Accelerated Monte Carlo Rendering of Finite-Time Lyapunov Exponents

Irene Baeza Rojo, Markus Gross and Tobias Günther

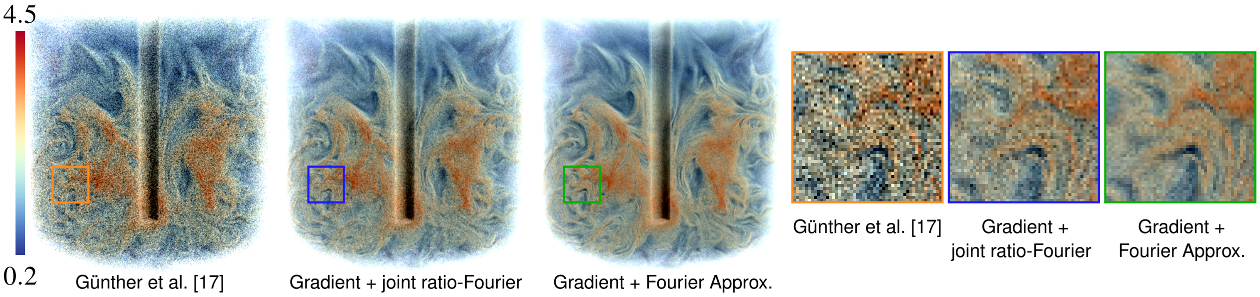

4.5
0.2
Günther et al. [17]
Gradient + joint ratio-Fourier
Gradient + Fourier Approx.
Günther et al. [17]
Gradient + joint ratio-Fourier
Gradient + Fourier Approx.

Fig. 1: We propose two orthogonal strategies to accelerate the rendering of Monte Carlo-based visualizations of finite-time Lyapunov exponent fields: (1) gradient domain rendering for single-scattered participating medium and (2) an acceleration of the light transmittance estimator. For the latter, we propose (2a) a joint estimator that combines ratio tracking [38] with a pre-computed Fourier approximation of the transmittance and (2b) a pure Fourier approximation of the transmittance. The above visualizations compare the method of Günther et al. [17] with our two new approaches after rendering for 75 seconds for the MIXER data set.

Abstract— Time-dependent fluid flows often contain numerous hyperbolic Lagrangian coherent structures, which act as transport barriers that guide the advection. The finite-time Lyapunov exponent is a commonly-used approximation to locate these repelling or attracting structures. Especially on large numerical simulations, the FTLE ridges can become arbitrarily sharp and very complex. Thus, the discrete sampling onto a grid for a subsequent direct volume rendering is likely to miss sharp ridges in the visualization. For this reason, an unbiased Monte Carlo-based rendering approach was recently proposed that treats the FTLE field as participating medium with single scattering. This method constructs a ground truth rendering without discretization, but it is prohibitively slow with render times in the order of days or weeks for a single image. In this paper, we accelerate the rendering process significantly, which allows us to compute video sequence of high-resolution FTLE animations in a much more reasonable time frame. For this, we follow two orthogonal approaches to improve on the rendering process: the volumetric light path integration in gradient domain and an acceleration of the transmittance estimation. We analyze the convergence and performance of the proposed method and demonstrate the approach by rendering complex FTLE fields in several 3D vector fields.

Index Terms—Scientific visualization, Monte Carlo, feature extraction, finite-time Lyapunov exponents, gradient domain, Fourier

1 INTRODUCTION

For the analysis of large and complex time-dependent vector fields, feature extraction is among the most promising approaches [42, 43], as it allows the user to focus on the relevant aspects of the data. The extraction of features from time-dependent vector fields is still a challenging and open research problem, both in terms of the definition of suitable features as well as in their efficient extraction [18]. Hyperbolic Lagrangian coherent structures [20] are among the most commonly-observed features for the analysis of the Lagrangian transport behavior of time-dependent vector fields. A common approximation is the finite-time Lyapunov exponent (FTLE) [19, 48], which measures the separation of nearby-released particles by estimating the so-called flow map gradient. Ridges in this scalar field represent transport barriers that give order and structure to the flow and thus FTLE fields found numerous applications and improvements [3, 13, 16, 26, 47, 54]. Since FTLE is a Lagrangian approach, its computation is rather expensive, as it requires the tracing of multiple particle trajectories. Aside from the computational effort involved in the determination of the FTLE value and its ridges, another great challenge arises: FTLE ridges may become arbitrarily thin for long integration durations. Shadden et al. [48] have shown that well-defined ridges minimize the flux across the ridge better than poorly defined ridges, and thus sharp ridges are of most interest when analyzing material separation with FTLE fields. The rendering of those sharp ridges is challenging, since traditional ray casting methods suffer two problems. First, the discretization onto a regular grid for subsequent direct volume rendering can result in visually noticeable discretization errors and color biases. Second, the step size discretization during ray marching can also cause heavy aliasing. The artifacts of ray casting-based approaches have previously been shown by Günther et al. [17]. Methods have been proposed to adaptively approximate the flow map gradient in a hierarchical data structure [3, 47]. The main use case was interactive exploration and thus quality was sacrificed for performance [3, 4]. If an accurate solution is required, these methods are unbound in their memory consumption. In order to obtain a high-quality visualization that avoids grid discretization and ray marching errors, a conceptually different visualization approach is required, which led Günther et al. [17] to the use of a Monte Carlo rendering approach. They formulated the FTLE rendering as a single scattering light transport problem in a participating medium, which is well-studied in the context of neutron transport [55], plasma physics [49], recently in computer graphics for offline light transport in movie production [33, 38, 51–53] and for scientific visualization [29]. The benefits of Monte Carlo methods over ray marching have been demonstrated for instance by Yue et al. [56]. Günther et al. [17] estimated the free-flight distance of photons to determine scattering locations as well as the transmittance inside the medium by using a technique called delta tracking [55]. This technique is *unbiased*, which means that, given a sequence of n measurements M_i with $i \in \{1, ..., n\}$ that try to match the unknown quantity Q, the

- *Irene Baeza Rojo and Tobias Günther are with the Computer Graphics Laboratory, ETH Zürich. E-mail: irene.baeza|tobias.guenther@inf.ethz.ch.*
- *Markus Gross is head of the Computer Graphics Laboratory, ETH Zürich. E-mail: grossm@inf.ethz.ch.*

expected value of the error $M_i - Q$ is zero. The computation of this Monte Carlo rendering approach is rather slow, as the rendering of a single noise-free image may require multiple days.

The goal of our paper is to accelerate Monte Carlo FTLE rendering in order to make it applicable to the GPU-based offline rendering of animations, including camera animation, slicing through the data and time animations. In principle, there are two strategies to accelerate the computation: either by a flow map approximation (which introduces bias) or by improving the radiance integration. In this paper, we accelerate the convergence of the rendering side using two different strategies. First, we extend gradient domain rendering [27,34] to single scattering in a participating medium. Gradient domain rendering is a recent denoising technique that estimates not only pixel colors but also their screen-space derivatives. Both are combined through an optimization to reconstruct the image that best explains the estimated colors and gradients. When reconstructing under the L_2 error norm, gradient domain rendering is unbiased. Second, we investigate accelerations of the transmittance estimate from the light source. Thereby, we consider a wide spectrum from the unbiased ratio tracking of Novák et al. [38] to the approximation of the transmittance using a continuous version of the Fourier opacity maps of Jansen and Bavoil [24]. Based on the two above, we propose a joint transmittance estimator that starts with ratio tracking and heuristically switches to a Fourier approximation to speed-up the calculation and to reduce variance. In summary, we propose three technical contributions to the Monte Carlo FTLE rendering:

1. An extension of gradient domain rendering to single scattering.
2. A pure Fourier approximation of the light transmittance, which is fast but potentially introduces large shading bias.
3. A joint transmittance estimator that combines ratio tracking with a Fourier approximation, which heuristically bounds the error.

The resulting algorithms allow us to render high quality video sequences of Monte Carlo FTLE visualizations achieving the same quality error on a single GPU about 8× faster than previous work. Fig. 1 demonstrates the techniques after 75 seconds at a small preview resolution of 350×364 pixels. While the visualization of Günther et al. [17] is still noisy, our methods show detailed ridges.

2 Related Work

This paper combines concepts from two different research areas, namely scientific visualization and physically-based volumetric light transport. In the following, we discuss the relevant concepts in more detail.

2.1 Finite-Time Lyapunov Exponents

Lagrangian Coherent Structures. In flow visualization, feature extraction methods can be divided into Eulerian and Lagrangian techniques [18]. The Eulerian techniques consider a single time slice only, which gives a transient impression of the flow behavior at a certain moment in time. Lagrangian techniques, on the other hand, observe particles over time and thereby consider the advection of the flow, which gives a better view onto the temporal evolution. The Lagrangian perspective leads to the definition of material lines (sets of particles) that behave coherently over time. Haller [20] gave a recent review of the three types of Lagrangian coherent structures (LCS), which are elliptic (vortex boundaries), parabolic (jet cores) and hyperbolic (separating structures). We refer to Onu et al. [39] for a discussion of various LCS extraction techniques.

