and more effective practical sampling algorithms than sequentially sampling of exponentials as used in ratio tracking. Exploring this type of new sampling algorithms are beyond the scope of this paper, and we leave this for future work.

Practical guidelines for σ_u - From the variance expression above it is clear that a higher upper bound σ_u always decreases the variance, but it also increases the cost of the algorithm. Figure 1
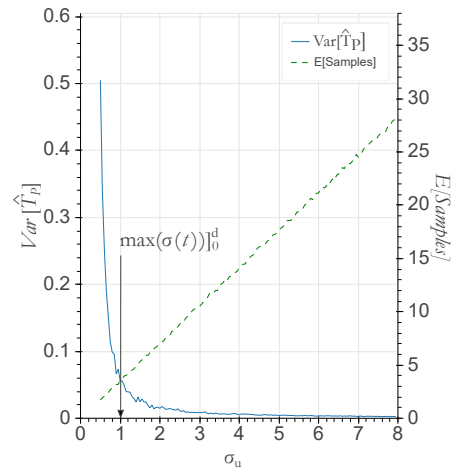

Fig. 1: Experimental test of the variance (left axis) and number of samples taken (right axis) as a function of varying σ_u for the Poisson tracking algorithm. The variance decreases rapidly until $\sigma_u \approx 3[max(\sigma(t))]_0^d$, while the number of samples increase linearly.

ALGORITHM 2: Residual Poisson estimator

Input : Upper bound $\sigma_u \geq \sigma(t)$, Control extinction σ_c

Set $T_c = \exp(-d\sigma_c)$
Sample $k \sim \mathscr{P}ois(d(\sigma_u - \sigma_c))$
Sample $\varepsilon_j \sim \mathscr{U}[0,d]$, for $j = 1..k$
Set $T_r = \prod_{i=1}^{k} 1 - (\sigma(\varepsilon_i) - \sigma_c)/(\sigma_u - \sigma_c)$
Return $T_c T_r$

---

shows the typical behavior of the estimated variance and number of samples taken for the Poisson estimator when varying σ_u for the second test function described in subsection 5.1. Figure 1 shows that $\sigma_u \approx 3\max_{[0,d]}(\sigma(t))]$ provides a good trade-off between variance reduction and computational load. Interestingly, in contrast to free-flight sampling estimators, this means that having a tight and exact upper bound, i.e. $\sigma_u = \max_{[0,d]}(\sigma(t))]$ is not important.

Another way to reduce the variance of the estimated transmittance is to take the average of several Poisson estimators with a lower upper bound $\sigma_{u_N} = \frac{\sigma_u}{N}$. Averaging N Poisson estimators with σ_{u_N} has the same expected cost as a single Poisson estimator with the upper bound σ_u. A natural question in practice is then which choice is more efficient? In Theorem 5, in the Appendix, we show that it is always better to use a single Poisson estimator with upper bound σ_U compared to using N average Poisson estimators with upper bound σ_{U_N}.

4.1.2 Residual Poisson Estimator

In some problems, it is possible to approximate the extinction along the ray where the Transmittance is to be estimated, for example using specialized data structures. In these cases it is possible to reduce the variance of the estimated transmittance by using an idea related to control variates, based on subtracting a known *control extinction*, σ_c, from the true extinction, $\sigma(t)$, along the ray [16]. In this section we describe how this idea can be used to extend the Poisson estimator presented in the last subsection.

First note that the equation for the transmittance along the ray can be rewritten, Equation 1, as

$$T(d) = \exp\left(-\int_0^d \sigma(t)dt\right)$$
$$= \exp(-d\sigma_c)\exp\left(-\int_0^d \sigma_r(t)dt\right),$$

where $\sigma_r(t) = \sigma(t) - \sigma_c$ is the *residual extinction* coefficient, and $\exp(-d\sigma_c)$ is a known constant.

We can now directly target the residual transmittance, $T_r(d) = \exp\left(-\int_0^d \sigma_r(t)dt\right)$ using a Poisson estimator with an intensity based on an upper bound on the residual extinction, i.e. $\sigma_u > \sigma_r(t) = \sigma(t) - \sigma_c$, as

$$\widehat{T}_{rp}(d) = \exp(-d\sigma_c)\prod_{j=1}^{k}\left(1 - \frac{\sigma_r(\varepsilon_j)}{\sigma_u}\right) \quad k \sim Poisson(k|\lambda_U).$$

We refer to this algorithm as the *Residual Poisson estimator*, see algorithm 2. We note that a similar estimator can also be derived mathematically by considering the Taylor expansion of the exponential around the point at distance $-d\sigma_c$ along the ray. The original Poisson estimator is obtained as a special case of the residual Poisson estimator by setting σ_c to zero. Furthermore, Novák et al. proposed to compute the residual transmittance using ratio tracking algorithm under the name Residual Ratio

Tracking [16], which can also be shown to be a special case of the Residual Poisson estimator (see Theorem 3 in the Appendix).

To compare the original Poisson estimator to the Residual Poisson estimator we can consider it's variance,

$$Var[\widehat{T}_{rp}(d)] = T(d)^2\left(\exp\left(\int_0^d \frac{\sigma_r(t)^2}{\sigma_u}dt\right) - 1\right). \quad (10)$$

Note that the variance expression above is similar to Equation 9, but with the residual extinction term $\sigma_r(t)$ exchanged for the extinction $\sigma(t)$. Thus, the previous guidelines for σ_u hold here as well. In addition, it can be seen that having a good control extinction is even more important given its squared term. Furthermore, we can now derive the following upper bound for the variance,

$$Var[\widehat{T}_{rp}] \leq T(d)^2\left(\exp\left(\frac{d(\max[\sigma(t)]_0^d - \sigma_c)^2}{\sigma_u}\right) - 1\right). \quad (11)$$

See appendix A.2 for proof. The bound provide us with an important hint when it comes to understanding how to choose σ_c, because the highest variance is given by the largest difference between the actual extinction along the path and the control extinction.

Practical guidelines for σ_c - By analyzing the expression for the variance given by Equation 10, we find that for a given upper bound σ_u the variance is minimized by setting σ_c to the mean of σ_t in the interval $[0,d]$. This is stated in the following collary:

Collary 1. *Let σ_u be fixed, the optimal σ_c, in terms of minimal variance, is then given by*

$$\sigma_c = \frac{1}{d}\int_0^d \sigma(t)dt. \quad (12)$$

4.1.3 Independent and Dependent Poisson Estimators

Inspired by Corollary 1, we can approximate the true mean by the simple estimate $\sigma_c = \frac{1}{N}\sum_{j=1}^{k}\sigma(u_i)$, where u_i are N i.i.d. uniform random variables over $[0,d]$. This requires us to sample $\sigma(t)$ N more times, but often provides a decrease in variance compared to a standard Poisson estimator. As we demonstrate in the experimental section, this can be a beneficial trade-off in practice. We refer to this estimator as the *Independent Poisson estimator* and its algorithm is summarized in algorithm 3. Another important aspect of this algorithm is that it does not require precomputation of the control extinction and therefore also does not require additional costly data structures to compute the control extinction.

Another, related idea, is to reuse the random uniform look-ups already performed in the Poisson tracking estimator, i.e. to reuse the N number of ε_j random variables already sampled to also estimate σ_c. In practical experiments we have found that this is often a useful tracking estimator, although it suffers from a small bias. Due to its dependency between samples, we refer to algorithm 4 as the *Dependent Poisson estimator*.

4.1.4 Poisson Distribution Alternatives

As the optimal intensity λ^* is generally intractable, a Poisson distribution might not necessarily be the best possible choice. Other natural choices in this context are the geometric distribution and the negative binomial distribution.

