

Image-based Visualization of Large Volumetric Data Using Moments

Tobias Rapp, Christoph Peters, and Carsten Dachsbaucher

Abstract—We present a novel image-based representation to interactively visualize large and arbitrarily structured volumetric data. This image-based representation is created from a fixed view and models the scalar densities along each viewing ray. Then, any transfer function can be applied and changed interactively to visualize the data. In detail, we transform the density in each pixel to the Fourier basis and store Fourier coefficients of a bounded signal, i.e. bounded trigonometric moments. To keep this image-based representation compact, we adaptively determine the number of moments in each pixel and present a novel coding and quantization strategy. Additionally, we perform spatial and temporal interpolation of our image representation and discuss the visualization of introduced uncertainties. Moreover, we use our representation to add single scattering illumination. Lastly, we achieve accurate results even with changes in the view configuration. We evaluate our approach on two large volume datasets and a time-dependent SPH dataset.

Index Terms—Image-based visualization, volume rendering, unstructured volumes, moments, MESE, Fourier reconstruction

1 INTRODUCTION

CONTINUING progress in high performance computing enables scientists to perform complex simulations in a high spatial and temporal resolution. Exploring and analyzing the resulting datasets is a daunting task. The traditional approach to transfer and analyze the data on an individual workstation is no longer feasible for large datasets due to storage, bandwidth, and compute constraints. *In situ* visualization [1], where the simulation is tightly coupled with the visualization pipeline, addresses this limitation, but limits exploration and interaction.

Alternatively, data reduction [2] can be used to compress, subsample, or to construct an approximate representation of the data. Loss of information is thereby unavoidable to achieve a meaningful data reduction, thus requiring the communication of introduced uncertainties. Image-based approaches [3] combine *in situ* visualization with the analysis on low-cost machines. There, the data is represented by images produced from fixed views and predefined parameters. Image-based approaches are also employed to render large or unstructured volumetric data [4] that could not be interactively visualized otherwise. However, the transfer function based exploration of volumetric data is difficult to integrate into image-based approaches as it requires a compact and accurate representation of the signal along each view ray.

In this work, we present a novel image-based data representation to visualize large structured and unstructured volumetric data. This representation allows us to store and reconstruct the scalar density along a viewing ray, thus enabling the application to arbitrarily structured data and a change of all aspects of the transfer function similar to recent work [5], [6], [7], [8]. We avoid storing discrete samples or modeling distributions along a ray. Instead we compactly represent the density in the Fourier basis. We store Fourier coefficients of a signal bounded between zero and one, also referred to as bounded trigonometric moments. This leads to a sparse and quantizable representation that can be linearly

interpolated in space and time. For reconstruction, we explore two options. Evaluating a truncated Fourier series is simple and efficient but causes well-known ringing artifacts. The recently introduced bounded maximum entropy spectral estimate (MESE) [9] offers an efficient and accurate reconstruction of a bounded signal from its moments.

We compress this representation in several ways. First, we determine the number of moments per pixel adaptively. To this end, we ray march the dataset once and generate a fixed number of moments. Then we select an appropriate number of moments per pixel by utilizing an error measure between all and a prefix of moments. Next, we employ a novel coding strategy to compress the moments in each pixel. This coding scheme uses information from the lower-order moments in a pixel to predict the next moment. Therefore, we only need to store differences to the actual moments, which are more amenable to quantization and compression. Accounting for the trade-off between quality and data size, we present a parameterizable quantization curve that is Pareto optimal. After compression, we store between 20 and 60 bytes per pixel, when using up to 100 moments. Furthermore, we quantify the uncertainty of this representation, include single scattering in the volume rendering, and support changes in the view configuration.

To summarize, our main contributions are:

- We propose an image-based representation for visualizing large, arbitrarily structured volumetric data,
- We adaptively determine the number of moments per pixel and present a novel coding and quantization strategy to compactly store bounded trigonometric moments,
- We describe methods for uncertainty quantification, single scattering, and changing the view configuration with our image-based representation.

• T. Rapp, C. Peters, and C. Dachsbaucher are with the Karlsruhe Institute of Technology, Germany.
E-mail: {tobias.rapp, christoph.peters, dachsbaucher}@kit.edu

Manuscript received 22 Oct. 2021; revised 14 Jan. 2022; accepted 9 Feb. 2022.

We make the full source code of our implementation available at <https://github.com/TobiasRp/mray>.

2 RELATED WORK

Data reduction is increasingly important for the visualization and analysis of scientific datasets [2]. Due to the stochastic nature of continuous scientific data, lossless and near lossless compression is generally unable to significantly reduce the data sizes. Moreover, we are not just interested in compression, but also in the interactive visualization and exploration of large data. For rendering structured volumes, compression [10] and multi-resolution [11] approaches exist. A popular family of lossy compression methods applies transformations, such as the Fourier or discrete cosine transforms [12], custom transforms [13], or transforms based on tensor decomposition [14]. These transformations are followed by quantization and encoding of the coefficients to achieve compression.

We similarly employ a transformation followed by a compact encoding of the resulting coefficients, but our signals are one-dimensional. This is due to our image-based perspective, where we only compress the subset of the dataset visible from a fixed view. This approach leads to a compact image representation, with an acceptable loss in accuracy, and enables its application to unstructured data. Moreover, by limiting our approach to one-dimensional signals, it is significantly faster and scalable, which is required for the *in situ* application.

Our method follows the image-based paradigm first introduced by Ahrens et al. [3]. The authors present a framework for *in situ* visualization of extreme scale data by collecting and organizing a large database of images taken with different parameters. The authors point out that storing a massive amount of images, e.g. in the order of 10^6 , still leads to a reduced amount of data compared to state-of-the-art simulations, which are in the order of $\geq 10^{15}$. Lukasczyk et al. [15] determine a minimal set of views that best approximates the data. Bujack et al. [16] detect features *in situ* to create clutter-free images. Deferred rendering [17] can be used to adjust some of the visualization parameters for surface rendering. However, with all of these approaches it is not possible to explore volumetric data by changing the transfer function. Tikhonova et al. [18], [19], [20] thus propose explorable images that store image slices to allow modifying the transfer function by solely relying on image-space operations. In contrast to explorable images, we propose an image-based representation that enables the reconstruction of scalar densities during ray marching.

An image-based approach for volume rendering has been introduced by Mueller et al. [21], which employs pre-rendered object slices, to address growing data sizes. Shareef et al. [4] perform volume rendering on unstructured grids by representing the scene with view-dependent information. This representation is based on layered depth images [22], where each pixel contains a list of depth-ordered samples. Volumetric depth images [5] store scalar samples along rays in image-space, but additionally partition them by similarity into larger segments. Volumetric depth images have been extended to space-time [6], by exploiting both inter-ray and inter-frame coherence. These approaches thus work best when the data is mostly homogeneous since storing discrete samples does not scale well to complex signals. Wang et al. [7] partition each ray into segments, but subdivide adaptively based on the Shannon entropy of ray densities. This requires repeated ray marching of the data to determine a suitable subdivision. In each segment, the density distribution is approximated with a histogram, which is storage intensive, disregards depth-ordering, and quantizes the values. Their work has been extended for time-varying data [8]

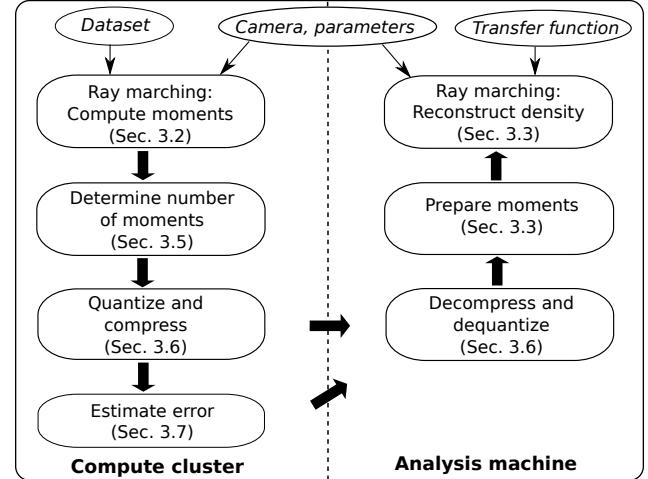


Fig. 1: A moment image is generated on the compute cluster (left) and is used for interactive volume rendering on the analysis machine (right).

where bandwidth or storage constraints limit the total amount of time steps. Temporal coherence is improved by storing ray histograms and depth-ordering information in-between time steps.

In this paper, we employ bounded trigonometric moments to compactly and efficiently represent bounded densities per pixel. This representation implicitly retains the depth order and thus avoids the need for dividing the ray segment or storing additional data structures. Although we perform aggressive quantization, our reconstructed signals are smooth. Lastly, generating our representation does not require repeated ray marching of the dataset, which makes it well suited for large and irregular data that is costly to sample.