Finite-Time Lyapunov Exponents. A frequently-used approximation to the hyperbolic Lagrangian coherent structures is the finite-time Lyapunov exponent (FTLE), which measures the separation of nearby-released particles. The ridges of this scalar field form transport barriers that particles do not cross [19, 21], which found numerous applications in predicting ocean pollution, spread of algae in water currents and the feeding of jellyfish [20]. The FTLE field can be described by the flow map $\phi_t^\tau(\mathbf{x})$, which maps a particle seeded at location $\mathbf{x}$ at time t to the location it reaches after advection for duration τ. The spatial gradient of the flow map $\nabla\phi = \frac{\partial}{\partial \mathbf{x}} \phi_t^\tau(\mathbf{x})$ characterizes the behavior of nearby-released particles. The largest separation of particles is approximated by the largest eigenvalue of the right Cauchy-Green deformation tensor $\nabla^{\mathrm{T}}\nabla$. By considering the integration duration and the exponential separation, the finite-time Lyapunov exponent is defined as, cf. Haller et al. [19,21]:

$$FTLE(\mathbf{x},t,\tau) = \frac{1}{|\tau|} \ln \sqrt{\lambda_{\max}(\nabla^{\mathrm{T}}\nabla)}. \tag{1}$$

To compute FTLE, we use central differences to estimate the flow map gradient, as suggested by Haller and Yuan [21]. Other approaches include localized FTLE by Kasten et al. [26] and timeline cell tracking by Kuhn et al. [30]. We refer to Kuhn et al. [31] for a benchmark comparison.

Approximations. The FTLE computation is usually rather expensive, as it requires the tracing of many particle trajectories. Thus, several approaches have been researched to speed up the process. One approach is to employ adaptive refinement of the flow map. For instance, Garth et al. [13] used Catmull-Rom interpolation, Sadlo et al. [47] observed filtered height ridges and Barakat and Tricoche [4] refined around automatically detected geometric structures. Another thread of research is the acceleration of particle tracing, which also benefits FTLE computations by concatenating intermediate flow maps via interpolation and reducing redundant particle integrations [6], using hierarchical line integration [22], interpolation methods [1,9] or edge maps [5]. Barakat et al. [3] implemented a hybrid visualization system to compute 3D FTLE fields interactively by interleaving the rendering and the view-dependent adaptive refinement of a hierarchical FTLE data structure. The method is bound by the available memory and, as it is based on ray marching, always contains discretization errors.

Interpretation of FTLE as Participating Medium. Depending on the integration duration, the FTLE field can contain extremely thin structures, which makes a discretization impractical. For this reason, Günther et al. [17] developed a Monte Carlo method that renders FTLE fields without the discretizations introduced by a grid or the finite step sizes during ray marching. They used transfer functions to map the FTLE value to color $\mathbf{c}$ and extinction σ_t, which casts the direct volume rendering into a light transport problem in participating medium. Albeit they used acceleration data structures, the rendering time for a single high-resolution image was in the order of days. There are two ways to improve the performance: approximating the flow map, which introduces bias and errors, and improving on the rendering side to increase the convergence rate, which we concentrate on this paper. For this, we extend gradient domain rendering [27,34] to single-scattered light transport in participating medium, and we propose two different approaches to reduce the variance of the transmittance estimation, namely a pure continuous version of the biased Fourier opacity mapping [24], which is fast but may contain illumination bias, and a combination of the unbiased ratio tracking [38] with the Fourier-based approximation to bound the error.

2.2 Light Transport in Participating Medium

The following section briefly introduces the unbiased rendering of heterogeneous participating medium. See Fig. 2 for an overview.

Volume Rendering Equation. We cast the rendering of the FTLE scalar field as volumetric light transport problem. The radiative transfer equation [7,25] describes the radiance L that arrives at location $\mathbf{x}$ on the sensor pixel of the viewport under the direction $\vec{\omega}$, which is integrated by collecting all in-scattered light L_i along the view ray $\mathbf{x}_s = \mathbf{x} + s \cdot \vec{\omega}$, with $s \in [0,d]$:

$$L(\mathbf{x} \leftarrow \vec{\omega}) = \int_0^d T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)\, \sigma_s(\mathbf{x}_s)\, L_i(\mathbf{x}_s \leftarrow \vec{\omega})\, ds. \tag{2}$$

Thereby, σ_s is the scattering coefficient that describes the amount of light being scattered per unit travel distance. At all scattering locations

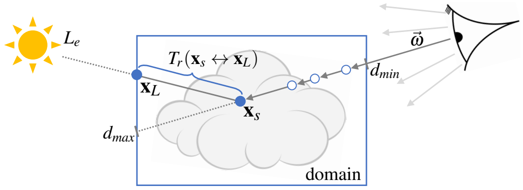


Fig. 2: Illustration of single-scattered light transport in participating medium showing the entry and exit distance d_{min} and d_{max}, the view ray (direction $\vec{\omega}$) and a real scattering event (blue) found at $\mathbf{x}_s$, where a transmittance estimate $T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)$ connects to the light point $\mathbf{x}_L$.

$\mathbf{x}_s$ along the view ray, the in-scattered radiance L_i is determined, which accounts for the amount of radiance reflected towards the viewer:

$$L_i(\mathbf{x}_s \leftarrow \vec{\omega}) = f_p(\mathbf{x}_s, \vec{\omega}_L \rightarrow \vec{\omega})\; T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)\; L_e, \tag{3}$$

where the phase function f_p models the amount of light that is reflected at $\mathbf{x}_s$ from the incoming light direction $\vec{\omega}_L$ towards $\vec{\omega}$, i.e., the viewer. Throughout this paper, we will assume an isotropic phase function, i.e., $f_p = 1/(4\pi)$. L_e is the radiance that is emitted by the light source. Without loss of generality, we assume a directional light.

In both Eqs. (2) and (3), T_r denotes the transmittance between two points, which accounts for the fraction of photons surviving the journey between the points. The transmittance is computed from the optical depth $\tau(d)$, which in turn is the integral of the extinction coefficient σ_t:

$$T_r = e^{-\tau(d)} \quad \text{with} \quad \tau(t) = \int_0^d \sigma_t(s)ds. \tag{4}$$

The extinction coefficient $\sigma_t = \sigma_s + \sigma_a$ describes how much light is lost per unit travel distance, either by scattering σ_s or absorption σ_a. The ratio between scattering and extinction is usually modeled per color channel with the scattering albedo $\mathbf{c} = \sigma_s/\sigma_t$. In our application, both $\mathbf{c}$ and σ_t are derived via transfer functions from the FTLE value.

Importance Sampling. The volume rendering equation in Eq. (2) is solved by Monte Carlo integration, which can be accelerated with importance sampling. In computer graphics, Raab et al. [45] popularized the usage of delta tracking, which was then frequently used and extended [12, 33, 38, 51–53]. Delta tracking estimates the free flight distance of a photon in a heterogeneous medium by sampling $\mathbf{x}_s$ along the view ray with probability $p(\mathbf{x}_s) = \sigma_t(\mathbf{x}_s)\; T_r(\mathbf{x} \leftrightarrow \mathbf{x}_s)$, which in turn concentrates the samples in areas of high transmittance and scattering. Essentially, delta tracking inserts fictitious particles to reach a homogeneous medium with upper extinction bound, called the majorant extinction $\bar{\sigma}_t$. It then repeatedly samples the free flight distance in the homogeneous medium analytically, which creates either virtual or real scattering interactions that are distinguished by Russian roulette sampling. Upon virtual interactions, the photon continues until a real interaction is found. The runtime of the method depends on the tightness of the upper bound, since the average step size in the homogeneous medium is $1/\bar{\sigma}_t$. The sampling can be accelerated by the use of local upper bounds stored in kd-trees [56] or by using piecewise linear or constants bounds [53]. In this work, we use piecewise constant bounds. For light sources inside the medium, Kulla and Fajardo [32] applied multiple importance sampling to select between distance and equi-angular sampling, which is orthogonal to our approach.

Applying delta tracking to compute scattering locations $\mathbf{x}_s$ in Eq. (2), and using an isotropic phase function f_p in Eq. (3) gives a Monte Carlo estimate, obtained after n iterations, cf. Günther et al. [17]:

$$L(\mathbf{x} \leftarrow \vec{\omega}) \approx \frac{L_e}{4\pi n} \sum_{i=1}^{n} \mathbf{c}(\mathbf{x}_s)\; T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L), \tag{5}$$

which only requires the sampling of the color $\mathbf{c}$ at $\mathbf{x}_s$ and an estimate of the transmittance $T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)$ from $\mathbf{x}_s$ towards the light source $\mathbf{x}_L$.

To reduce the number of extinction samples along the view ray, Kutz et al. [33] decomposed the medium into an analytic and a residual part, leading to the weighted decomposition tracking. More recently, Szirmay-Kalos et al. [52] applied control variates to lower the variance in multiple scattering. These methods are orthogonal to our work, since gradient domain rendering can be applied on top for single scattering.

Transmittance Estimation. The remaining ingredient is an estimator of the transmittance $T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)$ from the scattering point $\mathbf{x}_s$ to the light source $\mathbf{x}_L$. The simplest option is to compute the expected value of a binary experiment, where photons are repeatedly sent using delta tracking to test if they reach the scattering point (rated 1) or are absorbed on the way (rated 0), which was applied by Raab et al. [45], Szirmay-Kalos et al. [53] and Günther et al. [16, 17]. More recent techniques include the generalized residual ratio tracking [38, 51], the (spectral) weighted decomposition tracking [33] and the application of control variates [52], which all utilize a decomposition of the medium into a homogeneous and a residual part. The unbiased ratio tracking of Novák et al. [38] computes the joint probability of a photon to survive the light transport from the light source to the scattering point.