Geometric Distribution:

A simple choice is to set $q(k)$ to a geometric distribution, with parameter p in Equation 6, resulting in the estimator:

$$\widehat{T}_s(d) = \exp(-d\sigma_u)\frac{d^k}{(1-p)^k pk!}\prod_{j=1}^{k}\left(\sigma_u - \sigma(\varepsilon_j)\right), \quad (13)$$

ALGORITHM 3: Residual Poisson estimator with independently estimated control extinction

Input : Upper bound $\sigma_u \geq \sigma(t)$
Number of independent samples N

Sample $u_i \sim \mathscr{U}[0,d]$, for $i = 1..N$
Set $\sigma_c = \frac{1}{N} \sum_{i=1}^{k} \sigma(\varepsilon_i)$
Sample $k \sim \mathscr{P}ois(d(\sigma_u - \sigma_c))$
Sample $\varepsilon_j \sim \mathscr{U}[0,d]$, for $j = 1..k$
Set $T_c = \exp(-d\sigma_c)$
Set $T_r = \prod_{i=1}^{k} 1 - (\sigma(\varepsilon_i) - \sigma_c)/(\sigma_u - \sigma_c)$
Return $T_c T_r$

ALGORITHM 4: Residual Poisson estimator with dependent estimate of control extinction

Input : Upper bound $\sigma_u \geq \sigma(t)$

Sample $N \sim \mathscr{P}ois(d\sigma_u)$
Sample $\varepsilon_j \sim \mathscr{U}[0,d]$, for $j = 1..N$
Set $\sigma_c = \frac{1}{N} \sum_{i=1}^{N} \sigma(\varepsilon_i)$
Set $T_c = \exp(-d\sigma_c)$
Sample $k \sim \mathscr{P}ois(d(\sigma_u - \sigma_c))$
Sample $\varepsilon_j \sim \mathscr{U}[0,d]$, for $j = N..k$
Set $T_r = \prod_{i=1}^{k} 1 - (\sigma(\varepsilon_i) - \sigma_c)/(\sigma_u - \sigma_c)$
Return $T_c T_r$

---

where ε_j are i.i.d. uniform variables, i.e. $\varepsilon_j \sim \mathscr{U}[0,d]$.

Negative Binomial:

Another approach is to use a Poisson distribution but model λ with a Gamma distribution with mean λ^*, the resulting distribution $q(k)$ will then be a Gamma-Poisson distribution, also known as a negative Binomial distribution. This approach is explored in the context of stochastic differential equations by [6].

However, we have not seen any benefits from considering these proposal distributions in our practical experiments, so we leave further analysis to future work.

4.2 Russian Roulette

Ray tracing and particle transport applications extensively use Russian Roulette for terminating random walks with small contributions. Here, we describe how it can be used to truncate the infinite series in Equation 5. Our method works by replacing each term in the sum in Equation 5 with the unbiased estimator

$$\widehat{T(d)}_{RR}^{k} = \begin{cases} \frac{\exp(-d\sigma_u)\frac{d^k}{k!} \prod_{j=1}^{k}\left(\sigma_u-\sigma(\varepsilon_j)\right)}{\prod_{i=1}^{k}(1-q_i)} & \text{if } U_i > q_i \text{ for } i = 1...k \\ 0 & \text{otherwise} \end{cases}$$

$$(14)$$

where U_i are i.i.d. uniformly distributed random variables, and $q_1, q_2, ...q_i \in (0,1]$. The combined estimator is then given by

$$\widehat{T(d)}_{RR} = T(d)^0 + \sum_{k=1}^{\infty} T(d)_R^k. \qquad (15)$$

A simple choice is to set $q_i = q$ with $q \in (0,1]$, allowing for trade-offs between variance and number of samples taken.

A better choice is to adapt q to the the signal at hand. By taking inspiration from delta tracking we consider $q_i = \frac{\sigma(\varepsilon_i)}{\sigma_u}$, providing


ALGORITHM 5: Russian Roulette Tracking

Input : Upper bound $\sigma_u \geq \sigma(t)$

Set $T_r = 1$ (First term in Equation 5)
Set $k = 1$
Sample $\varepsilon_1 \sim \mathscr{U}[0,d]$
Set $q = \frac{\sigma(\varepsilon_1)}{\sigma_u}$
Set $q_k = q$
Sample $\eta \sim \mathscr{U}[0,1]$
While($\eta > q$)
    Set $T_r = T_r + \frac{1}{q_k} \frac{\sigma_u^k d^k}{k!}$
    Sample $\varepsilon_i \sim \mathscr{U}[0,d]$
    Set $q = \frac{\sigma(\varepsilon_i)}{\sigma_u}$
    Set $q_k = q_{k-1}q$
    Set $k = k+1$
    Sample $\eta \sim \mathscr{U}[0,1]$
end
Return $\exp(-d\sigma_u)T_r$

---

us with an estimator

$$\widehat{T(d)}_{RR}^{k} = \begin{cases} \frac{\exp(-d\sigma_u)\frac{d^k}{k!} \prod_{j=1}^{k}\left(\sigma_u-\sigma(\varepsilon_j)\right)}{\prod_{i=1}^{k}\left(1-\frac{\sigma(\varepsilon_i)}{\sigma_u}\right)} \\ = \exp(-d\sigma_u)\frac{\sigma_u^k d^k}{k!} & \text{if } U_i > q_i \text{ for } i = 1...k \\ 0 & \text{otherwise} \end{cases}$$

$$(16)$$

This Russian roulette tracking algorithm is summarized in algorithm 5. Note that the resulting algorithm is similar to the estimator of the transmittance based on delta tracking proposed in previous work (Section 2.3 in [16]). However, the delta tracking based estimator samples points sequentially from a homogeneous Poisson process, i.e. it generates a k and $\varepsilon_j : s$ such that $\varepsilon_1 < \varepsilon_2, ..., < \varepsilon_n$, but here we do not need to keep track of the order of the uniform variables in the interval.
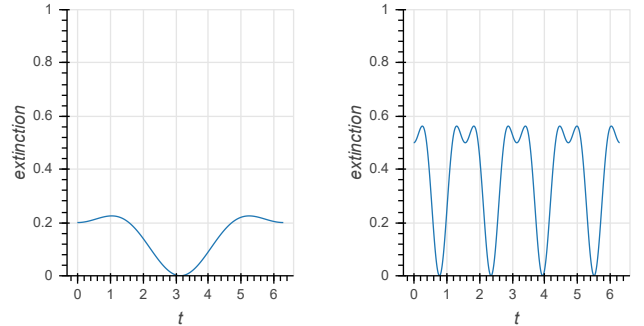
(a) Test function A,
$\alpha = 0.1$, $\beta = 1.0$.
Reference $T(d) = 0.3897$.

(b) Test function B,
$\alpha = 0.25$, $\beta = 4.0$.
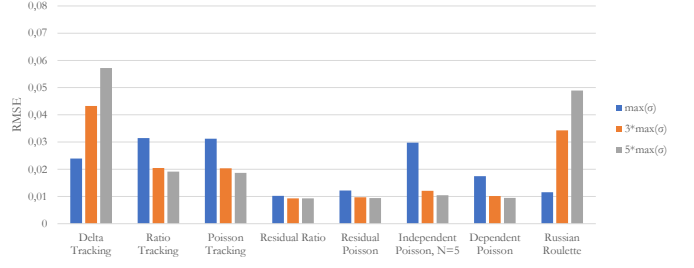Reference $T(d) = 0.094788$.

Fig. 2: Plots of the two test functions used for numerical illustration, where (a) represents more slowly varying functions while (b) represents more high frequency functions.

4.3 Partitioning Methods

The terms in the Taylor expansion, see Equation 5, will decay at a slower rate when the integral $\left(\int_0^d \sigma_u - \sigma(t)dt \right)$ is large compared
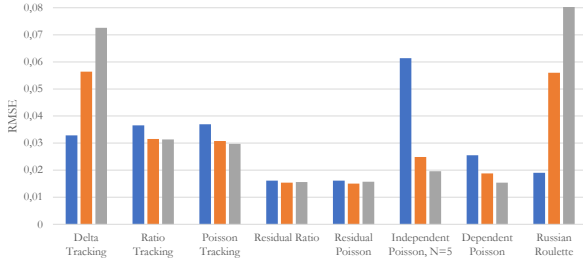
(a) Test function A

(b) Test function B

Fig. 3: Transmittance RMSE when taking equal number of samples for tight upper bound (blue bars) and loose upper bounds (orange/grey bars). Poisson/Ratio methods improve with higher upper bound, with independent/dependent Poisson having the greatest benefit .

to when the integral is small. One way to keep this quantity small is to partition the interval $[0, D]$ into M parts

$$T(d) = \exp(-d\sigma_u)\exp\left(\int_0^d \sigma_u - \sigma(t)dt\right)$$
$$= \prod_{m=0}^{M-1} \exp(-d\sigma_u)\exp\left(\int_{md/M}^{(m+1)d/M} \sigma_u - \sigma(t)dt\right) \quad (17)$$

Now, we can apply the Taylor expansions on each of the exponentials in this sum, similar to the above analysis. As $\int_0^{d/m} \sigma_u - \sigma(t)dt \leq \int_0^d \sigma_u - \sigma(t)dt$, the terms in the Taylor expansion will decay faster, and we thus need to evaluate less terms in the Taylor expansion using Russian Roulette for a given accuracy. This technique can be used together with any of the tracking methods and we therefore refer to it as *Partitioned Tracking*.