In graphics research, the theory of moments has been applied to reconstruct densities from power moments or Fourier coefficients. This has led to new approaches to shadow mapping [23], [24], transient imaging [25], spectral rendering [9], and order-independent transparency [26]. Recently, these results have been applied to visualization research for rendering lines with transparency [27] and for opacity optimization [28], [29]. In this work, we employ the approach by Peters et al. [9] to reconstruct bounded densities and apply it to direct volume rendering. This requires us to use orders of magnitude more moments and to devise novel compaction and compression methods to encode them efficiently. Although we derive these methods based on the theory of moments, they are equally applicable when using the Fourier reconstruction.

3 MOMENT-BASED VOLUME RENDERING

In this section, we discuss volume rendering using our image-based representation. Figure 1 gives an overview. Our proxy representation, which we refer to as moment image, is generated on a system with access to the data. The moment image then enables the interactive exploration with a user-controllable transfer function on a low-cost analysis machine.

We begin by reviewing the reconstruction of bounded densities using moments (Section 3.1), before we discuss the computation of moments during ray marching of arbitrarily structured data (Section 3.2) to create a moment image. Then, we examine ray marching using a moment image (Section 3.3). Afterwards, we analyze relations between valid sequences of moments (Section 3.4).

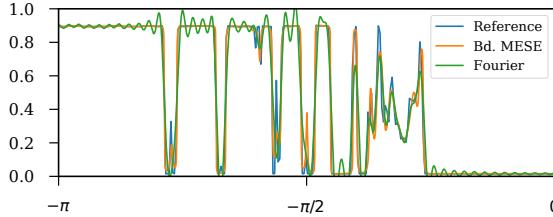


Fig. 2: Truncated Fourier and bounded MESE reconstruction from the same coefficients.

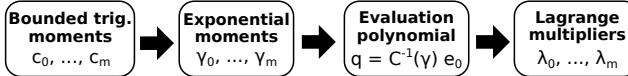


Fig. 3: Given bounded trigonometric moments, we compute the corresponding exponential moments, solve a linear system, and finally compute Lagrange multipliers, from which we can efficiently reconstruct a bounded signal.

We employ these insights to select an appropriate number of moments for each pixel (Section 3.5) and for our coding and quantization scheme (Section 3.6).

3.1 Using Moments to Reconstruct Bounded Densities

To represent the densities along a viewing ray, we map the length of the ray in the volume to $[-\pi, 0]$ linearly and we assume that the densities are bounded in $[0, 1]$. Since this signal is generally not periodic, we mirror the signal in $[-\pi, 0]$ to a periodic signal in $[-\pi, \pi]$. With these conventions, our goal is compact storage of a signal $s : [-\pi, \pi] \rightarrow [0, 1]$. We represent this signal in the Fourier basis using $m + 1 \in \mathbb{N}$ Fourier coefficients. The Fourier basis, written as a vector, is

$$\mathbf{c}(\varphi) := \frac{1}{2\pi} (\exp(-ij\varphi))_{j=0}^m \in \mathbb{C}^{m+1}$$

and the Fourier coefficients are

$$c := \int_{-\pi}^{\pi} s(\varphi) \mathbf{c}(\varphi) d\varphi \in \mathbb{R}^{m+1}.$$

Note that the coefficients are real since the mirroring makes the signal even. These Fourier coefficients compactly approximate a signal. However, reconstruction with a truncated Fourier series does not guarantee bounded values and exhibits well-known ringing artifacts as shown in Figure 2.

The Fourier coefficients of a bounded signal are also known as bounded trigonometric moments, or simply moments in the scope of this paper. Peters et al. [9] introduce the bounded MESE, which reconstructs a signal from bounded trigonometric moments. It never violates the bounds $[0, 1]$ and is less prone to ringing, see Figure 2. We briefly recapitulate the underlying algorithm [9] here because its intermediate results are important to our novel quantization schemes. Figure 3 gives an overview.

The bounded MESE is derived from a reconstruction for signals with values in $[0, \infty)$ known as MESE. The two signals are in a dual one-to-one relationship, defined by an integral transform [9]. Thus, we first transform the bounded trigonometric moments c into exponential moments $\gamma \in \mathbb{C}^{m+1}$ of its dual unbounded signal. For this we define

$$\check{\gamma}_0 := \frac{1}{4\pi} \exp\left(\pi i \left(c_0 - \frac{1}{2}\right)\right) \in \mathbb{C}.$$

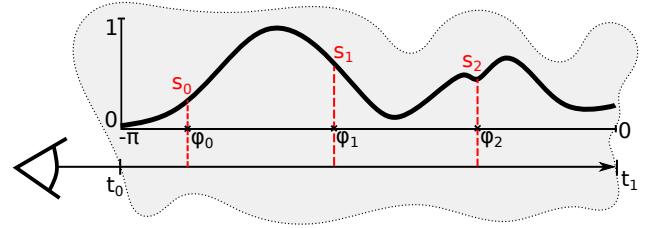


Fig. 4: During ray marching, we sample a scalar density along the ray parameterized by $[-\pi, 0]$. We compute $m + 1$ moments from this signal.

The exponential moments are defined by the recurrence

$$\begin{aligned} \gamma_0 &:= 2\Re \check{\gamma}_0 \in \mathbb{R}, \\ \gamma_l &:= \frac{2\pi i}{l} \left(l \check{\gamma}_0 c_l + \sum_{j=1}^{l-1} (l-j) \gamma_j c_{l-j} \right) \in \mathbb{C}, \end{aligned} \quad (1)$$

where $\Re \check{\gamma}_0$ denotes the real part and $l \in \{1, \dots, m\}$. This transformation is invertible.

The next steps utilize the Hermitian Toeplitz matrix

$$C(\gamma) := \frac{1}{2\pi} (\gamma_{j-k})_{j,k=0}^m \in \mathbb{C}^{(m+1) \times (m+1)},$$

where $\gamma_{-j} = \overline{\gamma_j}$ for all $j \in \{1, \dots, m\}$. It is possible to reconstruct a density with values in $[0, \infty)$ from the exponential moments γ if and only if $C(\gamma)$ is positive definite. In this case, the evaluation polynomial $q := C^{-1}(\gamma) e_0$, with $e_0 := (1, 0, \dots, 0)^T$ holds coefficients for the MESE [25].

An integral transform converts the MESE of the exponential moments γ into the bounded MESE of the bounded trigonometric moments c ; it can be efficiently evaluated [9]. We transform the evaluation polynomial into Lagrange multipliers $\lambda \in \mathbb{R}^{m+1}$. We denote $\check{\gamma}_j = \gamma_j$ for all $j \in \{1, \dots, m\}$. Then for all $l \in \{0, \dots, m\}$

$$\lambda_l := \frac{1}{\pi i q_0} \sum_{k=0}^{m-l} \check{\gamma}_k \sum_{j=0}^{m-k-l} \overline{q_{j+k+l}} q_j \in \mathbb{R}. \quad (2)$$

Now, the bounded MESE is given by

$$\hat{s}(\varphi) = \frac{1}{\pi} \arctan \left(\Re \lambda_0 + 2\Re \sum_{l=1}^m \lambda_l \exp(-il\varphi) \right) + \frac{1}{2}. \quad (3)$$

Since arctan maps to $(-\frac{\pi}{2}, \frac{\pi}{2})$, the reconstructed density \hat{s} is always in $(0, 1)$, as intended. Moreover, the Lagrange multipliers are constructed to ensure that

$$\int_{-\pi}^{\pi} \hat{s}(\varphi) \mathbf{c}(\varphi) d\varphi = c, \quad (4)$$

i.e. the bounded trigonometric moments are accounted for exactly. The ground truth s and its reconstruction \hat{s} may still differ but at least the stored Fourier coefficients agree perfectly. A truncated Fourier series also satisfies Equation 4 but does not exploit knowledge about the bounds. As shown in Figure 2 the bounded MESE captures complicated signals well while being less prone to ringing than a truncated Fourier series.

3.2 Moments of Ray Densities

We now discuss the creation of our proxy representation, the moment image. For every pixel, this image contains the moments of a scalar density sampled during ray marching, see Figure 4. To

this end, we first compute the intersections t_0 and t_1 of a viewing ray $r(t)$ with the volume. We map the parameterized ray $r(t)$ with $t \in [t_0, t_1]$ to phase space $[-\pi, 0]$ linearly and sample the volume at $\varphi_0, \dots, \varphi_{n-1}$, giving us the densities $s_0, \dots, s_{n-1} \in [0, 1]$. Since the actual method that produces these densities is not relevant, we can sample arbitrarily structured volumetric data.