2.3 Gradient Domain Rendering

Gradient domain rendering is a recent research branch that was initiated by the seminal work of Lehtinen et al. [34], which was shortly later expanded by Manzi et al. [36]. In principal, gradient domain techniques trace additional offset paths for each pixel, which are generated by a shift operator that reconnects the offset path to the base path as soon as possible. The offset paths allow for an image-space gradient estimation with low variance. The image reconstruction then becomes a minimization problem that searches for an image that best explains the base image and the estimated gradients. By now, most light path integrators have been extended to the gradient domain, including Metropolis light transport [34], path tracing [27], bi-directional path tracing [35] and photon density estimation [23]. Recently, Gruson et al. [15] extended gradient-domain photon density estimation to volumetric light transport for homogeneous medium. In the context of FTLE rendering, we apply gradient domain rendering to heterogeneous participating medium with single scattering, which had still remained an open challenge.

3 Accelerated Monte Carlo FTLE Rendering

Our main goal is to enable the efficient rendering of high-quality animations of finite-time Lyapunov exponent visualizations. For this, we present two acceleration strategies of a recent Monte Carlo-based renderer [17]. In the following, we outline the resulting algorithm. Afterwards, we discuss the two proposed strategies: a screened Poisson reconstruction with a correlated image gradient and the Fourier approximation of the light source transmittance that can be combined with ratio tracking to heuristically bound the error.

3.1 Algorithm Overview

Our rendering algorithm is based on two building blocks, which are explained below.

Base Image Construction. For each pixel of the image, we repeatedly perform the following steps and average the results per pixel:

- Trace a ray through a random location $\mathbf{x}$, positioned on the sensor pixel of the camera, in direction $\vec{\omega}$ into the scene. Intersect this view ray with the bounding box of the domain to determine the entry and exit distance d_{min} and d_{max}.
- Determine the scattering location $\mathbf{x}_s$ by applying delta tracking [55]. The delta tracking starts its walk along the ray at distance d_{min} and returns upon interaction with a real particle the scattering location $\mathbf{x}_s$ and the FTLE value f at this location.
- Map the FTLE value f that was computed at the scattering location via transfer function to the scattering albedo $\mathbf{c}_s$.
- Estimate the transmittance $T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)$ from the scattering location $\mathbf{x}_s$ towards the light source position $\mathbf{x}_L$. While previous methods used unbiased Monte Carlo sampling strategies [38, 45], we propose to use a Fourier approximation of the transmittance,

which can be combined with ratio tracking [38] to heuristically bound the error. Our approach is biased, but converges faster.

- Multiply the scattering albedo $\mathbf{c}_s$ with the transmittance $T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)$ and accumulate the result, see Eq. (5).

The final base image is obtained by multiplying with the emitted radiance L_e, the (constant) phase function f_p, and by dividing by the number of Monte Carlo iterations, cf. Eq. (5).

Gradient-based Rendering. The gradient domain rendering includes the following steps:

- Compute a base image and two additional offset images, using the algorithm above. Note that the offset images must be correlated with the base image to yield a gradient estimate with low variance.
- Perform a screened Poisson reconstruction to determine the image that best explains the base image and the estimated gradients.

3.2 Single-Scattered Gradient Domain Rendering

Gradient domain rendering is a powerful technique to reduce variance, which is orthogonal to almost all other acceleration strategies, such as FTLE approximations, free path samplers and transmittance estimators. For single-scattered medium, gradient domain rendering becomes remarkably simple and straightforward to implement. In principle, three quantities are approximated by a gradient domain Monte Carlo integration, i.e., the base image I^b and its gradients I^{dx} and I^{dy}:

$$I^b_{x,y} = \frac{1}{n}\sum_{i=1}^{n} \hat{I}_{x,y}(i)\,, \tag{6}$$

$$I^{dx}_{x,y} = \frac{1}{n}\sum_{i=1}^{n} \hat{I}_{x+1,y}(i) - \hat{I}_{x,y}(i)\,, \quad I^{dy}_{x,y} = \frac{1}{n}\sum_{i=1}^{n} \hat{I}_{x,y+1}(i) - \hat{I}_{x,y}(i)\,, \tag{7}$$

where $\hat{I}_{x,y}(i)$ is the Monte Carlo estimate at pixel (x,y) in iteration i. The image-space gradients are estimated by finite differences [44], which converge over time with increasing number of Monte Carlo iterations n. Thus, by denoting the finite-difference operators for an image by H^{dx} and H^{dy}, we get:

$$\lim_{n\to\infty} H^{dx} I^b_{x,y} = I^{dx}_{x,y}\,, \quad \lim_{n\to\infty} H^{dy} I^b_{x,y} = I^{dy}_{x,y}\,, \tag{8}$$

Thus, after an infinite number of Monte Carlo iterations, the finite differences that are computed from the base image will coincide with the estimated image-space gradients. This means the estimates of the base image I^b and the gradient images I^{dx} and I^{dy} explain the same image and can be combined to derive a better estimate of the image I.

Sample Generation. In each Monte Carlo iteration i, a base path is traced for each pixel (x,y), which results in a radiance estimate $\hat{I}_{x,y}(i)$. Afterwards, offset rays are generated by a suitable shift mapping to generate estimates $\hat{I}_{x+1,y}(i)$ and $\hat{I}_{x,y+1}(i)$. For this, a ray is traced through the right and upper adjacent pixel. In traditional gradient domain rendering, the offset paths reconnect to the base path as soon as possible, so that most vertices of the paths coincide [27, 34]. This way, the estimate of the gradient has low variance. In single scattering medium, with only one vertex inside the medium, there are no connecting light paths between the base path and the offset paths. Thus, the correlation between base path and offset paths only depends on the correlation of the underlying random number generation during the path construction. In practice, we render the estimates $\hat{I}_{x,y}$, $\hat{I}_{x+1,y}$ and $\hat{I}_{x,y+1}$ with the same random number generation seed.

Reusing Offset Paths. At first sight, the calculation of the base image and the gradient estimates requires the tracing of three rays per pixel (two of them with an offset) with identical random seeds. Since each pixel uses different random seeds, the cost of a gradient domain iteration is three times higher than for a traditional rendering of only the base image. The additionally traced images $\hat{I}_{x+1,y}$ and $\hat{I}_{x,y+1}$, however, can be reused for the base image estimate of the adjacent pixels, i.e., for $I_{x+1,y}$ and $I_{x,y+1}$. Thus, every estimate can be used to reduce the variance of the base image. This means gradient domain rendering has no overhead compared to the traditional base image rendering, except for the final reconstruction, which is described in the following.

Screened Poisson Reconstruction. Given an estimate of the base image I^b, and the estimates of its gradients I^{dx} and I^{dy}, we use a screened Poisson reconstruction to determine the image I that best explains all observations [34]:

$$\arg\min_I \left[\left\| \begin{pmatrix} H^{dx} I \\ H^{dy} I \end{pmatrix} - \begin{pmatrix} I^{dx} \\ I^{dy} \end{pmatrix} \right\|^p + \left\| \alpha (I - I^b) \right\|^p \right], \tag{9}$$

where $p \in \{1,2\}$ denotes the utilized L_p-norm and the α-weight in Eq. (9) determines a compromise between the estimate of the base image and the estimated gradients. Results of the screened Poisson reconstruction for different α values can be found in the additional material. Lehtinen et al. [34] recommended a default value of $\alpha = 1$, which we used in the paper.

L_1 vs. L_2 Norm. While the reconstruction with the L_2 norm was proven to be unbiased [34], a minimization under the L_1 norm may give better results, as it is less sensitive to bright noise spots caused by low-probability path contributions, known as fireflies. In single-scattered light transport such artifacts cannot exist, and thus the L_1 and L_2 norm give visually indistinguishable results. For the minimization of Eq. (9), we use the Poisson solver provided by Kettunen et al. [27]. To minimize under the L_1 norm, they use the iteratively reweighted least squares method (IRLS) to convert the L_1 optimization to a series of L_2 optimization problems. The L_2 minimization can directly be rephrased using the normal equations, casting it into a standard linear problem, which is solved using (preconditioned) conjugate gradients. Given that gradient domain with L_2 norm reconstruction is faster, unbiased and on par with the L_1 norm in our light transport problem, we used it in all our examples. A Delta tracking method (as in [17]), as well as the ratio tracking of Novák et al. [38], would converge to the same image after enough iterations, since both approaches are unbiased. Thus, we used our gradient domain (or ratio tracking) approach to generate the ground truth images of this paper.

3.3 Fourier Approximation of Transmittance

For every scattering point $\mathbf{x}_s$ found along the view ray, we have to estimate the amount of light reaching this scattering point from the light $\mathbf{x}_L$. The ratio of light arriving at the scattering point is expressed by the transmittance $T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)$, for which we need an estimator.