5 RESULTS

To evaluate how the methods behave under different practical circumstances, we first present the results of applying the methods to a known test function. Then, we show how the methods perform when applying them in real-world heterogeneous medium data sets, taking into account their evaluation time.

5.1 Numerical Illustration

To demonstrate the numerical performance of the different methods we will consider the test function

$$\sigma(t) = \alpha(\sin(\beta x)^2 + \cos(\beta x) + 1) \quad (18)$$

over the interval $t \in [0, 2\pi]$. Here, α controls the height and β controls the frequency of the function. Two sets of α and β values will be used, see Figure 2, one representing more slowly varying extinction coefficient and one representing higher frequency variations of the extinction coefficient.

The performance of the methods in the numerical illustration are measured using the root mean squared error (RMSE) when restricting the number of evaluations of $\sigma(t)$ to a fixed number (200). Each method is evaluated 1000 times and the average RMSE is reported. The control extinction is set to the average of the minimum and maximum extinction, $\sigma_c = 0.5(\max_{[0,d]}(\sigma(t)) + \min_{[0,d]}(\sigma(t)))$, unless otherwise specified.

5.1.1 Upper Bound

To analyze the impact of choosing the upper bound, we consider three possible settings: $\sigma_u = N\max_{[0,d]}(\sigma(t)), N \in [1,3,5]$.
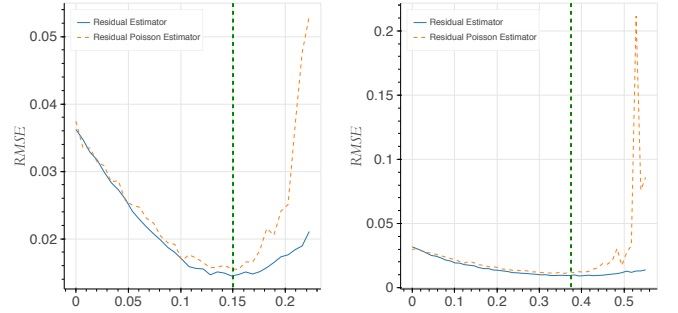
(a) Test function A

(b) Test function B

Fig. 4: Transmittance RMSE for varying control extinction, $\sigma_c \in [\min_{[0,d]}(\sigma(t))\ \max_{[0,d]}(\sigma(t))]$, and equal number of samples. Vertical dotted line depicts the average extinction. As theoretically shown, lowest error is achieved when σ_c equals the average extinction. The Poisson estimator is more prone to error as σ_c approaches the upper bound.

In Figure 3 it can be seen that increasing the upper bound lowers the RMSE for the dependent and independent Poisson methods. This is connected with the accuracy of the estimated control extinction made by these two methods, increasing the upper bound will cause more samples to be drawn from the Poisson distribution (higher k), which result in lower variance. The residual transmittance and residual Poisson transmittance estimators have a fixed control transmittance, which make them more sensitive to changes in control extinction than upper bound. The Russian roulette transmittance estimator, similar to delta tracking, benefits from having a tight upper bound.

5.1.2 Control Extinction Comparison

The residual transmittance estimator methods use control extinction to improve their performance. To understand and verify the impact of having a good control extinction estimate we varied the control extinction while keeping the upper bound and number of samples fixed. The results of this numerical experiment are inline with our theoretical results, c.f. Equation 12, and can be seen in Figure 4, where lowest errors are produced when the control extinction equals the average extinction. Interestingly, it can also be seen that the Poisson method tend to produce higher errors when σ_c approaches σ_u, i.e. low numbers of k.
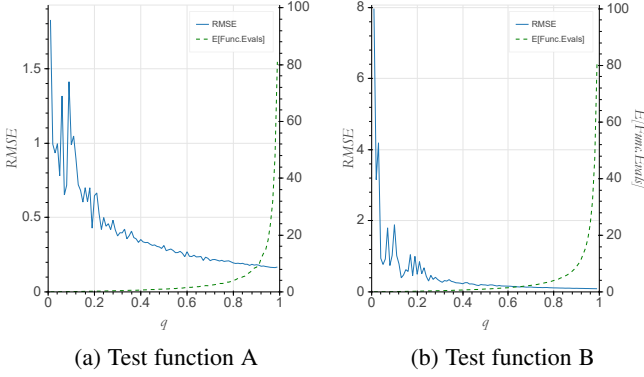
Fig. 5: Transmittance RMSE (left vertical axis) and average number of samples taken (right vertical axis) when varying q for Russian roulette transmittance estimator (Alg. algorithm 5). The RMSE decreases at the cost of increasing the number of samples, with $q > 0.8$ resulting in comparably many samples.

5.1.3 Varying q in Russian Roulette Tracking

The q variable in the Russian roulette tracking algorithm provides a trade-off between variance and number of samples taken. This connection is illustrated in Figure 5, where q varies along the horizontal axis. Here, RMSE is decreasing for increasing q, while the number of samples taken are increasing. It can be seen that $q < 0.2$ results in noisy behaviour for the test functions and that the number of samples increase significantly when $q > 0.8$. We noted that numerical instabilities occur when $k > 119$ for 64-bit floating point and $k > 16$ for 32-bit floating point numbers. We therefore limit the number of iterations depending on the precision.

5.2 Real-World Examples

The presented methods have been integrated into an OpenCL-based path tracer in the Inviwo [10] visualization framework running on both the CPU and the GPU. The path tracer implementation utilizes partitioned tracking through a super-grid (8^3 voxels per grid cell) storing the extinction bounds and average of each grid cell, see [16] and [11] for more details. An Intel Xeon 2.5 GHz CPU with 32 GB random access memory and a Geforce 1070 GPU is uses as testing hardware. Reference images have been generated using delta tracking-based path tracing using 50 000 iterations. The following three data sets, with increasing extinction coefficient difficulty level, have been used for evaluation.

Bunny cloud (Figure 9) has $577{\times}572{\times}438$ 32-bit (float) voxels. A directional light source has been placed to the left and the extinction is set to zero below values of 0.05 and mapped to an extinction of 22.5 for values above. Thus, this data set represents homogeneous mediums with changes only occurring when entering or leaving the extinction boundary between zero and 22.5.

Smoke2 (Figure 10) has $161{\times}610{\times}178$ 32-bit (float) voxels and represent relatively smoothly varying heterogeneous mediums. A directional light source has been placed behind the data set and the values are mapped to extinction by multiplication with 150.

Golden Lady (Figure 11) is a computed tomography scan of the upper torso and head of a human. It has $512{\times}512{\times}625$ 16-bit (uint) voxels. The voxel values have been mapped to display vessels and bone tissues and thereby represent heterogeneous mediums with high extinction coefficients.

First, we evaluate the impact of N and σ_u in algorithm 3. These results are depicted in Figure 6, where it can be seen that it is more important to use a high upper bound ($\sigma_u \geq 3 \max_{[0,d]}(\sigma(t))$) than improving the control estimate by taking additional samples, i.e., increasing N. For the GPU, the more complex the medium is, the more beneficial it is to spend computation time on improving the control estimate by increasing N. Not surprisingly, a single sample is enough for homogeneous mediums such as the Bunny. Comparing the behaviour between the CPU and the GPU, it can be seen that the results vary linearly with N on the CPU, while they are almost constant on the GPU. This difference is most likely caused by the the texture cache on the GPU, which significantly reduces the cost of taking additional nearby samples. Similarly, when observing the impact of N and σ_u for algorithm 4 in Figure 7, it can be seen that the CPU results benefit significantly from re-using samples, while it actually has a slightly negative impact on the GPU. We believe that this negative impact is caused by the additional registers required to explicitly cache the samples, preventing efficient parallelization, in combination with the low sampling cost of nearby samples due to the GPU texture cache. Notably, in Figure 7a it can be seen that the heterogeneous data sets (Smoke2, Golden Lady) benefit more from improving the control estimate using higher N for algorithm 4.

The impact of bounding k for algorithm 5 (Russian Roulette) can be seen in Figure 8. While limiting k can bias the results, it also can improve performance and we see from the orange bars in the figure that good choices of k lie around 6-8 for the CPU and 9-11 for the GPU.