To compute $m+1$ bounded trigonometric moments during ray marching, we assume linear interpolation and perform a quadrature [9, Appendix C]. At each ray marching step, we compute the gradient a_l and y-intercept b_l :

$$a_l := \frac{s_{l+1} - s_l}{\varphi_{l+1} - \varphi_l}, \quad b_l := s_l - a_l \varphi_l.$$

Then the interpolated signal at $\varphi \in [\varphi_l, \varphi_{l+1}]$ is $s(\varphi) := a_l \varphi + b_l$. The bounded trigonometric moments are iteratively computed as:

$$c_0 := \frac{1}{\pi} \sum_{l=0}^{n-2} \left[\frac{a_l}{2} \varphi^2 + b_l \varphi \right]_{\varphi_l}^{\varphi_{l+1}} \in \mathbb{R},$$

$$c_j := \frac{1}{\pi} \Re \sum_{l=0}^{n-2} \left[\left(a_l \frac{1+i j \varphi}{j^2} + b_l \frac{i}{j} \right) \exp(-i j \varphi) \right]_{\varphi_l}^{\varphi_{l+1}} \in \mathbb{R}.$$

Since we mirror the signal, the moments are real and we do not have to compute or store the imaginary parts.

Lastly, we compute a lower bound $s_{\min} \leq s(\varphi) - \varepsilon$ and an upper bound $s(\varphi) + \varepsilon \leq s_{\max}$ during ray marching. Here, ε is a small number to relax the bounds. For example, we set $\varepsilon = 0.005$ for our datasets. This relaxation improves the reconstruction since it stays away from 0 and 1, see Equation 3. After ray marching, we transform the moments to these bounds via

$$c'_0 := \frac{c_0 - s_{\min}}{s_{\max} - s_{\min}}, \quad c'_j := \frac{c_j}{s_{\max} - s_{\min}}. \quad (5)$$

Although we have to store the bounds for each pixel to invert this transformation after reconstruction, we found that it improves the reconstruction and enables a more aggressive quantization of the moments (Section 3.6).

3.3 Interactive Rendering

To render using a moment image, we first note that it is possible to linearly interpolate the moments of neighboring pixels or even between different time steps. For example, we can increase the resolution of the moment image using bilinear interpolation to render in a higher resolution. Although other types of interpolation are possible, negative weights in filter kernels may invalidate moments and should be avoided.

To ray march a moment image, we compute the Lagrange multipliers for each pixel upon loading. Subsequently, we perform ray marching using a ray $r(t)$ in the interval $[t_0, t_1]$, which we map to $[-\pi, 0]$. During ray marching, we use the Lagrange multipliers to efficiently reconstruct a density at each $\varphi \in [-\pi, 0]$ by evaluating Equation 3. Afterwards, we invert the transformation from Equation 5 using the bounds $[s_{\min}, s_{\max}]$, apply a transfer function, and perform compositing. Note that we can apply any transfer function that expects a scalar density as input. We employ a preintegrated transfer function [30]. Ray marching a moment image is fast, taking only a few milliseconds (cf. Table 3), which enables the interactive exploration with different transfer functions.

3.4 Relations Between Moments

It is viable to store moment images using one 32-bit float per moment but storage requirements are considerable. Thus, we strive to reduce the number of moments adaptively and to quantize the remaining moments to only a few bits. Our methods benefit from the underlying theory of the bounded MESE [9] but they are novel and independent of the used reconstruction. The bounded MESE and the truncated Fourier series both benefit. In this section, we cover the relevant mathematical results.

Recall from Section 3.1 that we construct the evaluation polynomial $q := C^{-1}(\gamma) e_0$ from the exponential moments $\gamma \in \mathbb{C}^{m+1}$. The Toeplitz matrix C has a special structure. Levinson's algorithm exploits this structure to solve for q in time $\mathcal{O}(m^2)$ instead of $\mathcal{O}(m^3)$ [31]. At the same time, it produces intermediate results that aid our quantization scheme. For all $l \in \{1, \dots, m\}$ Levinson's algorithm computes:

$$q_0^{(0)} := \frac{1}{\gamma_0}, \quad (6)$$

$$u^{(l)} := \sum_{k=0}^{l-1} q_k^{(l-1)} \gamma_{-k}, \quad (7)$$

$$q^{(l)} := \frac{\left(q_0^{(l-1)}, \dots, q_{l-1}^{(l-1)}, 0 \right) - u^{(l)} \left(0, \overline{q_{l-1}^{(l-1)}}, \dots, \overline{q_0^{(l-1)}} \right)}{1 - |u^{(l)}|^2}. \quad (8)$$

Then

$$q = C^{-1}(\gamma) e_0 = 2\pi \left(q_0^{(m)}, \dots, q_m^{(m)} \right).$$

Since the Toeplitz matrix is positive definite, we know $|u^{(l)}| < 1$ [31]. Combined with Equation 7, this inequality forces γ to reside in a disk of radius r_l with center $\hat{\gamma}$, where

$$r_l := \frac{1}{q_0^{(l-1)}} > 0, \quad \hat{\gamma} := -\frac{1}{q_0^{(l-1)}} \sum_{k=1}^{l-1} q_k^{(l-1)} \gamma_{-k} \in \mathbb{C}.$$

Inverting Equation 1 shows that the bounded trigonometric moment

$$c_l = \frac{\gamma_l}{2\pi i \hat{\gamma}_0} - \frac{1}{l \hat{\gamma}_0} \sum_{j=1}^{l-1} (l-j) \gamma_j c_{l-j} \quad (9)$$

lies in a disk as well. We base our coding strategy in Section 3.6 on this observation.

Incidentally, the center of this disk also has a compelling relation to the bounded MESE \hat{s} . In the case $l = m+1$, we find

$$\hat{c}_l := \frac{\hat{\gamma}_l}{2\pi i \hat{\gamma}_0} - \frac{1}{l \hat{\gamma}_0} \sum_{j=1}^{l-1} (l-j) \gamma_j c_{l-j} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{s}(\varphi) \exp(-il\varphi) d\varphi.$$

In other words, the bounded MESE places every unknown moment in the center of the disk where it must reside [9, Lemma B.2]. This behavior is in stark contrast to a truncated Fourier series, which just sets unknown Fourier coefficients to zero.

Applying this insight repeatedly lets us compute the full Fourier expansion of the bounded MESE [9, Proposition B.3]. In terms of the exponential moments, we obtain the linear recurrence

$$\gamma_{m+1+k} = -\frac{1}{q_0} \sum_{j=0}^m \gamma_{j+k} q_{m+1-j}, \quad (10)$$

for all $k \in \mathbb{N}$. Mapping these exponential moments to bounded moments through Equation 9, we obtain all unknown moments of the bounded MESE \hat{s} .

3.5 Determining the Number of Moments

To adaptively reduce the number of moments per pixel, we measure the error between a prefix of c_0, \dots, c_n moments and the full vector of moments c_0, \dots, c_m , where $n < m$. Based on the assumption that the full set of $m+1$ moments accurately captures the signal, we compute the error without accessing the original data or ray marching the dataset again.

To this end, we use the recurrence from Equation 10 and Equation 9 to compute the missing $m-n$ moments from the prefix of $n+1$ moments. In this manner, we obtain the exact moments $\hat{c}_{n+1}, \dots, \hat{c}_m$ of the bounded MESE \hat{s} , assuming that the moments c_{n+1}, \dots, c_m have been discarded. We measure the error introduced by discarding these moments using the relative RMSE

$$\text{rRMSE}((c_{n+1}, \dots, c_m), (\hat{c}_{n+1}, \dots, \hat{c}_m)) = \frac{1}{c_0} \sqrt{\sum_{i=n+1}^m (c_i - \hat{c}_i)^2}.$$

The rRMSE normalizes the root mean squared error (RMSE) with respect to the average value of the signal, i.e. the zeroth moment c_0 .

To determine the number of moments for each pixel, we find a value of n so that the error is just below a user-defined threshold. We keep only the first $n+1$ moments per pixel to create a compact moment image. Computing the moments of the bounded MESE $\hat{c}_{n+1}, \dots, \hat{c}_m$ takes time $\mathcal{O}(m(m-n))$ and we have to redo this work for each value of n that we try. To keep the overall cost low, we use bisection. It finds a suitable n in $\mathcal{O}(\log m)$ trials. Since the error is not guaranteed to decrease monotonically with n , the found n is not known to be minimal but it certainly satisfies the requested error threshold.