Motivation. We found that a small bias in the shadows is acceptable to us if it enables a more efficient computation. Instead of stochastically estimating the transmittance, we precompute the transmittance from the light source and store it in a suitable and compact representation. Generally, several choices are imaginable, such as the discretization of $T_r(\mathbf{x}_s \leftrightarrow \mathbf{x}_L)$ on a dense grid or a hierarchical and adaptive data structure. In this paper, we opt for a continuous function approximation using a Fourier series, as it is compact in terms of memory, it readily avoids bandwidth-related sampling artifacts and allows for an efficient estimation of integrals along the light ray. Our choice is inspired from real-time shadow computations for semi-transparent geometry. Namely, Jansen and Bavoil [24] proposed Fourier opacity maps, which compactly store the transmittance of a discrete set of surfaces in Fourier basis for each pixel. The technique can be interpreted as a kind of shadow mapping, comprising a view and projection matrix for the light source that project every point in the domain onto a shadow map, which stores a Fourier approximation of the transmittance along the light rays. This approach has recently been used for order-independent transparency in opacity optimization [2]. Jansen and Bavoil demonstrated that the method is superior to dense and hierarchical representations such as opacity shadow maps [28], density clustering [37] or deep shadow maps [57] if the underlying signal to approximate has low frequency. We found this to be a reasonable and useful assumption, since an accurate preservation of hard shadows may introduce numerous distracting discontinuities in image space.

Derivation. Our method can be seen as a continuous version of Fourier opacity maps. Given is the extinction $\sigma_t(z)$ along the light ray, ranging from entry to exit of the domain bounding box with $z \in [0,P]$, see Fig. 3. A Fourier approximation with m frequency bands is:

$$\widehat{\sigma}_t(z) = \frac{a_0}{2} + \sum_{k=1}^{m} a_k \cdot \cos\left(\frac{2\pi kz}{P}\right) + b_k \cdot \sin\left(\frac{2\pi kz}{P}\right), \qquad (10)$$

where the Fourier coefficients a_k and b_k with $k \in \{1,...,m\}$ are:

$$a_k = \frac{2}{P}\int_0^P \sigma_t(z)\cdot\cos\left(\frac{2\pi kz}{P}\right)dz, \;\; b_k = \frac{2}{P}\int_0^P \sigma_t(z)\cdot\sin\left(\frac{2\pi kz}{P}\right)dz. \qquad (11)$$

The coefficients a_k and b_k can be approximated by Monte Carlo integration. For this, the extinction $\sigma_t(z)$ is evaluated at n uniformly-distributed discrete locations d_i with $d_i \in [0,P]$:

$$a_k \approx \frac{2}{n}\sum_{i=1}^{n}\sigma_t(d_i)\cdot\cos\left(\frac{2\pi kd_i}{P}\right), \quad b_k \approx \frac{2}{n}\sum_{i=1}^{n}\sigma_t(d_i)\cdot\sin\left(\frac{2\pi kd_i}{P}\right).$$

This means a Fourier approximation can be computed for the extinction $\sigma_t(z)$ even if it can only be numerically sampled, which is the case for the extinction derived from FTLE fields. Given the Monte Carlo approximations of a_k and b_k for the Fourier representation of $\widehat{\sigma}_t(z)$ in Eq. (10), the optical depth $\tau(d)$ is conveniently computed using:

$$\tau(d) = \int_0^d \widehat{\sigma}_t(z)dz = \frac{a_0}{2}d + P\sum_{k=1}^{n}\frac{a_k}{2\pi k}\cdot\sin\left(\frac{2\pi kd}{P}\right) \qquad (12)$$

$$+ P\sum_{k=1}^{n}\frac{b_k}{2\pi k}\cdot\left(1-\cos\left(\frac{2\pi kd}{P}\right)\right). \qquad (13)$$

Given the optical depth $\tau(d)$, the transmittance $T_r(d) = e^{-\tau(d)}$ is easily computed. The Fourier coefficients a_0, a_i and b_i are precomputed using a finite number of 100 uniformly distributed samples along each light ray. Details on the performance follow later in Section 5.4. We tested different numbers of samples per pixel and settled for 100, since we did not experience further improvements when increasing further. This is because of the band limits inherently given by the number of Fourier frequency bands. The selection of the number of Fourier frequency bands is discussed later in Section 5.1.

3.4 Joint Ratio-Fourier Tracking

The Fourier-based transmittance approximation using Eq. (13) is well-suited to capture smooth extinction transitions. The method, however, may give poor approximations in areas of high-frequency changes, such as around transitions from empty to fully opaque regions. To remove this artifact, we employ a combination of ratio tracking [38] and the above Fourier transmittance approximation. Instead of terminating the path when finding a real scattering location, ratio tracking iterates the full light ray repeatedly sampling the free path travelled by the photon and calculates the joint probability of colliding with virtual particles along the way, which gives a transmittance estimate with lower variance compared to the binary estimator [45] used by Günther et al. [17]. Let $(\mathbf{x}_1, \mathbf{x}_2, \dots)$ be the sequence of scattering locations visited along a light ray by ratio tracking, and let $(d_1, d_2, \dots)$ be the corresponding distances to the entry point of the light ray into the domain. Starting from a ratio tracking, we switch to the Fourier approximation $\tau(d)$ in Eqs. (12)–(13) as soon as the Fourier approximation of the extinction $\widehat{\sigma}_t(d_k)$ in Eq. (10) approximates the real extinction $\sigma_t(\mathbf{x}_k)$ sufficiently well. The transmittance estimator can be formalized as follows:

$$\langle T_r(d)\rangle = \underbrace{\prod_{i=1}^{k}\left(1-\frac{\sigma_t(\mathbf{x}_i)}{\bar{\sigma}_t}\right)}_{\text{ratio tracking}} \cdot \underbrace{e^{-\tau(d_i)}}_{\text{Fourier}} \quad \text{for} \quad k : |\underbrace{\sigma_t(\mathbf{x}_k)}_{\text{real}} - \underbrace{\widehat{\sigma}_t(d_k)}_{\text{Fourier}}| < \varepsilon. \qquad (14)$$

Thereby, ε is an error tolerance set by the user, which balances between accuracy and speed. In all examples, we selected ε conservatively based on the majorant extinction $\bar{\sigma}_t$, i.e., $\varepsilon = \bar{\sigma}_t/100$, targeting an accurate approximation. Different choices are discussed later in Section 5.3.

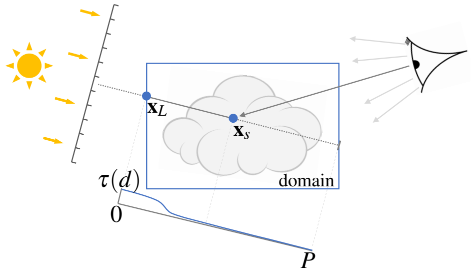


Fig. 3: The Fourier approximation of transmittance $\tau(d)$ is computed over the domain $[0,P]$ along the light ray, where $d = 0$ is the entry distance and $d = P$ is the exit distance. The Fourier coefficients of the approximation are stored for each pixel in the orthographic light view.

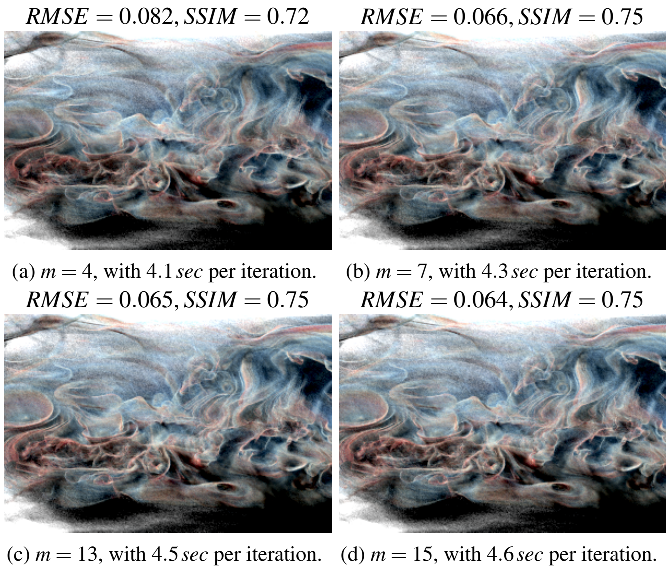
4 Implementation

In the following, we share details of our implementation. All algorithms were implemented on the GPU using compute shaders that parallelize the computation across the screen pixels.

Offset Path Computation. In addition to the base path, the gradient domain computation requires the tracing of two additional offset paths. Instead of writing a GPU kernel that calls three ray casts successively, we split the tracing into three separate kernel launches of the *same* light integration kernel. In our execution model, each thread is responsible for one pixel. The free path sampling and the transmittance estimator require a random number sequence for which we employ a linear congruential generator [40], which is represented by a state variable. Each pixel stores its random number state in global memory, which is only modified by the third kernel launch. This way, the base path and the two offset paths use identical random number sequences, which greatly reduces the variance in the gradient estimate. The pixel offsets (no offset, right neighbor and top neighbor) for the generation of the view ray are respectively stored in a constant buffer during the three kernel launches, which is coherently accessed by all threads being executed in parallel. The resulting image estimates $\hat{I}_{x,y}(i)$, $\hat{I}_{x+1,y}(i)$ and $\hat{I}_{x,y+1}(i)$ of iteration i are written to textures and a separate kernel combines the images into the accumulated values in Eqs. (6)–(7). Note that the estimates $\hat{I}_{x+1,y}(i)$ and $\hat{I}_{x,y+1}(i)$ can be reused for the accumulation of the radiance in the adjacent pixels. Thereby, the values have to be read by a gathering operation to avoid race conditions, i.e., each pixel collects the contributions from its neighboring pixels.