Based on the analysis above, we show equal rendering time results using the best combination for each method in Figures 9-11. We refer to the supplementary material for the specific parameter combinations used and an analysis of the upper bound parameter for the Ratio, Residual Ratio and Residual Poisson methods. Table 1 lists the RMSE of each method and data set compared to their corresponding reference image. Our proposed independent and dependent Poisson transmittance estimation algorithms, algorithm 3 and algorithm 4, perform best for the more difficult data sets with heterogeneous materials using the GPU and CPU, respectively. The Residual ratio and Poisson transmittance methods perform best for the data sets with more slowly varying or constant extinction.

TABLE 1: Equal time (25 s) rendering RMSE (lower is better) of each method and data set using the CPU and the GPU. Best result for each data set and computing platform is highlighted in bold. Note that algorithms 3-5 do not require precomputed control estimates σ_c.

Method	Data set (RMSE) CPU / GPU		
	Bunny	Smoke2	Golden lady
Delta Tracking	1513 / 301.2	280 / 50.1	1994 / 317
Ratio Tracking [16]	947 / 197.5	181 / 46.1	1440 / 231
Residual Ratio [16] (Alg. 1)	856 / **185.7**	175 / **44.8**	1401 / 230
Residual Poisson (Alg. 2)	**847** / 186.4	**173** / 46.2	1380 / 228
Ind. Poisson (Alg. 3)	899 / 186.4	174 / 47.1	1397 / **223**
Dep. Poisson (Alg. 4)	908 / 189.7	174 / 48.0	**1351** / 234
Russian Roulette (Alg. 5)	895 / 187.4	201 / 51.8	1476 / 245

6 DISCUSSION

The results show that the different algorithms often result in similar performance for mediums with small changes in the extinction coefficients, e.g., Bunny and Smoke2. For these types of mediums Residual Poisson perform slighly better on the CPU, while Residual
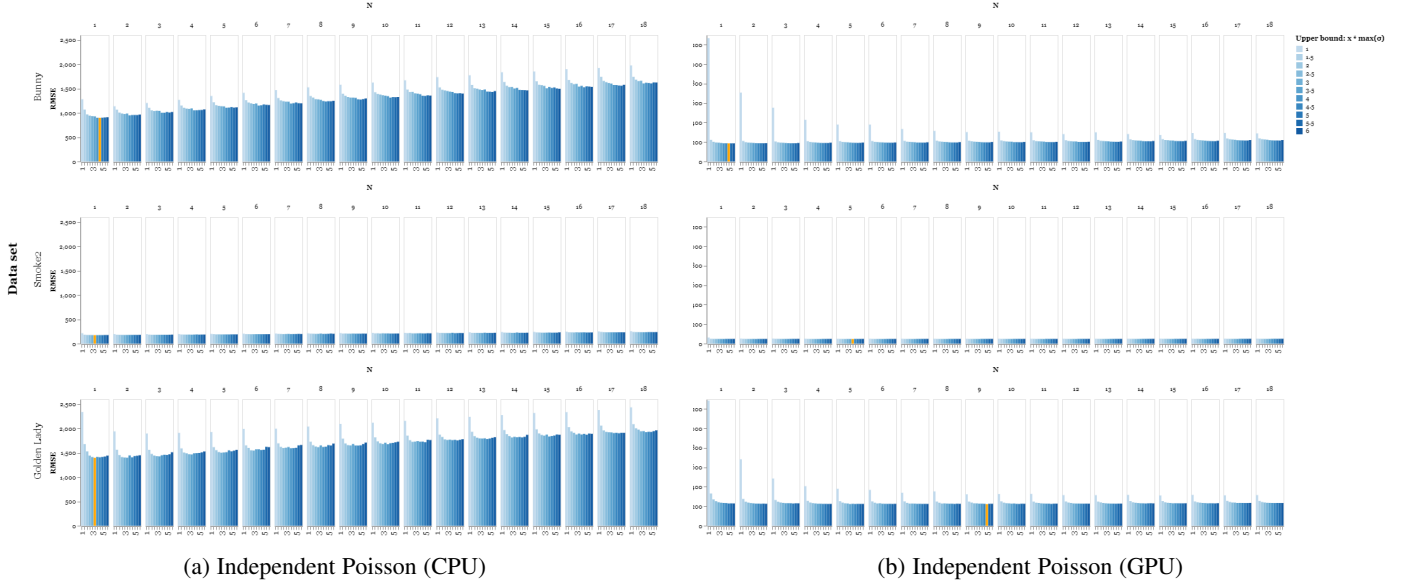
(a) Independent Poisson (CPU) (b) Independent Poisson (GPU)

Fig. 6: Equal time renderings (25 s) for varying N and σ_u in algorithm 3 using (a) the CPU and (b) the GPU. The combination resulting in best RMSE is highlighted in orange for each data set. Independent of N it is important to have loose upper bounds, e.g. $\sigma_u \geq 3 \max_{[0,d]}(\sigma(t))]$. For GPU:s, larger N are beneficial for heterogeneous mediums (Smoke2 and Golden lady).
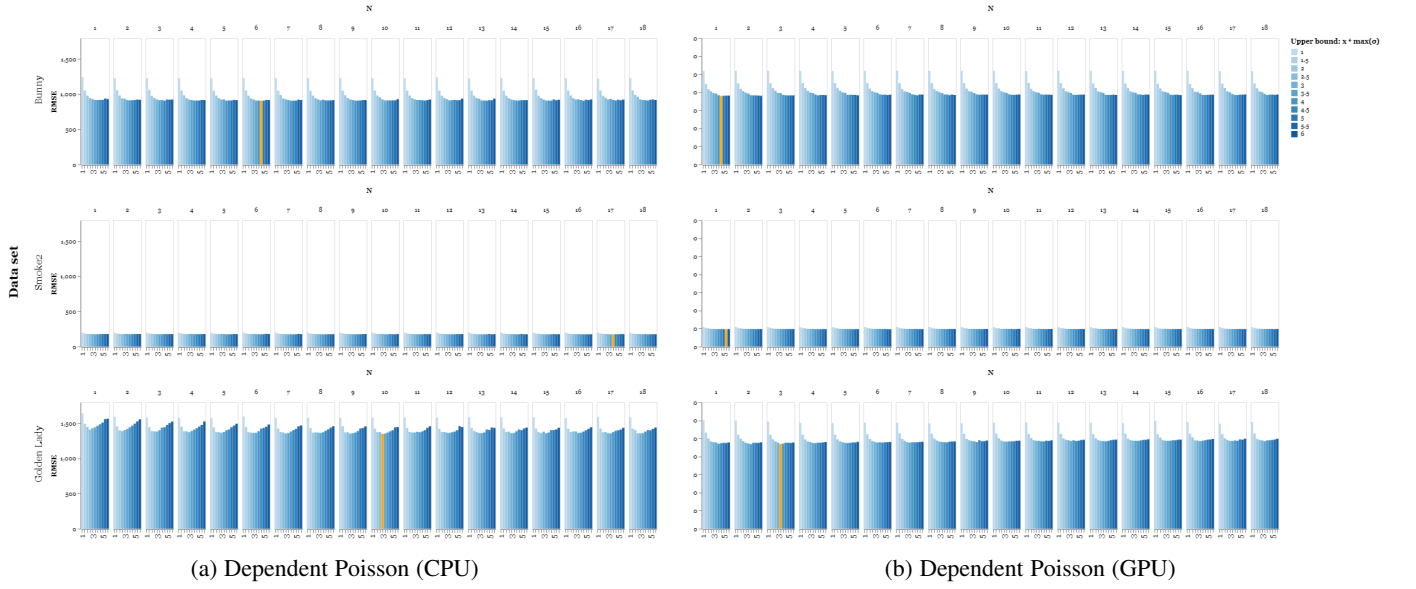

(a) Dependent Poisson (CPU) (b) Dependent Poisson (GPU)

Fig. 7: Equal time renderings (25 s) for varying N and σ_u in algorithm 4 using (a) the CPU and (b) the GPU. The combination resulting in best RMSE is highlighted in orange for each data set. Reusing samples is highly beneficial on the CPU, because the computation time otherwise increase approximately linearly with increasing N. Larger N produces better control transmittance, which is beneficial for heterogeneous mediums (Smoke2, Golden lady).