3.6 Compression and Quantization

Moment images might be produced and archived in large quantities and are transferred over network to the analysis machine. Therefore, small file sizes are paramount. A baseline approach directly quantizes bounded trigonometric moments $c_0 \in [0, 1]$ and $c_i \in [-\frac{1}{\pi}, \frac{1}{\pi}]$. Since the representation of moments is essential to reduce storage requirements, we propose a novel coding scheme, a Pareto optimal quantization curve, and discuss lossless compression to pack more information into significantly fewer bits.

3.6.1 Coding

In Section 3.4 we observed that each exponential moment γ_l lies in a disk characterized by the previous moments $\gamma_0, \dots, \gamma_{l-1}$. We exploit this constraint in our coding strategy by only storing the difference to the center of the disk. These differences exhibit lower entropy compared to the moments and are better suited for quantization and compression.

First, we explicitly store c_0 , then we transform the bounded trigonometric moments c to exponential moments γ . We execute Levinson's algorithm and at each step store the difference between γ_l and the center of the disk $\dot{\gamma}$ given by the previous exponential moments, scaled relative to the radius r_l :

$$u^{(l)} = \frac{\gamma_l - \dot{\gamma}}{r_l} \in \mathbb{C}.$$

To revert this encoding, we simply execute Levinson's algorithm with γ_0 and the stored values of $u^{(l)}$ to solve the system for q . Furthermore, we reconstruct the exponential moments as

$$\gamma_l = \dot{\gamma} + r_l u^{(l)}.$$

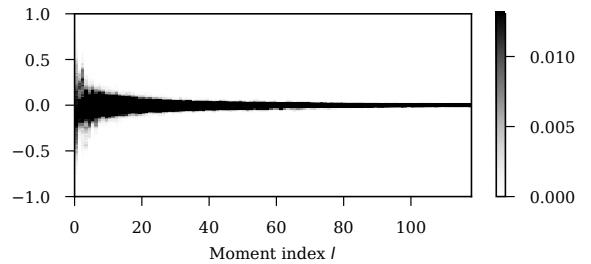


Fig. 5: Distribution of values $u_T^{(l)}$ that we store for each moment at index l .

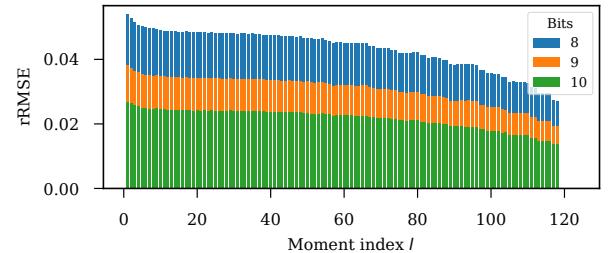


Fig. 6: We quantize only a single moment index l at a time and measure the resulting error. This shows that the error from quantization depends on the index l .

3.6.2 Transformation

We perform two important transformations that enable a more aggressive quantization. First, for real moments, the corresponding $u^{(l)} \in \mathbb{C}$ with $l > 0$ vary only along the axis aligned with $i\dot{\gamma}_0$. We transform $u^{(l)}$ to the real axis:

$$u_T^{(l)} := u^{(l)} \frac{|i\dot{\gamma}_0|}{i\dot{\gamma}_0} \in (-1, 1).$$

The values for $u_T^{(l)}$ are distributed mostly around zero, see Figure 5. Therefore, we compute $\min(u_T^{(l)})$ and $\max(u_T^{(l)})$ for each $l \in [1, m]$ once per moment image and transform the $u_T^{(l)}$ from this range to $(0, 1)$ before quantization. Especially for large l , this transformation leads to a significantly more accurate representation.

3.6.3 Quantization

First, we quantize the zeroth moment c_0 to 16-bit in $[0, 1]$. Although we might be able to use fewer bits in some cases, the quality can deteriorate quickly. In general, quantization errors are amplified from coefficients with a lower to a higher index. This is shown in Figure 6, where we apply quantization to only a single moment at a time. Then, we measure the rRMSE between the bounded trigonometric moments of the quantized and the original image. Consequently, this shows how much each moment influences the error of the reconstruction. We use this observation to determine the number of bits for quantizing a moment at index $l \in \{1, \dots, m\}$. Specifically, we want to quantize such that at each index we apply approximately the same error. The quantization curve, i.e. the number of bits used to quantize each index is thus determined from the errors shown in Figure 6.

Depending on the trade-off between acceptable error and image size, we want to use a different quantization curve. By selecting the number of bits b_1 to quantize the moment at index 1, the resulting error determines the whole curve. We thus employ b_1

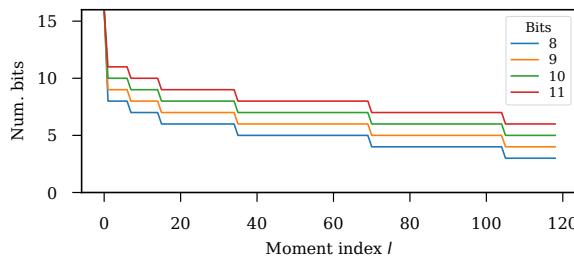


Fig. 7: Quantization curves for different parameter values.

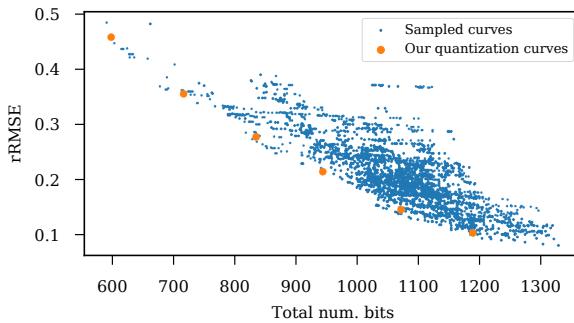


Fig. 8: Comparison of our Pareto optimal quantization curves (see Figure 7) and randomly perturbed curves.

as a parameter to create different quantization curves. For each subsequent index l , the algorithm tries to decrease the number of bits as much as possible, but stays below the error threshold determined from b_1 . The algorithm is described in detail in the supplemental document.

We illustrate several quantization curves in Figure 7. To determine whether these curves are optimal, we sample randomly perturbed quantization curves and plot the total number of bits against the error in Figure 8. This shows that our proposed quantization curves are Pareto optimal, i.e. no change of the curve leads to a reduction in both error and size.

Although the quantization curves are always qualitatively similar, they still differ between different datasets and view configurations. We thus determine an optimal quantization curve for each moment image. For images with a large resolution, this computation can be quite expensive. Therefore, we recommend to use a downsampled image of size 64×64 , which gives nearly identical results and is fast to compute.

3.6.4 Compression

To further reduce the size of a moment image, we apply lossless compression to the quantized coefficients. Figure 5 shows that the frequency of values for each moment is non-uniform. Thus, entropy encoding can reduce the data size by storing variable-length codewords for each coefficient at index l containing 2^{p_l} symbols. The length of the codewords is selected proportional to the frequency of occurrence. We employ arithmetic coding [32], which can estimate the frequencies during encoding. Although the compression rate depends on the dataset and quantization, we achieve a 20 – 40% reduction in size. Lastly, we perform fast dictionary coding on the resulting byte stream using LZ4, further reducing the image size.

4 APPLICATIONS

Now that we have a compact representation, we discuss additional applications. We propose a principled way to visualize uncertainty introduced by the compression (Section 4.1), implement single scattering for a more compelling visualization (Section 4.2), and exploit the volumetric nature of our representation to enable changes to the view configuration (Section 4.3).

4.1 Uncertainty Quantification

Since our approach introduces information loss, we want to convey the resulting uncertainty. For example, we visualize the uncertainty using a heat map, temporal animation [33], [34], or by integrating the uncertainty in the transfer function classification [35].

Due to space constraints, we are limited to few statistics about the distribution of errors along a viewing ray in each pixel. Computing these statistics is the only step, and an optional one, of our method that requires ray marching the dataset a second time. In detail, at each step during ray marching, the signal is reconstructed from our representation and is compared to the original dataset.

As statistics, we propose to compute a single or select percentiles of the error. These order statistics can be used to create robust and expressive error bounds. However, order statistics are expensive to compute and would require $\mathcal{O}(n)$ space during ray marching, which is problematic for parallel computation on a cluster or GPU. Therefore, we employ the heuristic P^2 -algorithm [36] to perform online estimation of a percentile with fixed storage requirements.

4.2 Single Scattering

Volumetric shadows can improve the perception of spatial depth in direct volume visualizations [37]. Although different illumination models exist, the physically-based single scattering model [38] is often employed. At each step during ray marching, this requires evaluating the transmittance to the light source.