Fourier Coefficient Maps. We store the Fourier coefficients in four-component float variables in global memory. The calculation of the cosine and sine terms is vectorized and always computed for two bands at a time by storing the cosine and sine term of the first band in the first two components and the cosine and sine term of the second band in third and fourth component. Thus, the storage of 13 Fourier frequency bands (a_0 and a_i, b_i with $i \in \{1,\dots,13\}$), requires 7 four-component variables (using 31 floats). We always used a resolution of 512 × 512 pixels for the Fourier maps, which we found to be sufficient. A study of the impact of different Fourier map resolutions can be found in the additional material. In total, 28 MB are required to store the transmittance approximation using 13 frequency bands in the Fourier representation. The projection of the extinction coefficients into Fourier basis happens once in a preprocess. For this, we sample the light ray between entry and exit in the domain bounding box at 100 uniformly-spaced locations and project the sampled extinction values into Fourier basis, which requires the calculation and mapping of FTLE values.

Spatially-Varying Majorant Extinction. The runtime of both delta tracking [55] and ratio tracking [38] depends on the tightness of the upper extinction bound $\bar{\sigma}_t$, which is why upper bounds should be


$RMSE = 0.082, SSIM = 0.72$

$RMSE = 0.066, SSIM = 0.75$

(a) $m = 4$, with 4.1 *sec* per iteration.

(b) $m = 7$, with 4.3 *sec* per iteration.

$RMSE = 0.065, SSIM = 0.75$

$RMSE = 0.064, SSIM = 0.75$

(c) $m = 13$, with 4.5 *sec* per iteration.

(d) $m = 15$, with 4.6 *sec* per iteration.

Fig. 4: Comparison of our Fourier transmittance estimator for different numbers of frequency bands. All images were computed with 100 iterations of gradient domain in the ECMWF flow. We also list Root-mean-square error (RMSE) and structural similarity index (SSIM).

estimated locally [53, 56]. We store piecewise constant upper bounds in a regular grid [53]. Since a tight upper FTLE bound is not known apriori, we start with an estimate and progressively converge with each visited sampling location as described by Günther et al. [17]. The ray traversal through the majorant extinction grid requires care at the voxel boundaries. Both delta tracking and ratio tracking perform free path sampling steps in the virtual homogeneous medium, with the majorant extinction being read from the current voxel. If the tentative scattering location was found inside the next voxel, the traveled world distance inside the next voxel has to be converted into the corresponding optical depth (by multiplying with the majorant extinction of the last voxel) and then translated back into the correct world distance (by dividing by the majorant extinction of the next voxel). The grid resolution of the majorant extinction grids was chosen manually, as by Günther et al. [17]. This manual step could be avoided by using a kd-tree with a subdivision heuristic [56] to store the majorant extinction grids.

5 Results

In the following sections, we discuss parameter choices, performance and compare the proposed methods with previous work.

5.1 Frequency Bands in Fourier Approximation

The number of frequency bands m used to compute the extinction in Eq. (10) determines the quality of the approximated $\widehat{\sigma}_t(z)$. In the limit, it converges to the real extinction $\sigma_t(z)$. Fig. 4 shows a comparison of the approximation of transmittance using different numbers of bands in the ECMWF flow, reporting the RMSE and SSIM when comparing to a ground truth image generated with $1k$ Monte Carlo iterations of our gradient domain with ratio tracking ($t = 2.16\,h$). When using a small number of bands, the reconstructed signal overestimates the transmittance and misses high frequencies, which results in darker and softer shadows. We found that smooth shadows are visually more pleasing, which comes to our advantage since it allows us to obtain convincing images with a relatively small number of frequency bands. It can be seen in Eq. (13) that the terms a_k and b_k decay at a rate of $\frac{1}{\pi k}$ and thus components for $k > 10$ have a magnitude of about 3% or less compared to a_0. Differences between 13 and 15 bands are already almost imperceptible which is why we selected 13 bands.

5.2 Comparison with Previous Work

In Fig. 5, we visually and quantitatively compare the different techniques in the Wall-Mounted Cylinder (left) and in the CTBL flow (right). In the first row, we applied our gradient domain rendering to generate a ground truth image of the Wall-Mounted Cylinder using $1k$ iterations and $10k$ iterations of ratio tracking for the CTBL. The second row shows delta tracking as used by Günther et al. [17], which is the noisiest, since it uses a binary transmittance estimator. In the third row, we again use gradient domain rendering with delta tracking, which results are smoother than the previous due to the gradient estimate used by the screened Poisson reconstruction. The fourth row uses the ratio tracking of Novák et al. [38] to estimate the transmittance towards the light source. This method computes the joint probability for a photon to reach the scattering point, which has lower variance than the binary estimator. The fifth row shows our joint ratio-Fourier tracking with gradient domain rendering applied on top. In the Wall-Mounted Cylinder the noise is visibly reduced, while there is still no improvement achieved in the CTBL data set. The last row shows a pure Fourier approximation with gradient domain rendering applied on top. Here, the noise is significantly reduced in both scenes. In the CTBL flow, a difference in the illumination is apparent (reflected in a higher RMSE), which seems acceptable, since all structures are still well visible. The structural similarity index (SSIM), which predicts the perceived quality of pictures, is nevertheless higher than delta for our methods, since the smoothing is visually more pleasant than noise.

5.3 Error Tolerance

In Fig. 6, we study the error tolerance ε of our joint ratio-Fourier transmittance estimator in Eq. (14) in comparison with the ratio tracking estimator of Novák et al. [38] and a Fourier approximation using Eqs. (12)–(13). The ECMWF flow is a particularly challenging test case, as it contains numerous thin ridge structures that are difficult to estimate with a low number of Fourier frequency bands. For the ground truth image, we used $1k$ Monte Carlo iterations of our gradient domain with ratio tracking ($t = 1.85\,h$). Fig. 6a shows an early reference solution after 300 iterations. For $\varepsilon = \bar{\sigma}_t/100$ our joint ratio-Fourier transmittance estimator in Fig. 6b gives acceptable results that are very similar to the reference solution. For a value of $\varepsilon = \bar{\sigma}_t/10$, however, artifacts appear in Fig. 6c, which are caused by the local extinction-based error estimate in Eq. (14). In scenes with less variation in the illumination, we found this threshold to be acceptable as well, giving a greater performance increase. A transmittance approximation based on Fourier only, as in Fig. 6d, gives by far the fastest convergence, since the light transmittance has zero variance. In this difficult case, the changes in the illumination are clearly visible when comparing with the reference solution. A close up of this figure can be found in the additional material.

5.4 Convergence and Performance

In this section, we analyze the runtime and convergence of our method. All results were rendered with an Intel Core i7-6700K CPU with 4 GHz, 32 GB RAM and NVIDIA GeForce GTX 1080 GPU. Table 1 lists the runtime in *sec* per Monte Carlo iteration for delta tracking (D), gradient domain (G), gradient with ratio tracking (GR), gradient domain with pure Fourier approximation (GF) and ratio-Fourier tracking (GRF). The precomputation time of the Fourier approximation of $\widehat{\sigma}(z)$ is listed in the eighth column and the last column reports the time required for the final screened Poisson reconstruction of Eq. (9). We observe that the pure Fourier approximation can be $2-3\times$ faster than our joint ratio-Fourier tracking. The cost for individual iterations is not only lower, it also contains less noise, since the pure Fourier-based transmittance approximation has no variance at all. The joint ratio-Fourier method, however, can bound the illumination error, which might also be desirable. The time required for computing the Fourier approximation gives an indicator for the overall cost of pathline tracing in the data sets, since all examples use the same resolution of 512×512 pixels for the Fourier coefficient map. The time required for the Poisson reconstruction scales with the screen resolution. Results for high resolutions are shown in the additional material. In general, we