Ratio perform slightly better on the GPU. For the more challenging heterogeneous mediums, e.g., Golden Lady, the Independent and Dependent Poisson methods perform best on the CPU and GPU, respectively. We note that because the Ratio and Residual Ratio tracking algorithms are subsets of the Poisson tracking algorithms, their theoretical behaviours are similar and the main differences lie in their algorithmic design. With this being said, the presented methods have a range of trade-offs and characteristics to consider. As shown in Figure 4, the residual methods are largely dependent on having good estimates of the control extinction. Increasing the upper bound has a larger effect on the independent and

dependent Poisson methods compared to the residual ratio and Poisson methods. This is particularly prominent in heterogeneous media with extinction peaks, see left most bar for each N in Figure 6, causing large RMSE. We believe that the reason for this is that their control extinction estimate is on average worse than the pre-calculated average extinction provided to the other two methods. This reasoning is in line with the results in Figure 6 and Figure 7, which shows that the control estimate is less important for higher upper bounds (the RMSE is dominated by the choice of upper bound). Still, especially for heterogeneous media with extinction peaks, e.g. Golden lady, it is beneficial to use independent
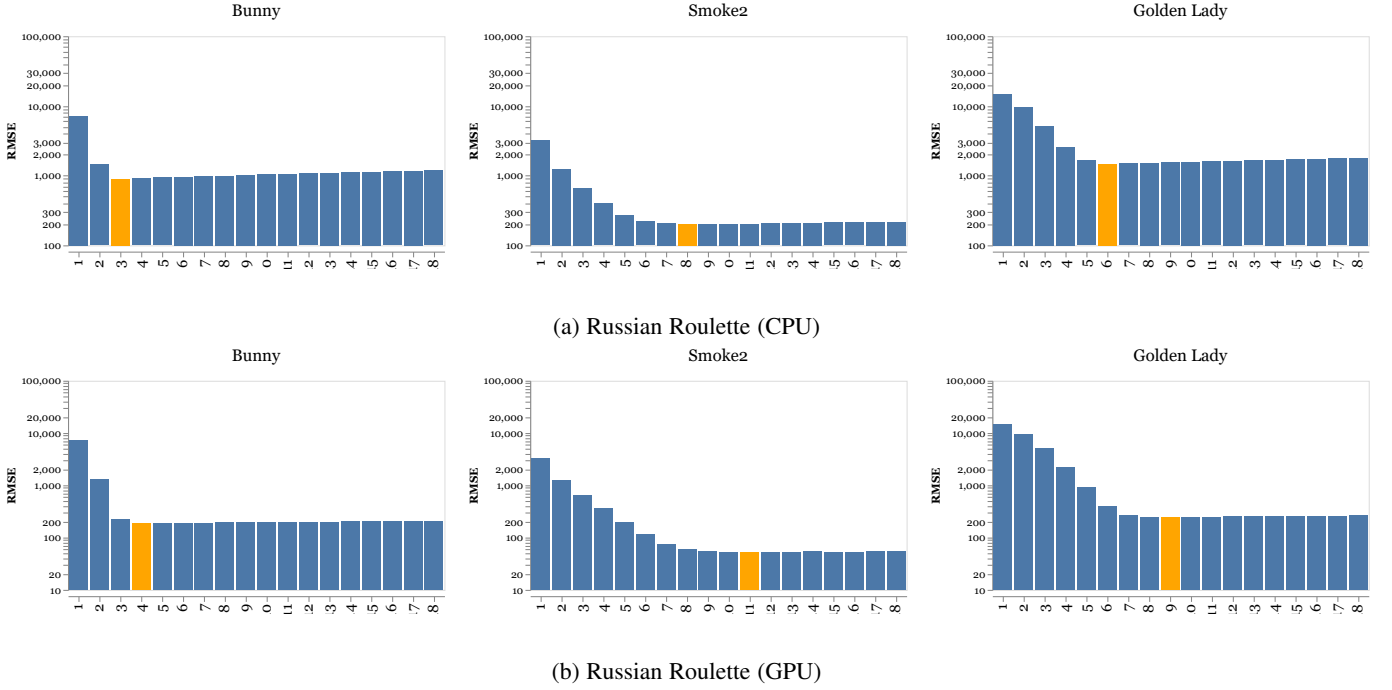
Fig. 8: Equal time (25 s) rendering $log(RMSE)$ for varying upper limit of k in algorithm 5. The k resulting in best RMSE is highlighted in orange for each data set. While lower k can result in a slight bias it can be beneficial from a performance perspective and good choices of k in this case lie around 6-8 for the CPU and 9-11 for the GPU.
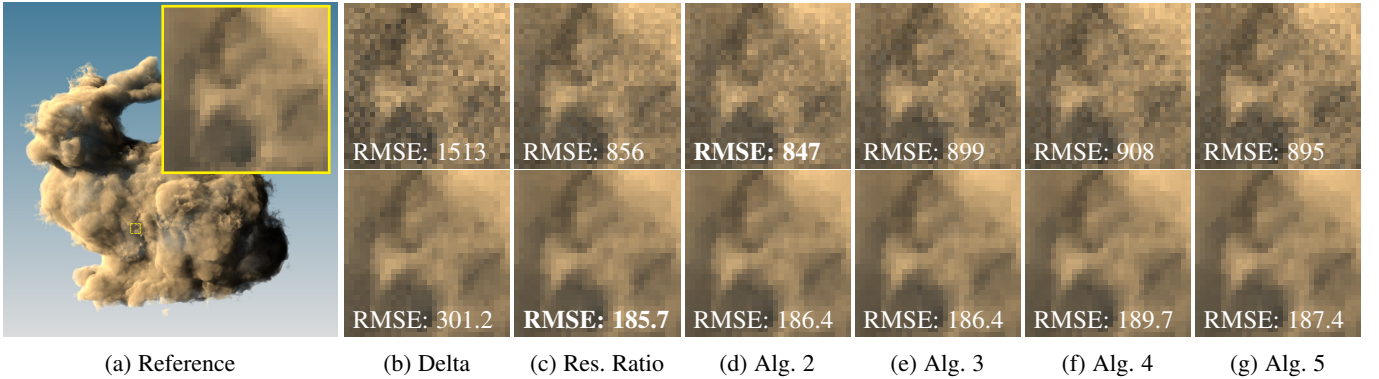

Fig. 9: Equal rendering time (25 s) of each method applied to a homogeneous medium using the CPU (top row) and the GPU (bottom row). Data set is courtesy of OpenVDB.

estimates of the control extinction when combined with a higher upper bound. Another benefit of using the independent/dependent Poisson methods is that there is no need to determine the control extinction beforehand, which improves the performance when changing the extinction coefficient to explore the data.

Our assumption was that the dependent Poisson transmittance estimator would perform better than the independent Poisson method in practice, since it reuses samples. However, this assumption only held using the CPU. It seems as the graphics card's internal caching is performing well enough to remove benefits from reusing samples. In addition, it might be that the slight bias of reusing samples is revealed for the constant medium, i.e., Bunny, because algorithm 4 has higher RMSE for the CPU in this case.

6.1 σ_u-only methods

Comparing results from taking the same number of samples for the methods not relying on control extinction in Figure 3, e.g., Delta tracking [20], [25] Ratio tracking [16], and Russian Roulette transmittance estimation, it can be seen that the proposed Russian Roulette method outperforms the other methods. However, when comparing equal-time rendering results in Table 1, it can be seen that the Ratio tracking algorithm is best for both the Smoke2 and the Golden lady data sets. This is most likely caused by the low cost of taking samples, i.e. the extra numerical computation required by the Russian roulette algorithm outweighs the time it takes to sample the volume. For out-of-core methods, where the sampling cost may be higher it could potentially be more beneficial to use the Russian roulette method.