To incorporate single scattering, we create a moment image from the perspective of the light source similar to shadow mapping [39]. Moreover, we recommend to use a smaller number of moments and a lower resolution since single scattering illumination is generally of lower frequency. During rendering, we could ray march along secondary rays from the single scattering moment image to each sample point, but this is computationally expensive. Instead, we ray march the single scattering image, compute the transmittance, and cache it on a regular grid. Then, we sample the cached transmittance during ray marching.

4.3 View Projection

The method presented thus far is limited to a static camera and this is the use case we are primarily interested in. However, a moment image encodes the entire volume within the view frustum. We exploit this insight to allow for changes in the view configuration. For example, interactively adjusting the viewing angle can improve depth perception. However, visible data in the moment image as well as aliasing due to the discrete moment image resolution indicates the limits of this approach.

To render a changed view, we ray march starting from the changed camera and project each point that is inside the view frustum to the moment image. Then, we perform bilinear interpolation of the Lagrange multipliers and reconstruct a density. The interpolation makes the rendering quite expensive, but is

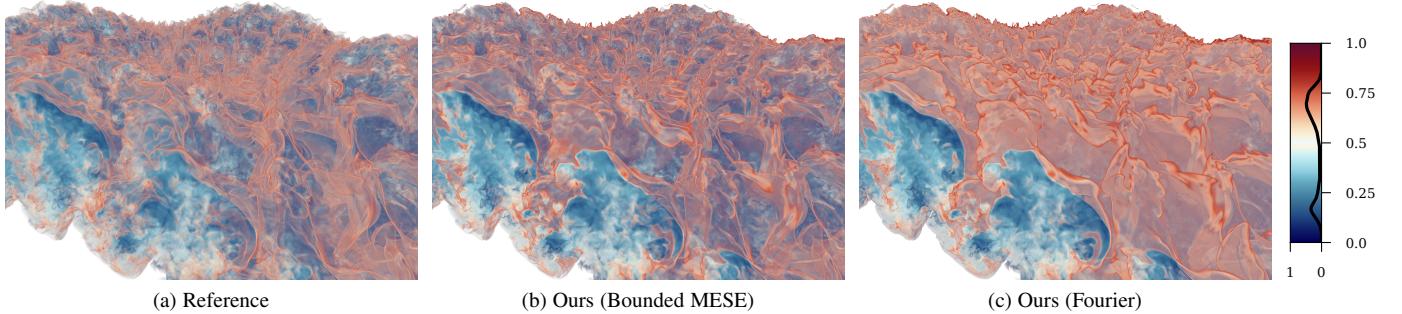


Fig. 9: Entropy field of a Richtmyer-Meshkov instability rendered with direct volume rendering (a), with our approach using the bounded MESE reconstruction (b), and using the Fourier reconstruction (c).

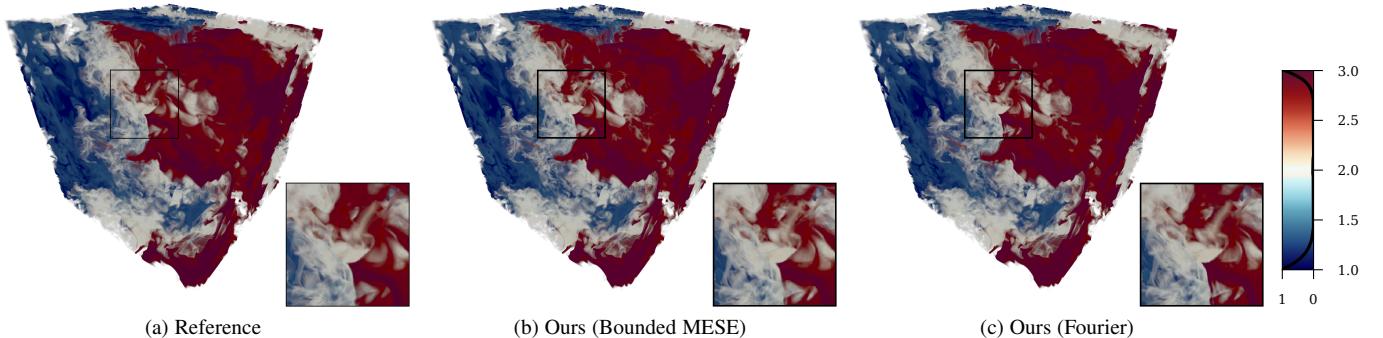


Fig. 10: The Rayleigh-Taylor instability shows the density of two mixing fluids. We render a ray marching reference (a), our approach with the bounded MESE reconstruction (b), and with Fourier reconstruction (c).

necessary to avoid aliasing artifacts. For faster rendering at the expense of memory requirements, we resample a moment image to a regular grid. To avoid aliasing, we either require a sufficiently high image resolution or we upsample the moment image. We use this approach in our supplemental video.

5 EVALUATION

We evaluate our approach on three real-world datasets. The entropy field of a Richtmyer-Meshkov instability (Figure 9) is given in a resolution of $2048 \times 2048 \times 1920$ with 8 bits per cell. The Rayleigh-Taylor instability dataset (Figure 10) consists of a density field showing two mixing fluids. It is stored in single-precision in a resolution of 1024^3 . Lastly, the turbine dataset stems from a smoothed-particle hydrodynamics (SPH) simulation of 100 million particles per time step, where each particle consists of a position and a scalar value in single-precision. We sample the scattered data by performing SPH interpolation with a cubic spline kernel. We employ a uniform grid for accelerating the neighborhood search during ray marching.

We evaluate our approach on these datasets qualitatively (Section 5.1), quantitatively (Section 5.2), and discuss the performance (Section 5.3). We compare the bounded MESE with the truncated Fourier reconstruction, which are both applicable to our moment images. Furthermore, we compare our approach to the ray-histogram approach by Wang et al. [7] and to ZFP [13], in fixed-accuracy mode, and TTHRESH [14]. To apply the latter two techniques, we compress the three-dimensional perspective grid that is defined by the image plane and the samples along each ray.

This enables a direct comparison to our approach since this grid is similarly limited to visible data and the resolution is increased near the camera. Note that both ours as well as the ray-histogram approach sparsely sample the three-dimensional grid, but do not explicitly store it.

If not noted otherwise, we use an image resolution of 1024×768 with a maximum of 100 moments per pixel. We employ our novel coding technique and quantize the moments as discussed in Section 3.6 with the quantization curve given by $b_1 = 10$. We further evaluate the impact of our coding and quantization scheme, the number of moments, the introduced uncertainty, and the scalability of our method in the supplemental document.

5.1 Qualitative Evaluation

The Richtmyer-Meshkov instability is shown in Figure 9 using direct volume rendering (a), our approach using the bounded MESE (b), and with the Fourier reconstruction (c). This complicated dataset is difficult to represent, see e.g. the reconstruction of a single pixel in Figure 2. Although the Fourier reconstruction generally leads to good results, it introduces strong ringing artifacts for this dataset. The bounded MESE performs better. Figure 10 shows the Rayleigh-Taylor instability. Here, the Fourier method and the bounded MESE produce visually similar results.

The turbine dataset is shown in Figure 11 using various techniques. The transfer function, illustrated on the right, maps low and high velocities to non-zero opacities. This reveals the rotating turbine blades and indicates the presence of several vortices. Whilst our approach shows no obvious artifacts, the ray-histogram in (d)