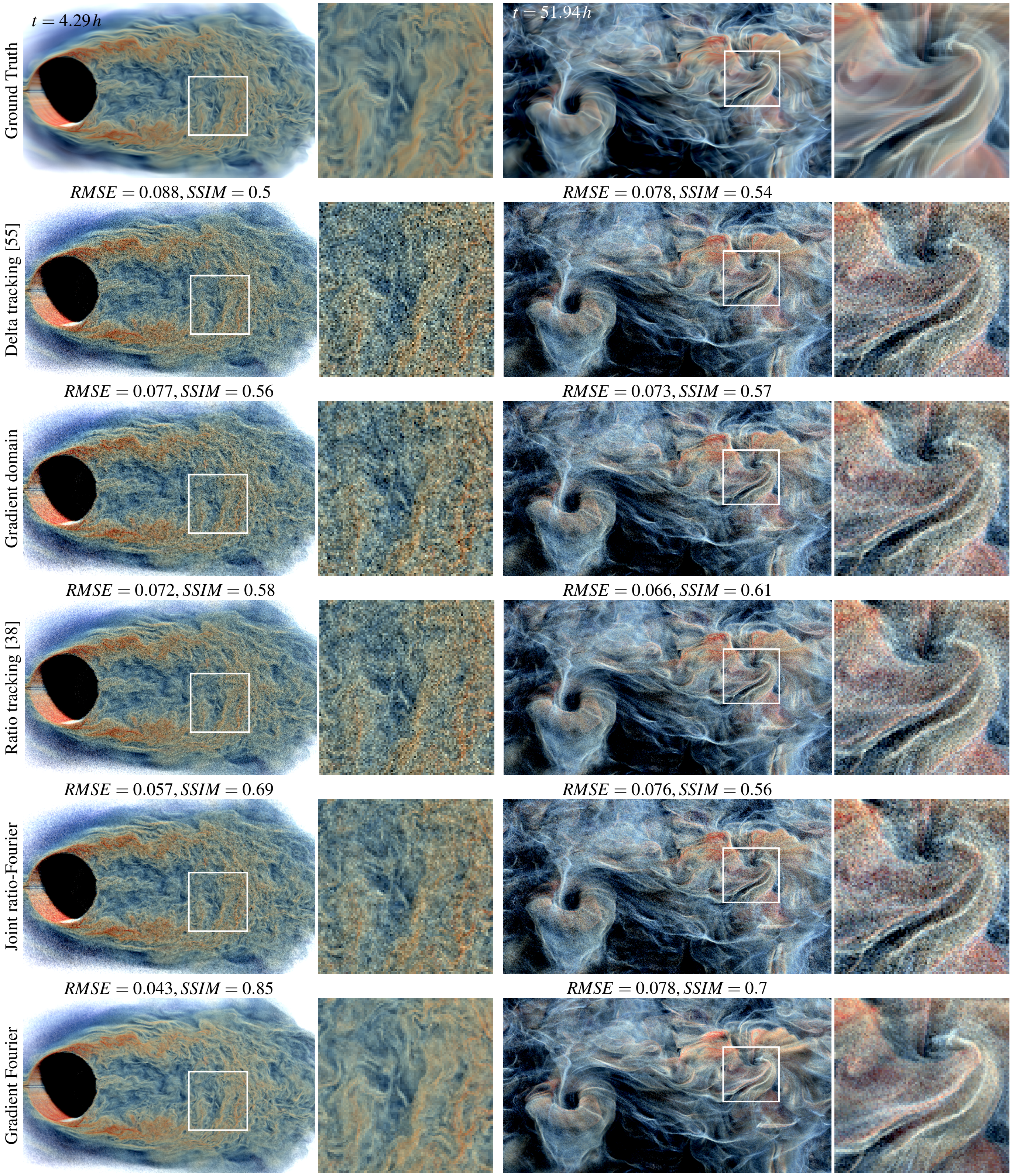
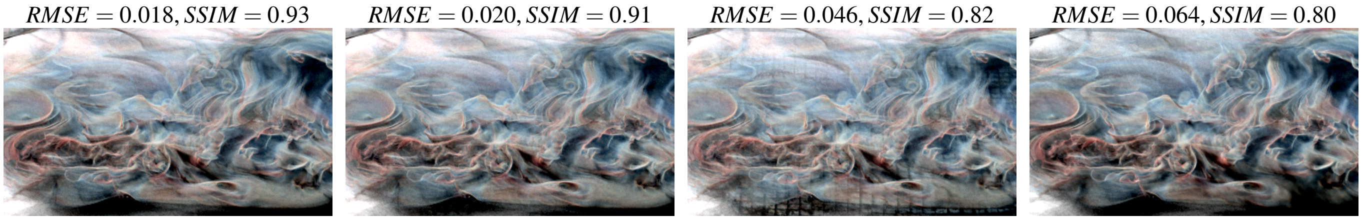

$t = 4.29h$
$t = 51.94h$
Ground Truth
$RMSE = 0.088, SSIM = 0.5$
$RMSE = 0.078, SSIM = 0.54$
Delta tracking [55]
$RMSE = 0.077, SSIM = 0.56$
$RMSE = 0.073, SSIM = 0.57$
Gradient domain
$RMSE = 0.072, SSIM = 0.58$
$RMSE = 0.066, SSIM = 0.61$
Ratio tracking [38]
$RMSE = 0.057, SSIM = 0.69$
$RMSE = 0.076, SSIM = 0.56$
Joint ratio-Fourier
$RMSE = 0.043, SSIM = 0.85$
$RMSE = 0.078, SSIM = 0.7$
Gradient Fourier

Fig. 5: Equal-time comparisons for 360 *sec* in the WALL-MOUNTED CYLINDER flow (left) and for 680 *sec* in the CTBL flow (right). From top to bottom: ground truth image with gradient domain (1*k* iterations, left) and ratio (10*k* iterations, right), delta tracking as used by Günther et al. [17], gradient domain rendering applied to delta tracking, the ratio tracking of Novák et al. [38], our joint ratio-Fourier tracking with gradient domain rendering and a pure Fourier approximation with gradient domain rendering. We list root-mean-square error (RMSE) and structural similarity index (SSIM). In the WALL-MOUNTED CYLINDER, our methods visibly reduce the noise with little bias. In the CTBL, our joint method shows only a small improvement. The pure Fourier approximation, however, reduces the noise successfully, but also contains a visible bias in the illumination. Since all FTLE structures can still be perceived well, this bias is an acceptable trade-off for the reduced noise.



$RMSE = 0.018, SSIM = 0.93$

$RMSE = 0.020, SSIM = 0.91$

$RMSE = 0.046, SSIM = 0.82$

$RMSE = 0.064, SSIM = 0.80$

(a) Ratio-tracking only (reference solution), 10.7 *sec* per iteration.

(b) Our ratio-Fourier with $\varepsilon = \bar{\sigma}_t/100$, 8.9 *sec* per iteration.

(c) Our ratio-Fourier with $\varepsilon = \bar{\sigma}_t/10$, 7.4 *sec* per iteration.

(d) Fourier approximation of transmittance only, 5.4 *sec* per iteration.

Fig. 6: Comparison of our joint ratio-Fourier transmittance estimator with ratio tracking [38] (6a) and our pure Fourier approximation of the transmittance (6d). All images were computed using 300 Monte Carlo iterations with gradient domain. We list the root-mean-square error (RMSE) and the structural similarity index (SSIM) when comparing to a ground truth image. This example in the ECMWF flow contains numerous thin ridge structures, which is a particularly difficult case for the Fourier approximation. The reference solution in 6a is best reached for $\varepsilon = \bar{\sigma}_t/100$ with our joint method in 6b. A value of $\varepsilon = \bar{\sigma}_t/10$ contains very noticeable artifacts in 6c, and our approximation only based on Fourier in 6d gives the fastest convergence, but also shows the largest illumination differences.

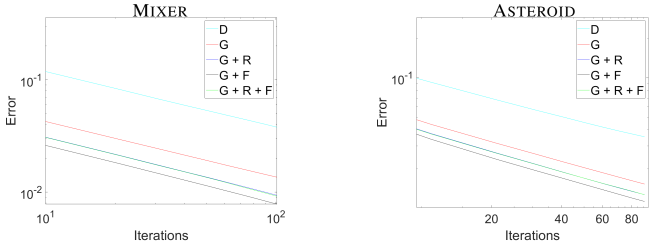


Fig. 7: Logarithmic convergence of Mixer and Asteroid data sets.

	Resolution	D	G	GR	GF	GRF	$\widehat{\sigma}(z)$	L_2
ABC	500×500	0.07	0.2	0.28	0.11	0.29	0.13	20.74
ASTEROID	480×360	9.86	30.18	24.88	17.8	24.3	22.4	15.76
CTBL	450×240	7.66	23.66	33.72	10.3	33.7	13.2	10.39
W-M CYL	500×300	5.15	15.45	17.43	5.35	17.8	1.81	14.03
DKRZ	800×800	3.53	10.95	14.99	7.18	9.04	21.4	43.99
ECMW	350×200	2.43	7.13	7.78	4.96	7.35	15.9	6.77
MIXER	500×500	4.32	13	17	7.53	15	40.2	23.2

Table 1: Runtime in *sec* for a single iteration of delta (D), gradient domain (G), gradient ratio (GR), gradient Fourier (GF) and gradient ratio-Fourier (GRF), the precomputation of the transmittance approximation $\widehat{\sigma}(z)$ and the final Poisson reconstruction using the L_2 norm.

found that 200 gradient domain Monte Carlo iterations with the pure Fourier approximation of the transmittance are usually sufficient to obtain frame-coherent high-quality visualizations of FTLE fields.

Fig. 7 shows convergence plots for the Mixer and the Asteroid data sets after 80 iterations. In the additional material, we show the convergence over time. For our two methods, gradient domain with Fourier (GF) or with ratio-Fourier (GRF), the error is measured with respect to their own ground truth images. For delta tracking (D), gradient domain (G) and gradient ratio (GR), the error is measured with the same ground truth image generated by 1000 iterations of gradient domain. These plots show that, to have an error measure similar to our gradient with Fourier (GR), delta tracking needs around 10× more Monte-Carlo iterations. A detailed comparison between images after 80 iterations for the Mixer data set are shown in the additional material, as well as several time-lapse convergence animations in the video.

5.5 Applications

The two proposed approaches enable the practical rendering of animations over time, movements along camera paths and moving clipping planes through the data domain, allowing domain scientists a more in depth exploration and understanding of the data. In the following, we present some data sets where our animations led to new insights on the data. We refer to the video for the full animations.

Asteroid. This data set is a numerical multi-physics hydrodynamics simulation using the xRage code [14] that aimed to gain new insights into possible water impact scenarios. We used the *yA31* series, where the asteroid impacts the water with an angle of 45° and no airbust. In Fig. 8 top, two frames of the animation are shown, where we can observe the corridor formed by the falling asteroid, the uplift paths of particles rising up into the stratosphere and the high degree of turbulence that followed the strong impact. We refer to the video to observe the vortical motion left of the impact site and below the entry corridor, which could be formed by the forward rising water vapor plume and the strong shear along the entry corridor, respectively. Time-varying effects such as the vortical motion would not be visible in a single image.