Note that numerical aspects need to be considered when implementing the Russian roulette method. We found that limiting
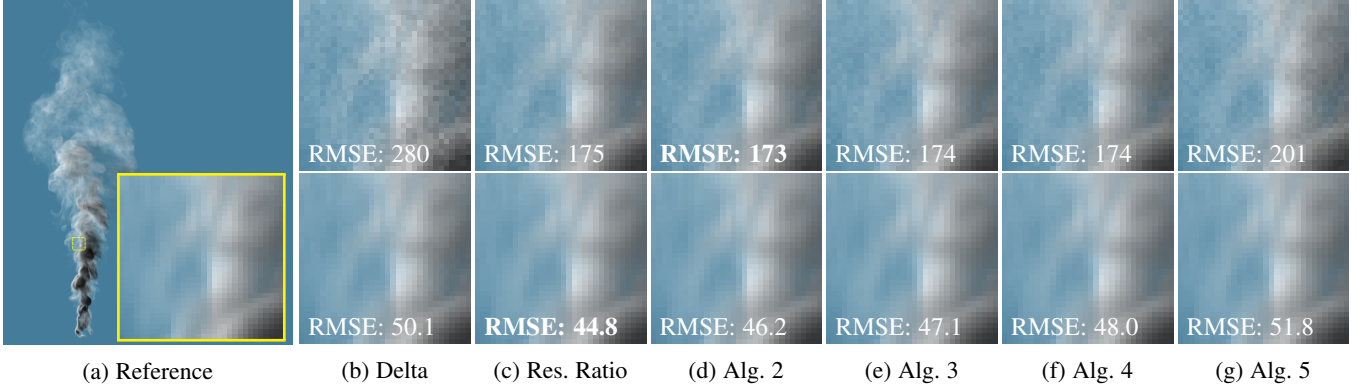
Fig. 10: Equal rendering time (25 s) of each method applied to the smoothly varying Smoke2 data set using the CPU (top row) and the GPU (bottom row). Data set is courtesy of OpenVDB.
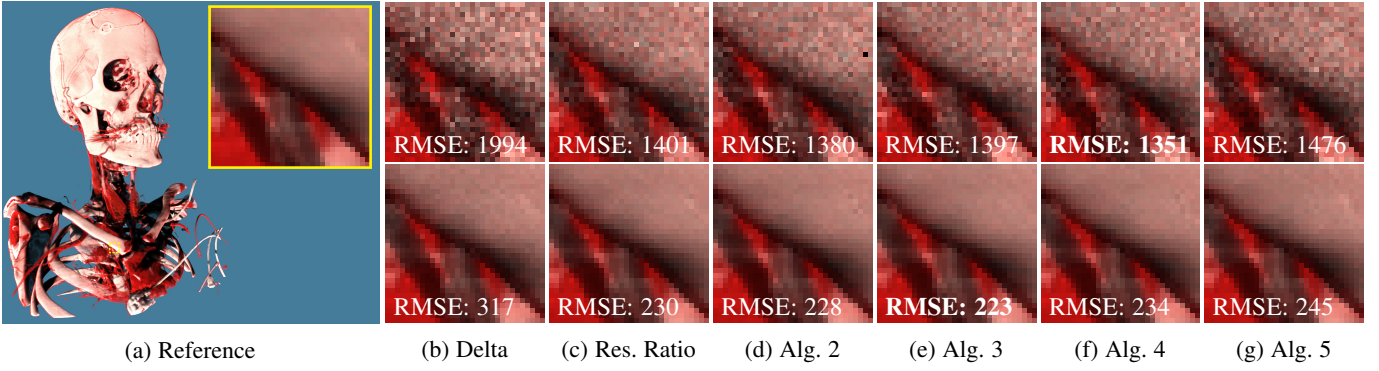

Fig. 11: Equal rendering time (25 s) of each method applied to the heterogeneous Golden lady data set using the CPU (top row) and the GPU (bottom row). The benefit of reusing samples become evident on the CPU (f), whereas on the GPU it actually better to let rely on the internal caching (e). Data set is courtesy of the Center for Medical Image Science and Visualization (CMIV), Linköping University.

the number of iterations worked well in this regard. On the graphics card used for the experiments, the Russian roulette method becomes three times slower when using double precision compared to 32-bit floating point precision, which consequently increases the RMSE for equal time renderings. Thus, given the setup used in this work, it is better to limit k than to switch to double precision. Furthermore, as seen in Figure 8, limiting k even further can improve the equal time rendering results at the expense of slightly biasing the result.

Finally, we have shown that the control extinction, σ_c, and Russian roulette q variables are key components in achieving good results. An interesting area for future work is therefore to develop more advanced methods for estimating these variables.

7 CONCLUSIONS AND FUTURE WORK

In this paper we explored a new mathematical formulation for estimating the transmittance of heterogeneous participating media. To our knowledge, our approach and mathematical formulation of using unbiased or biased techniques to approximate the Taylor expansion of the exponential has not been explored before for transmittance estimation. Although our work has some similarities to the independent, contemporary, work of Georgiev et al. [9], our mathematical formulation of the problem is fundamentally different, and provides additional theoretical and practical insights into the problem, for example, in the form of several new theorems and proofs. In particular, based on our formulation, we also introduced two new practical rendering algorithms, Independent

and Dependent Poisson (Alg. 3 & 4) that perform best in many applications, e.g. for medical volume rendering. Also, from the comparison studies between CPU and GPU implementations, we can conclude that transmittance estimation algorithms performing well on the CPU does not necessarily translate to the GPU, and provided several practical guidelines for GPU implementations. In future work it would be interesting to explore the combination of importance sampling and Russian roulette approaches, using for example multiple importance sampling (MIS). Another topic left for future work is to investigate the use of the debasing techniques introduced in computational statistics, see e.g. [14], utilizing a convergent sequence/levels of biased estimators, for example obtained using standard quadrature techniques for the inner integral in Equation 1 with an increasing number of samples in the quadrature for each level.

REFERENCES

[1] P. Andreo. Monte carlo techniques in medical radiation physics. *Physics in Medicine & Biology*, 36(7):861, 1991.

[2] A. Beskos, O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):333–382, 2006.

[3] F. B. Brown and W. R. Martin. Direct sampling of Monte Carlo flight paths in media with continuously varying cross-sections. In *Proc. ANS Mathematics & Computation Topical Meeting*, 4 2003.

[4] L. L. Carter and E. D. Cashwell. Particle-transport simulation with the monte carlo method. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1975.

[5] W. Coleman. Mathematical verification of a certain monte carlo sampling technique in radiation transport problems. *Nuclear Science and Engineering*, 32(1):76–81, 1968.

[6] P. Fearnhead, O. Papaspiliopoulos, and G. O. Roberts. Particle filters for partially observed diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):755–777, 2008.

[7] J. Fong, M. Wrenninge, C. Kulla, and R. Habel. Production volume rendering: Siggraph 2017 course. In *ACM SIGGRAPH 2017 Courses*, p. 2. ACM, 2017.

[8] R. G. Gallager. *Stochastic processes: theory for applications*. Cambridge University Press, 2013.

[9] I. Georgiev, Z. Misso, T. Hachisuka, D. Nowrouzezahrai, J. Křivánek, and W. Jarosz. Integral formulations of volumetric transmittance. *ACM Transactions on Graphics (TOG)*, 38(6):154, 2019.

[10] D. Jönsson, P. Steneteg, E. Sundén, R. Englund, S. Kottravel, M. Falk, A. Ynnerman, I. Hotz, and T. Ropinski. Inviwo - A Visualization System with Usage Abstraction Levels. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2019.

[11] D. Jönsson and A. Ynnerman. Correlated Photon Mapping for Interactive Global Illumination of Time-Varying Volumetric Data. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 23(1):901–910, 2017.

[12] P. Kutz, R. Habel, Y. K. Li, and J. Novák. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics (TOG)*, 36(4):1–16, 2017.

[13] A.-M. Lyne, M. Girolami, Y. Atachadé, H. Strathmann, and D. Simpson. On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods. *Statistical Science*, 30(4):443–467, 2015.

[14] D. McLeish. A general method for debiasing a Monte Carlo estimator. *arXiv preprint arXiv:1005.2228*, 2010.

[15] L. Morgan and D. Kotlyar. Weighted-delta-tracking for Monte Carlo particle transport. *Annals of Nuclear Energy*, 85:1184 – 1188, 2015. doi: 10.1016/j.anucene.2015.07.038

[16] J. Novák, A. Selle, and W. Jarosz. Residual ratio tracking for estimating attenuation in participating media. *ACM TOG (SIGGRAPH Asia)*, 33(6):179, 2014.

[17] A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[18] M. Pharr, W. Jakob, and G. Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[19] F. Salvat, J. M. Fernández-Varea, and J. Sempau. Penelope-2006: A code system for monte carlo simulation of electron and photon transport. In *Workshop proceedings*, vol. 4, p. 7. Nuclear Energy Agency, Organization for Economic Co-operation and Development Barcelona, Spain, 2006.

[20] H. Skullerud. The stochastic computer simulation of ion motion in a gas subjected to a constant electric field. *Journal of Physics D: Applied Physics*, 1(11):1567, 1968.

[21] J. Spanier and E. M. Gelbard. *Monte Carlo principles and neutron transport problems*. Dover Publications, 31 East 2nd Street, Mineola, N.Y. 11501, USA, 2 2008.

[22] L. Szirmay-Kalos, I. Georgiev, M. Magdics, B. Molnár, and D. Légrády. Unbiased light transport estimators for inhomogeneous participating media. In *Computer Graphics Forum*, vol. 36, pp. 9–19. Wiley Online Library, 2017.