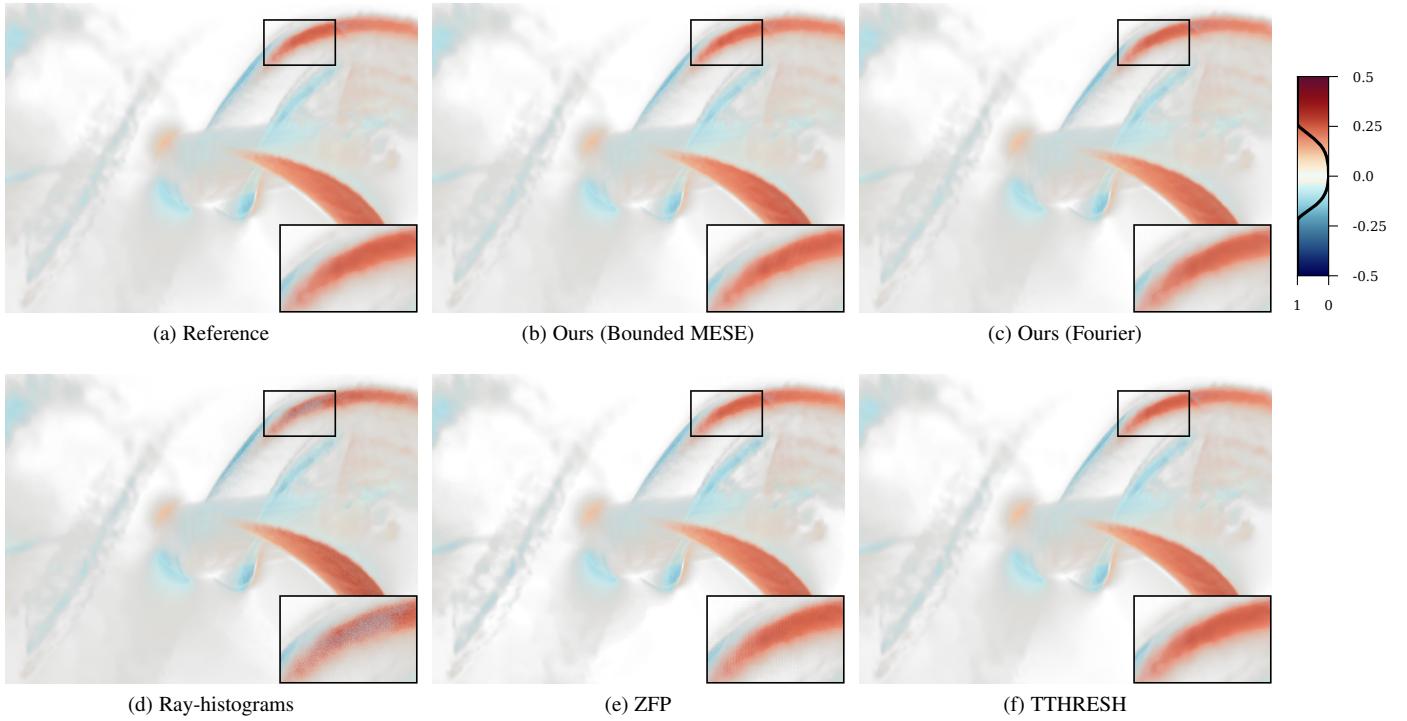


Fig. 11: The turbine dataset with direct volume rendering (a), using our approach with the bounded MESE (b) and Fourier reconstruction (c), ray-histograms (d), and by compressing the perspective grid using ZFP (e) and TTHRESH (f).

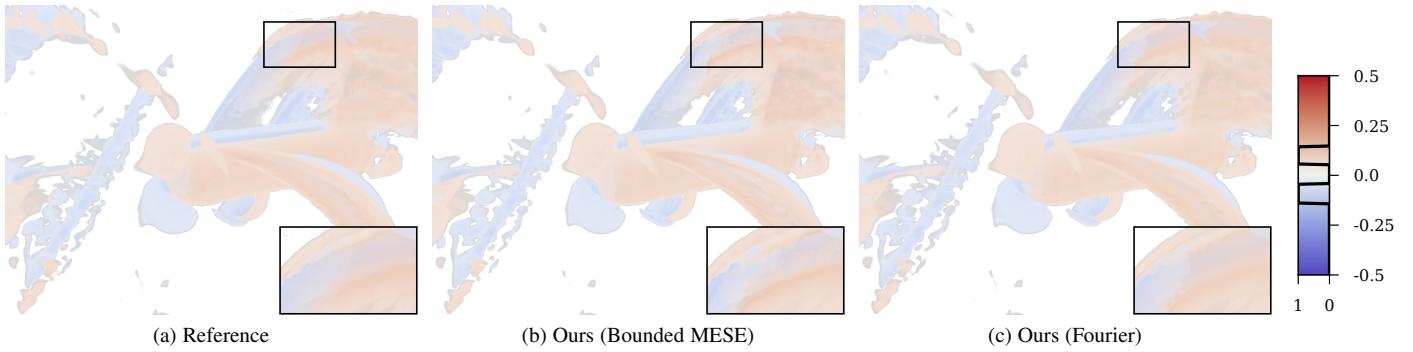


Fig. 12: We render the turbine dataset with direct volume rendering (a), using our approach with the bounded MESE (b), and with Fourier reconstruction (c).

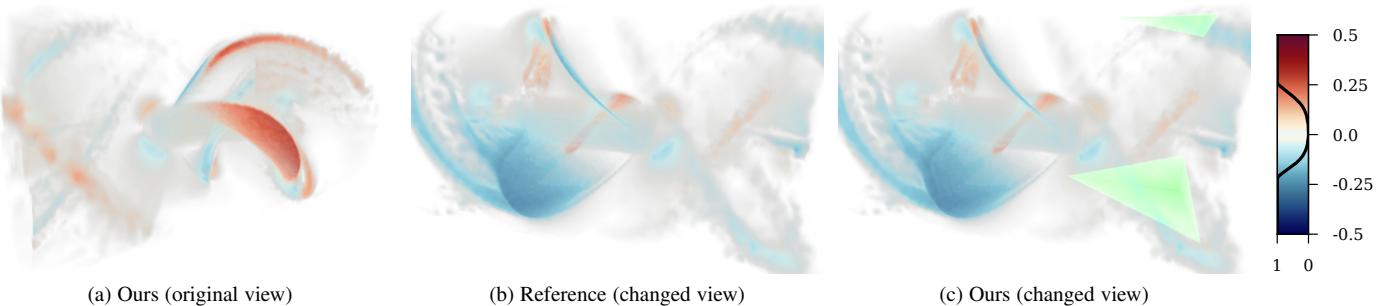


Fig. 13: We create a single moment image of the turbine (a), change the view configuration (b) and use the moment image to render from this view (c). Regions of the volume that are outside the view frustum in (a) cannot be reconstructed and are shown in green in (c).

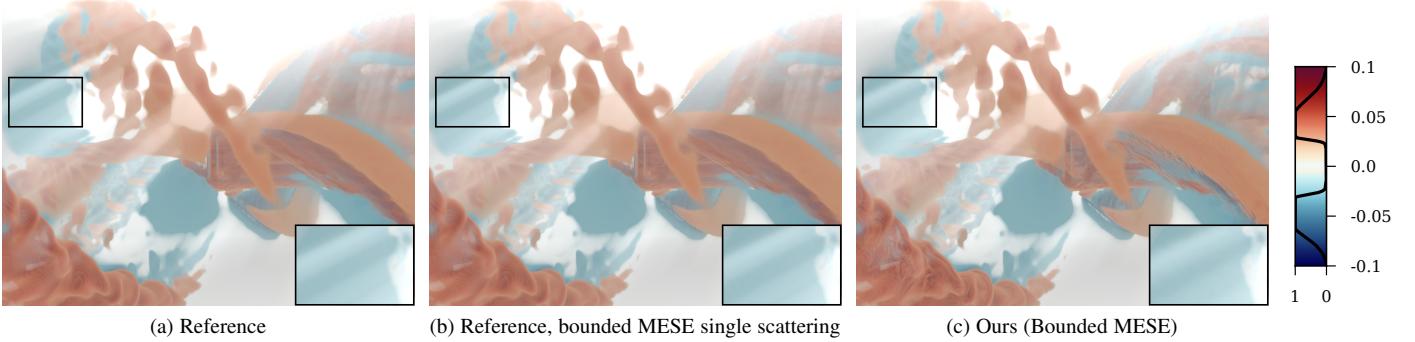


Fig. 14: The turbine dataset with direct volume rendering and single scattering using brute-force ray marching (a), using a moment image to compute single scattering from the directional light source (b), and using moment images for both (c).

contains noise, for example on the upper right side. These artifacts might be due to the depth ordering of samples that is not considered in their approach. They might also stem from quantization of the floating point values due to the use of histograms. Note that our approach does not quantize the sampled densities. Instead, we quantize the moments from which we reconstruct a smooth signal. Lastly, ZFP in (e) shows block artifacts whilst the results of TTHRESH (f) are similar to ours.

In Figure 12, we use the same moment image as for Figure 11, but change the transfer function to show two small intervals. Our bounded MESE (b) and Fourier (c) reconstructions accurately incorporate this transfer function and lead to results that are nearly indistinguishable from the reference. Note that the transfer function can be changed interactively, whilst the reference performs expensive SPH interpolation during rendering. The image-based approaches enable the interactive exploration of this large dataset. In the supplemental video, we employ spatial and temporal interpolation of moment images to visualize the full time-dependent dataset.

We use a single moment image to render Figure 13 (a). Then, we rotate the view to the other side of the volume (b). We reproject our moment image (c), which still leads to accurate results. Some regions of the volume are outside the view frustum of the moment image. These regions are shown in green since we do not have any information in these areas. The employed direct reprojection renders in 1.2 seconds. In the supplemental video, we render each frame in 4 milliseconds after spending one second on resampling to a uniform grid of size $512 \times 256 \times 256$ (Section 4.3).