Cloud-Topped Boundary Layer. Fig. 8 middle shows a cloud-resolving boundary layer simulation. The data set has a 25 meters spatial resolution, uses double-periodic boundary conditions and homogeneous surface forcing. By creating a rotating animation, we can observe from all possible angles the detailed convection structures and the turbulent plumes of rising cumulus clouds, which exhibit higher FTLE values and therefore more red ridges. In fact, the rising vortex ring in Fig. 5 (right) has not been known to exist in the first introduction of Monte Carlo FTLE [17] and was only discovered later after rerendering the same scene from a different view [16]. This feature could have been discovered earlier if the render time would have been lower.

Rotating Mixer. This sequence is an unsteady flow containing the simulation of a liquid that is stirred into motion by a small propeller. The FTLE field is shown in Fig. 1 with the delta tracking approach used by Günther et al. [17] and using our two novel methods. With our methods, the thin FTLE structures are apparent already after a minute, whereas the previous approach [17] still contains a significant amount of noise. In the presence of noise, multiple nearby ridges are easily mistaken for a single ridge. Further, obtaining noise-free previews earlier eases the setting of transfer functions and camera parameters. In the additional material, we compare all techniques after 80 iterations to show that we need 10 hours to render a 60 frames noise-free animation, whereas the approach by Günther et al. [17] takes 72 hours.

Cloud Evolution. Fig. 8 bottom shows selected frames of a slicing animation through this air flow. The camera setup of Rimensberger et al. [46] shows the atmospheric air flow profile from South to North over Germany, with the ground at the top and the stratosphere at the bottom. We explored this complex 3D data set by animating a clip plane that slices through the domain from East to West, allowing us to observe the thin horizontal movement layer near the ground, the tropopause (shown at the bottom of the domain by horizontal wind shear-induced FTLE structures) and the vertical energy transport between the ground and the tropopause. Due to the high extinction, the tropopause layer passing horizontally through the entire domain would not have been visible without the slicing animation.

ECMWF. Our next use case demonstrates another meteorological air flow. Following Günther et al. [17], we select a time-averaged tropospheric air flow in the Northern hemisphere. The air flow is shown in Fig. 6, where the thin and complex ridge surface structures make

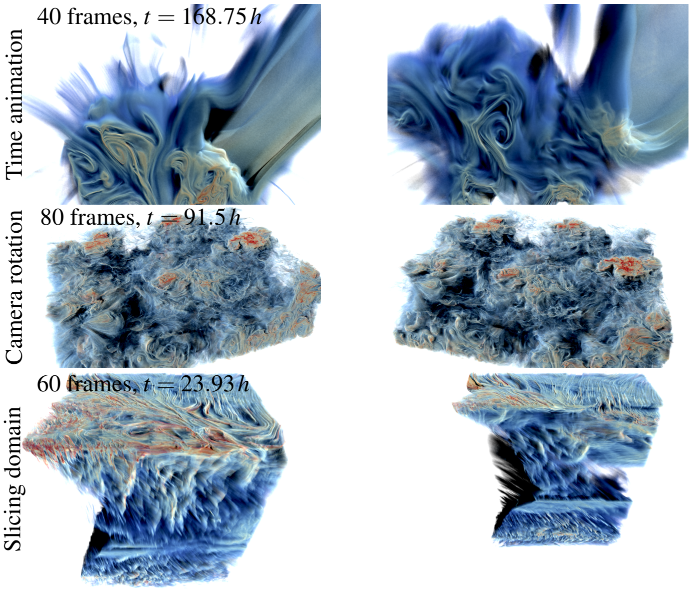


Fig. 8: Two frames of the ASTEROID time animation (top), the camera flight around the CTBL simulation (middle) and a slicing animation through the CLOUD EVOLUTION (bottom), all using 200 it per frame.

the transmittance estimation challenging. In this flow, convective flow features are apparent, such as vortices (cyclones) and stream-like patterns that guide the transport of trace gases. Such FTLE visualizations provide a Lagrangian perspective on the mixing and transport behavior.

WALL-MOUNTED CYLINDER. Our last example contains a Large-Eddy simulation (LES) around a WALL-MOUNTED CYLINDER. Due to a high Reynolds number of $Re = 200,000$, this flow contains a significant amount of turbulence, leading to rapid mixture, which is visible by the highly-folded FTLE structures. In Fig. 5 (left), we used this vector field to compare our gradient domain-based methods with existing approaches. The accompanying video contains a time series animation of this data set, showing the temporal evolution of the FTLE structures, showing the interaction of hyperbolic LCS during mixing.

5.6 Discussion

Illumination Bias. The use of the Fourier-approximated transmittance accelerates the rendering considerably at the cost of losing accuracy in the illumination, which is visible in Fig. 5 (right). For this reason, we introduced the joint ratio-Fourier estimator, which heuristically bounds this error. The adjustment of the threshold ε between the unbiased method [38] and our Fourier approximation then becomes a trade-off between speed and accuracy. When the user is more concerned about time, such as for video generation, switching to a purely Fourier approximation is recommended even though some areas of the image can be darker or lighter than the ground truth. These changes are hardly visible in most examples and, in extreme hard illumination cases such as flows containing numerous internal ridge structures, the illumination difference could be avoided with the ratio-Fourier approach.

Approximation Artifacts. In addition to the illumination bias, introducing a Fourier-based approximation of the transmittance can generate ringing artifacts in the shadows when estimating high frequency changes of light with only few coefficients. Using more coefficients hardly eliminates this artifact due to the low impact of larger k coefficients in the overall signal. Thus, changing to a ratio-Fourier approach is more effective in this case. Detailed figures of approximation and illumination artifacts can be found in the additional material. In Fig. 6, we used the ECMWF data set to show a difficult situation for the Fourier transmittance approximation, which led us to a conservative choice for the error tolerance ε. Setting the value too high results in artifacts, since when the difference is small enough, the Fourier approximation is used, which is darker than the reference, which creates visible changes.

Parameter Adjustment. For the adjustment of the transfer functions and camera parameters, we inspect the early results, obtained after 10–20 iterations. For a more interactive preview, it is imaginable to use other techniques such as the interactive method of Barakat et al. [3].

Multiple Scattering. In this paper, we concentrated on single scattering, as it is more suited for visualization. Multiple scattering essentially diffuses the radiance, which results in color bleeding that might lead to misinterpretations of color-coded attributes. For the visualization of smoke and clouds [32, 38], multiple scattering is highly relevant. In that case, our Fourier-based transmittance estimators are applicable in the final connection of a path to the light source.

Other Function Approximations. We followed Jansen and Bavoil [24] in their choice to Fourier-approximate an extinction signal, since Fourier approximations are compactly stored, continuous, smooth and can be analytically integrated. Our joint ratio-Fourier estimator could similarly be combined with any other function approximation, such as a grid discretization [28], an adaptive representation [37, 57] or another continuous approximation using any other orthonormal basis.

Denoising. To reduce the noise, orthogonal denoising approaches such as the recent work by Chaitanya et al. [8] could be used to let machine learning reconstruct Monte Carlo rendered image sequences.

6 CONCLUSIONS

In this paper, we accelerated the Monte Carlo rendering approach of Günther et al. [17] for the visualization of finite-time Lyapunov exponent fields. Our technique enables the practical rendering of FTLE animations, including time sequences, camera paths and slicing through the domain. For this, we improved on the light transport simulation using two complementary strategies. First, we extended gradient domain rendering [27, 34] to single scattering in heterogeneous participating medium, which leads to a variance reduction due to an estimate of correlated gradients that are considered in a screened Poisson reconstruction. Second, we approximated the transmittance from the light source using a Fourier representation, which we combined with ratio tracking [38] in a joint transmittance estimator. The joint estimator provides a trade-off between the unbiased ratio tracking and the efficient Fourier-based approximation of the transmittance. Our improvements enable for the first time the efficient Monte Carlo rendering of high-quality FTLE visualizations, which enables the computation of video sequences.

In the future, we would like to investigate orthogonal strategies to increase the performance of the FTLE computation, namely flow map approximations [1, 9], and to reduce the noise, like the use of reconstruction filters [58].

ACKNOWLEDGMENTS

We wish to thank Gabor Janiga (University of Magdeburg) for providing the ROTATING MIXER sequence. The European Centre for Medium-Range Weather Forecasts (ECMWF) provided a large collection of meteorological reanalysis simulation data [10]. The ASTEROID and the CLOUD EVOLUTION data sets were provided for the IEEE Scientific Visualization Contest 2018 (Los Alamos National Laboratory [41]) and 2017 (HD(CP)^2 project), respectively. Alexander Kuhn provided the CTBL flow, originally from the UCLA-LES model [50]. The WALL-MOUNTED CYLINDER sequence was provided by Frederich et al. [11]. This work was supported by the Swiss National Science Foundation (SNSF) Ambizione grant no. PZ00P2_180114.

REFERENCES

[1] A. Agranovsky, H. Obermaier, C. Garth, and K. I. Joy. A multi-resolution interpolation scheme for pathline based Lagrangian flow representations. In *Proc. SPIE, Visual Data Analysis Conference*, 2015.

[2] I. Baeza Rojo, M. Gross, and T. Günther. Fourier opacity optimization for scalable exploration. *IEEE Transactions on Visualization and Computer Graphics*, page to appear, 2019.

[3] S. S. Barakat, C. Garth, and X. Tricoche. Interactive computation and rendering of finite-time Lyapunov exponent fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1368–1380, 2012.

[4] S. S. Barakat and X. Tricoche. Adaptive refinement of the flow map using sparse samples. *IEEE Transactions on Visualization and Computer Graphics (Proc. SciVis)*, 19(12):2753–2762, 2013.