[23] L. Szirmay-Kalos, B. Tóth, and M. Magdics. Free path sampling in high resolution inhomogeneous participating media. In *Computer Graphics Forum*, vol. 30, pp. 85–97. Wiley Online Library, 2011.

[24] R. Villemin, M. Wrenninge, J. Fong, and P. A. Studios. Efficient unbiased rendering of thin participating media. *Journal of Computer Graphics Techniques (JCGT)*, 2018.

[25] E. Woodcock, T. Murphy, P. Hemmings, and S. Longworth. 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*, pp. 557–579, 1965.

[26] Y. Yue, K. Iwasaki, B.-Y. Chen, Y. Dobashi, and T. Nishita. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Transactions on Graphics (TOG)*, 29(6):177, 2010.

Daniel Jönsson received the MSc and PhD degrees in media technology in 2009 and 2016 from Linköping University, Sweden. Since 2017, he is a research fellow at the division for Media and Information Technology (MIT) at the Department of Science and Technology (ITN), Linköping University. His current main research interests lie in the intersection between visualization and artificial intelligence.

Joel Kronander is currently machine learning engineering manager at Nines. He received the MS degree in applied physics and electrical engineering in 2009 and PhD degree in 2015 from Linköping University, Sweden. His research interests include machine learning, volume rendering, deep learning and medical imaging.

Jonas Unger is a professor in computer graphics at Linköping University, Sweden. Unger received MSc and PhD degrees in media technology from Linköping University in 2003 and 2009 respectively, became a docent in 2015, and has been a visiting researcher at the University of Southern California. Since 2009, he is leading the Computer Graphics and Image Processing group at Linköping University with a focus on research and development of new theory and technology for computational imaging; fusing computer graphics, vision and sensors with human perception and machine learning. With a strong foundation in theoretically oriented research, Unger and his group are working closely with industry to develop state-of-the-art methods for 3D scene reconstruction, photo-realistic image synthesis and digitization of optical material properties, heart surgery capturing, perceptual displays, and generation of data for autonomous systems in applications domains such as self driving vehicles and medical imaging.

Thomas B. Schön is the Beijer Professor of Artificial Intelligence with the Department of Information Technology at Uppsala University. He received the BSc degree in Business Administration and Economics in Jan. 2001, the MSc degree in Applied Physics and Electrical Engineering in Sep. 2001, and the PhD degree in Automatic Control in Feb. 2006, all from Linköping University. He has held visiting positions with the University of Cambridge (UK), the University of Newcastle (Australia) and Universidad Técnica Federico Santa María (Valparaíso, Chile). In 2018, he was elected to The Royal Swedish Academy of Engineering Sciences (IVA) and The Royal Society of Sciences at Uppsala. He received the Tage Erlander prize for natural sciences and technology in 2017 and the Arnberg prize in 2016, both awarded by the Royal Swedish Academy of Sciences (KVA). He was awarded the Automatica Best Paper Prize in 2014, and in 2013 he received the best PhD thesis award by The European Association for Signal Processing (EURASIP). He received the best teacher award at the Institute of Technology, Linköping University in 2009. Schön has a broad interest in developing new algorithms and mathematical models capable of learning from data. He is a Senior member of the IEEE.

Magnus Wrenninge is a Technical Lead at Aurora. He started his career in the visual effects industry at Digital Domain where he created fluid simulation and environment rendering software, and later worked in R&D, he also held the roles as a Lead Technical Direction at Sony Pictures Imageworks, and as a Principal Engineer at Pixar Animation Studios. In 2015 he was awarded with an Academy Technical Achievement Award for the development of Field3D, and he currently serves on the Academy of Motion Pictures' Scientific and Technical Awards committee.

Supplementary material for
Direct Transmittance Estimation in Heterogeneous Participating Media Using Approximated Taylor Expansions

Joel Kronander, Daniel Jönsson, Jonas Unger, *Member, IEEE,* Thomas B. Schön, *Senior Member, IEEE,* and Magnus Wrenninge

✦

1 INTRODUCTION

This supplementary material presents additional results for the upper bound σ_u (Figure 1) along with parameters used for the real-world data set experiments (Table 1). Please refer to the paper for information on the methods, experimental setup, and data sets.

TABLE 1: Parameters used for each method and combination of CPU/GPU. For methods with σ_u, we use a multiplier based on the maximum extinction coefficient in the grid cell.

Method	Data set (RMSE) CPU / GPU		
	Bunny	Smoke2	Golden lady
Delta Tracking	- / -	- / -	- / -
Ratio Tracking: σ_u	5.0 / 6.0	6.0 / 2.5	3.5 / 5.5
Residual Ratio (Alg. 1): σ_u	3.0 / 6.0	2.0 / 2.5	3.0 / 5.5
Residual Poisson (Alg. 2): σ_u	4.0 / 5.5	5.0 / 5.0	3.5 / 5.0
Ind. Poisson (Alg. 3): σ_u, N	4.5, 1 / 5.0, 1	3.5, 1 / 4.0, 5	3.5, 1 / 5.0, 9
Dep. Poisson (Alg. 4): σ_u, N	4.5, 6 / 4.5, 1	4.0, 17 / 5.5, 1	3.0, 10 / 3.5, 3
Russian Roulette (Alg. 5): k	3 / 4	8 / 11	6 / 9

- *J. Kronander, D. Jönsson, J. Unger are with Linköing University, Sweden. E-mail: joelkronander@gmail.com, {daniel.jonsson, jonas.unger}@liu.se.*
- *T. Schön is with Uppsala University, Sweden. E-mail: thomas.schon@it.uu.se.*
- *M. Wrenninge is with Pixar Animation Studios, USA. E-mail: magnus@pixar.com.*
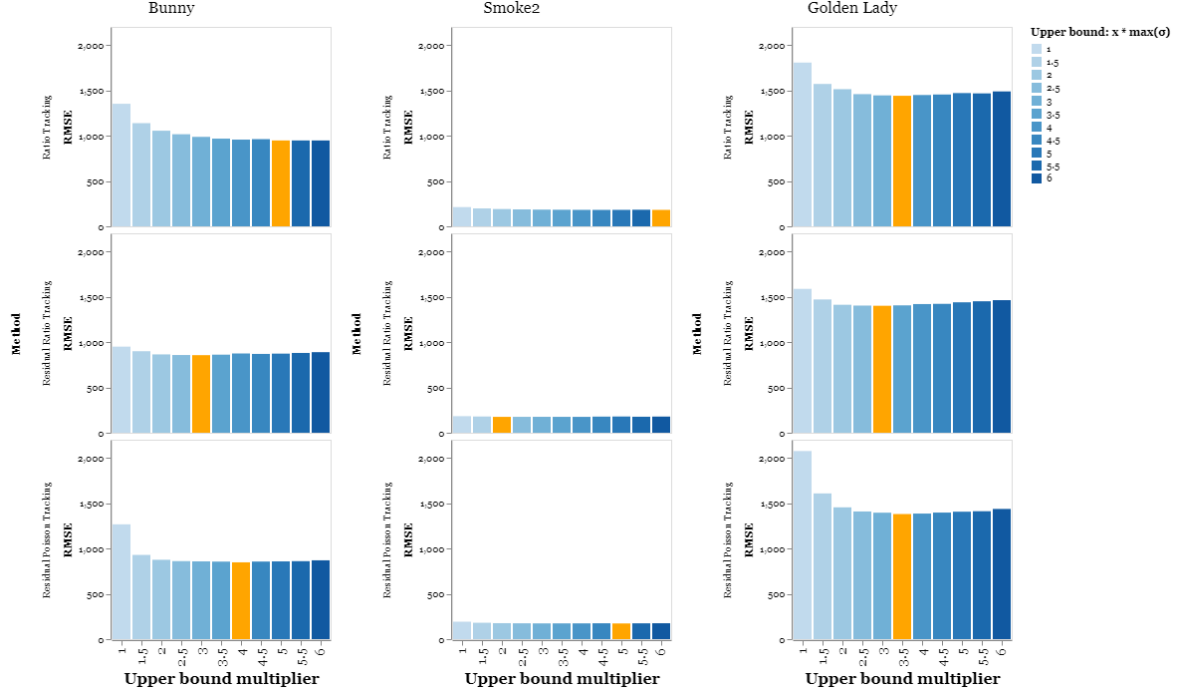
(a) CPU
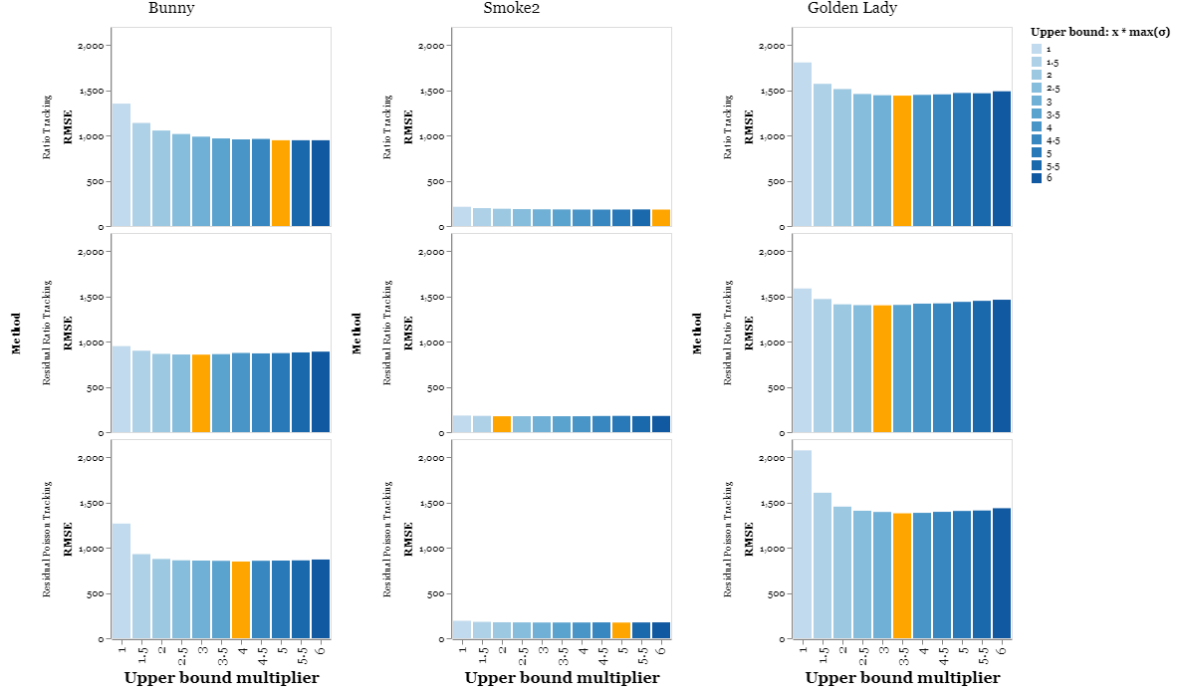
(b) GPU