In Figure 14 we show the turbine dataset with direct volume rendering and single scattering using brute-force ray marching (a), using a moment image for single scattering (b), and using moment images for both single scattering and volume rendering (c). The single scattering moment image is computed in a resolution of 512^2 with a maximum of 50 moments. Using our coding and quantization scheme it is only 5.3 MB in file size, but the single scattering is nearly indistinguishable from the reference.

5.2 Quantitative Evaluation

Table 1 shows the size of moment images after adaptively determining the number of moments and after performing quantization and compression. Reducing the number of moments and especially quantizing the moments lead to the most significant reduction in size. Our compact compression thus makes it possible to store and transfer larger quantities of moment images.

Table 2 compares our approach using the bounded MESE and Fourier reconstruction to ray-histograms, as well as ZFP and TTHRESH. Note that the middle row of the turbine dataset corresponds to the results shown in Figure 11. We compare all of these methods by measuring the compression rate in bits per sample (bps) and distortion of scalar samples along each ray as the peak signal-to-noise ratio (PSNR) in the logarithmic scale (db). Additionally, we use the structural similarity index measure (SSIM) [40] to assess the accuracy of composited color images.

Our approach performs similarly with the bounded MESE and the Fourier reconstruction. The bounded MESE performs best for the Richtmyer-Meshkov dataset, whilst the Fourier reconstruction leads to the best results for the turbine dataset. Note that the image-based representation is identical, regardless of the reconstruction. The ray-histogram approach shows a low and noisy PSNR due to its stochastic reconstruction, but the final image accuracy is better. Still, the distortion as well as the compression rate are noticeably worse compared to our approach.

ZFP and especially TTHRESH lead to a similar rate-distortion at often better compression rates compared to our approach. With the exception of the Richtmyer-Meshkov dataset, the quality of composited images is worse, especially for lower bitrates. Since ZFP and TTHRESH lead to better results when compressing the original volumetric datasets, the perspective transformation seems to be responsible. For example, we noticed block artifacts for images generated after ZFP compression of the perspective grid.

5.3 Performance Analysis

For comparability, we measure the generation and reconstruction steps of our approach on the same system: An AMD Ryzen 5 3600 with 16GB RAM and an NVIDIA GeForce 1080Ti. We accelerate all steps using CUDA, except for the lossless compression. The

TABLE 1: Storage sizes after reducing the number of moments (32-bit floats), quantization ($b_1 = 10$), and lossless compression. Initially, each image contains $1024 \times 768 \times 100$ moments, i.e. 300 MB.

Dataset	Reduced (Sec. 3.5)	Quantized (Sec. 3.6.3)	Compressed (Sec. 3.6.4)
Rayleigh-Taylor	129.02MB	37.23MB	25.84MB
Richtmyer-Meshkov	279.02MB	56.09MB	39.34MB
Turbine	254.24MB	62.31MB	51.89MB

TABLE 2: We compare our method using the bounded MESE and Fourier reconstruction, ray-histograms, and the compression methods ZFP and TTHRESH. For all datasets, we implicitly or explicitly compress a $768 \times 1024 \times 768$ perspective grid. We measure the required bits per sample (bps), distortion (PSNR), and quality of composited color images (SSIM).

	Ours (Bounded MESE)			Ours (Fourier)			Ray-histograms			ZFP			TTHRESH		
	bps	PSNR	SSIM	bps	PSNR	SSIM	bps	PSNR	SSIM	bps	PSNR	SSIM	bps	PSNR	SSIM
Richtmyer-Meshkov	0.43	24.33	0.645	0.43	23.03	0.550	2.85	10.08	0.448	0.29	23.42	0.579	0.01	21.17	0.875
	0.53	25.19	0.683	0.53	23.83	0.585	4.00	10.36	0.503	0.37	25.92	0.720	0.03	24.33	0.908
	0.87	27.41	0.754	0.87	25.82	0.675	5.55	10.54	0.550	0.44	30.91	0.931	0.26	32.10	0.928
Rayleigh-Taylor	0.25	30.31	0.939	0.25	30.83	0.951	1.78	13.91	0.831	0.37	26.99	0.850	0.06	28.06	0.846
	0.38	35.50	0.977	0.38	36.41	0.985	2.05	14.14	0.869	0.44	30.91	0.910	0.32	33.55	0.936
	0.53	38.24	0.986	0.53	39.42	0.992	3.20	13.68	0.937	0.53	34.95	0.956	0.71	36.73	0.963
Turbine	0.33	44.63	0.983	0.33	45.15	0.983	0.44	30.12	0.948	0.38	38.61	0.948	0.34	42.58	0.982
	0.51	50.11	0.995	0.51	51.81	0.996	0.57	29.87	0.959	0.46	48.49	0.984	0.58	48.01	0.995
	0.98	55.43	0.998	0.98	57.21	0.998	1.70	30.00	0.989	0.50	53.16	0.992	0.82	53.66	0.996

TABLE 3: Run-time measurements from our datasets.

Dataset	Reference	Generation						Rendering		
		Ray marching	Determining moments	Coding	Quantization curve	Lossless encoding	Lossless decoding	Moment preparation	Ray marching	
Richtmyer-Meshkov	28.9 ms	1025.6 ms	1722.7 ms	926.5 ms	565.5 ms	235.1 ms	458.7 ms	708.9 ms	136.2 ms	
Rayleigh-Taylor	27.1 ms	244.7 ms	827.3 ms	411.6 ms	226.7 ms	124.7 ms	221.9 ms	318.4 ms	20.6 ms	
Turbine	18,731 ms	6392.6 ms	1810.3 ms	825.3 ms	507.0 ms	312.1 ms	507.5 ms	647.7 ms	49.9 ms	

performance measurements are shown in Table 3. Note that the Fourier reconstruction uses the exact same steps except for the final ray marching. In comparison to the ray-histogram approach, generating an image is significantly faster with our approach since we do not have to repeatedly ray march the data. The encoding steps for ZFP and TTHRESH require significantly more compute and memory resources. For example, TTHRESH encodes the Richtmyer-Meshkov data in between 152s and 180s, whilst ZFP takes between 4s and 22s. This excludes the already costly generation of the perspective grid. In comparison, our approach requires less memory, takes only a few seconds, and is highly parallelizable. In the supplemental document, we perform additional scalability measurements on a CPU.

In comparison to a reference ray marching implementation, the generation of a moment image is several times slower. Since each step is trivial to parallelize, generating the moment images on a compute cluster would be significantly faster or could produce multiple images in parallel. Moreover, ray marching SPH data is extremely costly and not suited for interactive rendering. With our approach, the rendering step is decoupled from data access and the SPH interpolation and is thus interactive.

5.4 Number of Moments

In Figure 15, we change the maximum number of moments for all datasets and measure the accuracy of the reconstruction. Increasing this maximum improves the overall accuracy of moment images, but also leads to higher computational and storage requirements. Additional experiments regarding the maximal number of moments can be found in the supplemental document.

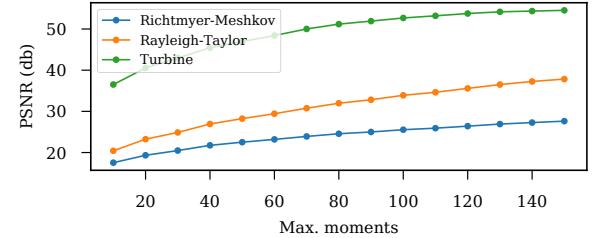


Fig. 15: We compute moment images with a differing maximum number of moments and determine the respective accuracy of reconstructed scalar values.

5.5 Discussion

Our evaluation shows that our proposed moment image representation is both compact and accurate. Selecting the number of moments per pixel, encoding, and quantizing the moments is key to achieve small image sizes. Our technique enables scientists to create a large amount of images from different views or simulation time steps. However, moment images are still an approximate representation, which is necessary to compactly represent large volumetric data. As such, moment images are unsuited for data that have to be represented as accurately as possible, such as medical scans. Furthermore, our evaluation shows that datasets containing discontinuous or high frequent signals, for example the Richtmyer-Meshkov dataset, are more difficult to represent than datasets consisting mostly of continuous and smooth signals.

Compared to ray-histograms [7], our representation is both smaller and more accurate since we do not store distributions, which cannot reconstruct the ordering of densities. Moreover, our evaluation shows that our approach outperforms more general

compression methods applied to image-based volume visualization, regarding both the quality of composited images and performance. Compared to the bounded MESE, Fourier reconstruction yields surprisingly good results, especially for smooth signals such as the SPH dataset. However, ringing can cause strong artifacts for the Fourier reconstruction. In comparison, the bounded MESE yields good and more predictable results. Therefore, we recommend the Fourier reconstruction for known smooth datasets, such as SPH data, and the bounded MESE otherwise. Note that most of the concepts discussed in this paper, including our coding and quantization strategy, are equally applicable to both.