[5] H. Bhatia, S. Jadhav, P.-T. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci. Flow visualization with quantified spatial and temporal errors using edge maps. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1383–1396, 2012.

[6] S. L. Brunton and C. W. Rowley. Fast computation of FTLE fields for unsteady flows: A comparison of methods. *Chaos*, 20:017503, 2010.

[7] E. Cerezo, F. Pérez, X. Pueyo, F. J. Seron, and F. X. Sillion. A survey on participating media rendering techniques. *The Visual Computer*, 21(5):303–328, 2005.

[8] C. R. A. Chaitanya, A. S. Kaplanyan, C. Schied, M. Salvi, A. Lefohn, D. Nowrouzezahrai, and T. Aila. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph.*, 36(4):98:1–98:12, July 2017.

[9] J. Chandler, H. Obermaier, and K. I. Joy. Interpolation-based pathline tracing in particle-based flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):68–80, 2015.

[10] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and F. Vitart. The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137(656):553–597, 2011.

[11] O. Frederich, E. Wassen, and F. Thiele. Prediction of the flow around a short wall-mounted cylinder using LES and DES. *Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM)*, 3(3-4):231–247, 2008.

[12] M. Galtier, S. Blanco, C. Caliot, C. Coustet, J. Dauchet, M. E. Hafi, V. Eymet, R. Fournier, J. Gautrais, A. Khuong, B. Piaud, and G. Terrée. Integral formulation of null-collision Monte Carlo algorithms. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 125:57–68, 2013.

[13] C. Garth, F. Gerhardt, X. Tricoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization)*, 13(6):1464–1471, 2007.

[14] M. Gittings, R. Weaver, M. Clover, T. Betlach, N. Byrne, R. Coker, E. Dendy, R. Hueckstaedt, K. New, W. R. Oakes, D. Ranta, and R. Stefan. The RAGE radiation-hydrodynamic code. *Computational Science and Discovery*, 1(1):015005, 2008.

[15] A. Gruson, B.-S. Hua, N. Vibert, D. Nowrouzezahrai, and T. Hachisuka. Gradient-domain volumetric photon density estimation. *ACM Trans. Graph.*, 37(4):82:1–82:13, July 2018.

[16] T. Günther, A. Kuhn, H.-C. Hege, M. Gross, and H. Theisel. Progressive monte carlo rendering of atmospheric flow features across scales. *Phys. Rev. Fluids*, 2:090502, Sep 2017.

[17] T. Günther, A. Kuhn, and H. Theisel. MCFTLE: Monte Carlo rendering of finite-time Lyapunov exponent fields. *Computer Graphics Forum (Proc. EuroVis)*, 35(3):381–390, 2016.

[18] T. Günther and H. Theisel. The state of the art in vortex extraction. *Computer Graphics Forum*, 37(6):149–173, 2018.

[19] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D: Nonlinear Phenomena*, 149(4):248–277, Mar. 2001.

[20] G. Haller. Lagrangian coherent structures. *Annual Review of Fluid Mechanics*, 47:137–162, 2015.

[21] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D: Nonlinear Phenomena*, 147(3–4):352–370, 2000.

[22] M. Hlawatsch, F. Sadlo, and D. Weiskopf. Hierarchical line integration. *IEEE Trans. on Vis. and Computer Graphics*, 17(8):1148–1163, 2011.

[23] B. Hua, A. Gruson, D. Nowrouzezahrai, and T. Hachisuka. Gradientdomain photon density estimation. *Computer Graphics Forum*, 36(2):31–38, 2017.

[24] J. Jansen and L. Bavoil. Fourier opacity mapping. In *Proc. Symposium on Interactive 3D Graphics and Games*, pages 165–172, New York, NY, USA, 2010. ACM.

[25] W. Jarosz. *Efficient Monte Carlo Methods for Light Transport in Scattering Media*. PhD thesis, UC San Diego, La Jolla, CA, USA, 2008.

[26] J. Kasten, C. Petz, I. Hotz, B. Noack, and H.-C. Hege. Localized finite-time Lyapunov exponent for unsteady flow analysis. In *Proceedings of Vision, Modeling and Visualization*, pages 265–274, 2009.

[27] M. Kettunen, M. Manzi, M. Aittala, J. Lehtinen, F. Durand, and M. Zwicker. Gradient-domain path tracing. *ACM Transactions on Graphics*, 34(4):123:1–123:13, July 2015.

[28] T.-Y. Kim and U. Neumann. Opacity shadow maps. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 177–182, London, UK, UK, 2001. Springer-Verlag.

[29] T. Kroes, F. H. Post, and C. P. Botha. Exposure render: An interactive photo-realistic volume rendering framework. *PLoS ONE*, 7(7):e38586, 07 2012.

[30] A. Kuhn, W. Engelke, C. Rössl, M. Hadwiger, and H. Theisel. Time line cell tracking for the approximation of Lagrangian coherent structures with subgrid accuracy. *Computer Graphics Forum*, 33(1):222–234, 2014.

[31] A. Kuhn, C. Rössl, T. Weinkauf, and H. Theisel. A benchmark for evaluating FTLE computations. In *Proceedings of 5th IEEE Pacific Visualization Symposium (PacificVis 2012)*, pages 121–128, Songdo, Korea, 2012.

[32] C. Kulla and M. Fajardo. Importance sampling techniques for path tracing in participating media. *Computer Graphics Forum (Proc. Eurographics Symposium on Rendering)*, 31(4):1519–1528, 2012.

[33] P. Kutz, R. Habel, Y. K. Li, and J. Novák. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, 36(4):111:1–111:16, 2017.

[34] J. Lehtinen, T. Karras, S. Laine, M. Aittala, F. Durand, and T. Aila. Gradient-domain metropolis light transport. *ACM Trans. Graph.*, 32(4):95:1–95:12, July 2013.

[35] M. Manzi, M. Kettunen, M. Aittala, J. Lehtinen, F. Durand, and M. Zwicker. Gradient-domain bidirectional path tracing. In *Proc. Eurographics Symposium on Rendering*, volume 1, page 3, 2015.

[36] M. Manzi, F. Rousselle, M. Kettunen, J. Lehtinen, and M. Zwicker. Improved sampling for gradient-domain metropolis light transport. *ACM Transactions on Graphics*, 33(6):178:1–178:12, Nov. 2014.

[37] T. Mertens, J. Kautz, P. Bekaert, and F. V. Reeth. A self-shadow algorithm for dynamic hair using density clustering. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*, EGSR'04, pages 173–178, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[38] J. Novák, A. Selle, and W. Jarosz. Residual ratio tracking for estimating attenuation in participating media. *ACM Transaction on Graphics (SIGGRAPH Asia)*, 33(6):179:1–179:11, 2014.

[39] K. Onu, F. Huhn, and G. Haller. LCS tool: a computational platform for Lagrangian coherent structures. *Journal of Computational Science*, 7:26–36, 2015.

[40] S. K. Park and K. W. Miller. Random number generators: Good ones are hard to find. *Commun. ACM*, 31(10):1192–1201, Oct. 1988.

[41] J. Patchett and G. Gisler. Deep water impact ensemble data set. *Los Alamos National Laboratory, LA-UR-17-21595, available at http://dssdata. org*.

[42] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser. The state of the art in topology-based visualization of unsteady flow. *Computer Graphics Forum*, 30(6):1789–1811, 2011.

[43] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

[44] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C*, volume 2. Cambridge university press Cambridge, 1996.

[45] M. Raab, D. Seibert, and A. Keller. Unbiased global illumination with participating media. In *Monte Carlo and Quasi-MC Methods 2006*, pages 591–605. Springer Berlin Heidelberg, 2008.

[46] N. Rimensberger, M. Gross, and T. Günther. Visualization of clouds and atmospheric air flows. In *IEEE Scientific Visualization Contest 2017*, 2017.

[47] F. Sadlo and R. Peikert. Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction. *IEEE Trans. on Visualization and Computer Graphics (IEEE Visualization)*, 13(6):1456–1463, 2007.

[48] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3-4):271–304, 2005.

[49] H. R. 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.

[50] B. Stevens. Introduction to UCLA-LES, 2013.

[51] L. Szirmay-Kalos, I. Georgiev, M. Magdics, B. Molnár, and D. Légrády. Unbiased light transport estimators for inhomogeneous participating media. 36(2):9–19, 2017.

[52] L. Szirmay-Kalos, M. Magdics, and M. Sbert. Multiple scattering in inhomogeneous participating media using rao-blackwellization and control variates. *Computer Graphics Forum*, 37(2):63–74, 2018.

[53] L. Szirmay-Kalos, B. Tóth, and M. Magdics. Free path sampling in high resolution inhomogeneous participating media. *Computer Graphics Forum*, 30(1):85–97, 2011.

[54] T. Wilde, C. Rössl, and H. Theisel. FTLE ridge lines for long integration times. In *Proceedings IEEE Scientific Visualization Short Papers*, page to appear, 2018.

[55] 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*, volume 557. Argonne National Laboratory, 1965.

[56] Y. Yue, K. Iwasaki, B.-Y. Chen, Y. Dobashi, and T. Nishita. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Transaction on Graphics (Proc. SIGGRAPH Asia)*, 29(6):177:1–177:8, 2010.

[57] C. Yuksel and J. Keyser. Deep opacity maps. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2008)*, 27(2):675–680, 2008.

[58] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon. Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum (Proc. Eurographics)*, 34(2):667–681, 2015.