Fig. 1: Equal time renderings (25 s) for varying the upper bound σ_u for the Ratio, Residual Ratio, and Residual Poisson methods. The upper bound multiplier resulting in best RMSE is highlighted in orange for each data set and method.
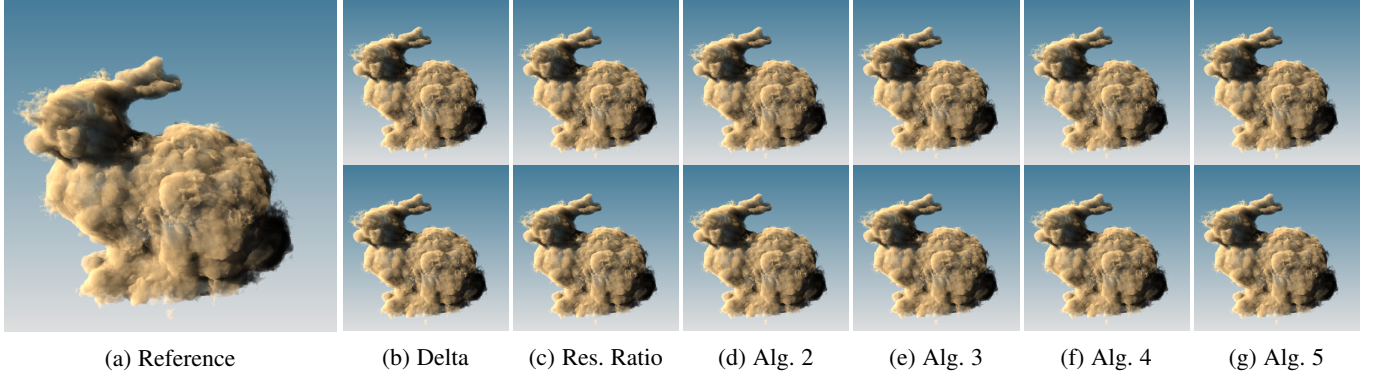
Fig. 2: Equal rendering time (25 s) of each method applied to a homogeneous medium using the CPU (top row) and the GPU (bottom row). Zoom for details. Data set is courtesy of OpenVDB.
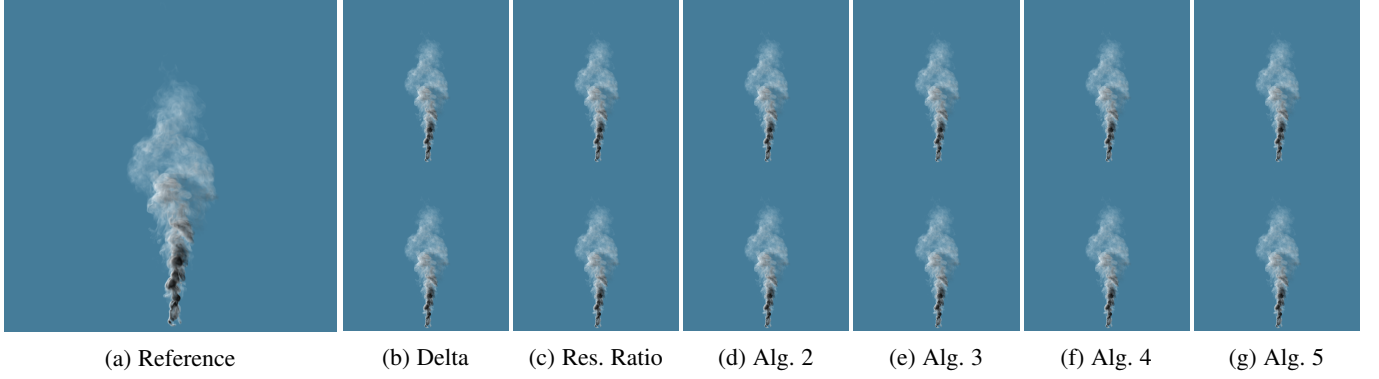

Fig. 3: Equal rendering time (25 s) of each method applied to the smoothly varying Smoke2 data set using the CPU (top row) and the GPU (bottom row). Zoom for details. Data set is courtesy of OpenVDB.
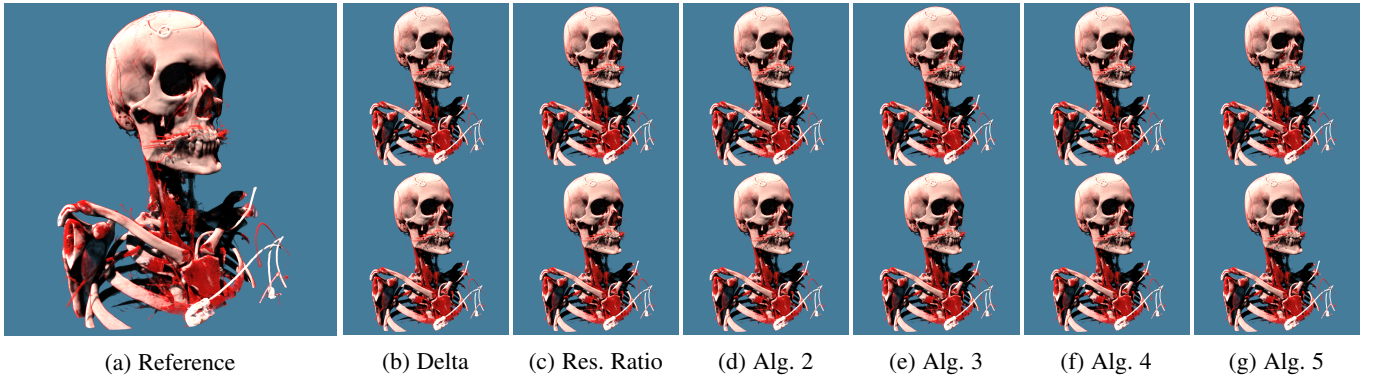

Fig. 4: Equal rendering time (25 s) of each method applied to the heterogeneous Golden lady data set using the CPU (top row) and the GPU (bottom row). Zoom for details. Data set is courtesy of the Center for Medical Image Science and Visualization (CMIV), Linköping University.