We have shown that moment images can be used for single scattering illumination. Since that requires less accuracy, the corresponding moment images take only a few megabytes. Lastly, we were surprised how changing the view configuration still leads to accurate results. This shows that our representation is indeed a volumetric representation, but compactly encoded in an image.

6 CONCLUSION AND FUTURE WORK

Our image-based representation enables the interactive exploration of large and arbitrarily structured volumetric data by decoupling the access to the data from interactive rendering and exploration. It is suited for in situ applications since it is fast, highly parallelizable, and produces small images. Since our representation is completely volumetric, it allows for some changes in the view configuration and the exploration with arbitrary transfer functions.

In the future, we want to investigate whether subdividing a ray into less complex signals can increase the accuracy or decrease the total number of moments per pixel. Lastly, we want to explore if we can combine multiple moment images to reconstruct a complete volume with high accuracy.

ACKNOWLEDGMENTS

This work has been supported by the bwVisu and bwVisu2 projects funded by the Ministry for Science, Research and Art of the country of Baden-Württemberg, Germany.

The Rayleigh-Taylor instability is due to A. W. Cook, W. Cabot, and P. L. Miller. The simulation of a Richtmyer-Meshkov instability is due to R. H. Cohen, W. P. Dannevick, A. M. Dimits, D. E. Eliason, A. A. Mirin, and Y. Zhou.

REFERENCES

- [1] H. Childs, “Data exploration at the exascale,” *Supercomputing frontiers and innovations*, vol. 2, no. 3, pp. 5–13, 2015.
- [2] S. Li, N. Marsaglia, C. Garth, J. Woodring, J. Clyne, and H. Childs, “Data reduction techniques for simulation, visualization and data analysis,” *Computer Graphics Forum*, vol. 37, no. 6, pp. 422–447, 2018.
- [3] J. Ahrens, S. Jourdain, P. O’Leary, J. Patchett, D. H. Rogers, and M. Petersen, “An image-based approach to extreme scale in situ visualization and analysis,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2014, pp. 424–434.
- [4] N. Shareef, T.-Y. Lee, H.-W. Shen, and K. Mueller, “An image-based modelling approach to GPU-based unstructured grid volume rendering,” in *Volume Graphics*, 2006.
- [5] S. Frey, F. Sadlo, and T. Ertl, “Explorable volumetric depth images from raycasting,” in *XXVI Conference on Graphics, Patterns and Images*, 2013, pp. 123–130.
- [6] O. Fernandes, S. Frey, F. Sadlo, and T. Ertl, “Space-time volumetric depth images for in-situ visualization,” in *IEEE 4th Symposium on Large Data Analysis and Visualization*, 2014, pp. 59–65.
- [7] K. Wang, N. Shareef, and H. Shen, “Image and distribution based volume rendering for large data sets,” in *IEEE Pacific Visualization Symposium*, 2018, pp. 26–35.
- [8] K. C. Wang, T. H. Wei, N. Shareef, and H. W. Shen, “Ray-based exploration of large time-varying volume data using per-ray proxy distributions,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3299–3313, 2020.
- [9] C. Peters, S. Merzbach, J. Hanika, and C. Dachsbaecher, “Using moments to represent bounded signals for spectral rendering,” *ACM Transactions on Graphics*, vol. 38, no. 4, pp. 136:1–136:14, 2019.
- [10] M. Balsa Rodríguez, E. Gobbetti, J. Iglesias Gutián, M. Makhinya, F. Marton, R. Pajarola, and S. Suter, “State-of-the-art in compressed GPU-based direct volume rendering,” *Computer Graphics Forum*, vol. 33, no. 6, pp. 77–100, 2014.
- [11] J. Beyer, M. Hadwiger, and H. Pfister, “State-of-the-art in GPU-based large-scale volume visualization,” *Computer Graphics Forum*, vol. 34, no. 8, pp. 13–37, 2015.
- [12] B.-L. Yeo and B. Liu, “Volume rendering of DCT-based compressed 3D scalar data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 1, pp. 29–43, 1995.
- [13] P. Lindstrom, “Fixed-rate compressed floating-point arrays,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.
- [14] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola, “TTHRESH: Tensor compression for multidimensional visual data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 9, pp. 2891–2903, 2020.
- [15] J. Lukasczyk, E. Kinner, J. Ahrens, H. Leitte, and C. Garth, “Voidga: A view-approximation oriented image database generation approach,” in *IEEE 8th Symposium on Large Data Analysis and Visualization*, 2018, pp. 12–22.
- [16] R. Bujack, D. H. Rogers, and J. Ahrens, “Reducing occlusion in cinema databases through feature-centric visualizations,” in *Leipzig Symposium on Visualization In Applications (LEVIA)*, 2018.
- [17] J. Lukasczyk, C. Garth, M. Larsen, W. Engelke, I. Hotz, D. Rogers, J. Ahrens, and R. Maciejewski, “Cinema darkroom: A deferred rendering framework for large-scale datasets,” in *IEEE 10th Symposium on Large Data Analysis and Visualization*, 2020, pp. 37–41.
- [18] A. Tikhonova, C. D. Correa, and K.-L. Ma, “Explorable images for visualizing volume data,” in *IEEE Pacific Visualization Symposium*, 2010, pp. 177–184.
- [19] A. Tikhonova, C. D. Correa, and K.-L. Ma, “An exploratory technique for coherent visualization of time-varying volume data,” *Computer Graphics Forum*, vol. 29, no. 3, pp. 783–792, 2010.
- [20] A. Tikhonova, C. D. Correa, and K.-L. Ma, “Visualization by proxy: A novel framework for deferred interaction with volume data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1551–1559, 2010.
- [21] K. Mueller, N. Shareef, J. Huang, and R. Crawfis, “IBR-assisted volume rendering,” in *Late Breaking Hot Topics, Proceedings of IEEE Visualization*, 1999, pp. 5–8.
- [22] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, “Layered depth images,” in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1998, pp. 231–242.
- [23] C. Peters and R. Klein, “Moment shadow mapping,” in *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*. ACM, 2015, pp. 7–14.
- [24] C. Peters, C. Münstermann, N. Wetzstein, and R. Klein, “Improved moment shadow maps for translucent occluders, soft shadows and single scattering,” *Journal of Computer Graphics Techniques (JCGT)*, vol. 6, no. 1, 2017.
- [25] C. Peters, J. Klein, M. B. Hullin, and R. Klein, “Solving trigonometric moment problems for fast transient imaging,” *ACM Transactions on Graphics*, vol. 34, no. 6, 2015.
- [26] C. Münstermann, S. Krumpen, R. Klein, and C. Peters, “Moment-based order-independent transparency,” *Proceedings of the ACM Computer Graphics and Interaction Techniques*, vol. 1, no. 1, 2018.
- [27] M. Kern, C. Neuhauser, T. Maack, M. Han, W. Usher, and R. Westermann, “A comparison of rendering techniques for 3D line sets with transparency,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2020.
- [28] I. Baeza Rojo, M. Gross, and T. Günther, “Fourier opacity optimization for scalable exploration,” *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019.
- [29] M. Zeidan, T. Rapp, C. Peters, and C. Dachsbaecher, “Moment-based opacity optimization,” in *Eurographics Symposium on Parallel Graphics and Visualization*, 2020.
- [30] K. Engel, M. Kraus, and T. Ertl, “High-quality pre-integrated volume rendering using hardware-accelerated pixel shading,” in *Proceedings of the*

- ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware.*
ACM, 2001, p. 9–16.
- [31] C. Peters, S. Merzbach, J. Hanika, and C. Dachsbacher, “Spectral rendering with the bounded MESE and sRGB data,” in *Workshop on Material Appearance Modeling*. The Eurographics Association, 2019.
 - [32] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Communications of the ACM*, vol. 30, no. 6, pp. 520–540, 1987.
 - [33] C. Lundström, P. Ljung, A. Persson, and A. Ynnerman, “Uncertainty visualization in medical volume rendering using probabilistic animation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1648–1655, 2007.
 - [34] S. Liu, J. A. Levine, P. Bremer, and V. Pascucci, “Gaussian mixture model based volume visualization,” in *IEEE Symposium on Large Data Analysis and Visualization*, 2012, pp. 73–77.
 - [35] E. Sakhaei and A. Entezari, “A statistical direct volume rendering framework for visualization of uncertain data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 12, pp. 2509–2520, 2017.
 - [36] R. Jain and I. Chlamtac, “The p^2 algorithm for dynamic calculation of quantiles and histograms without storing observations,” *Communications of the ACM*, vol. 28, no. 10, pp. 1076–1085, 1985.
 - [37] F. Lindemann and T. Ropinski, “About the influence of illumination models on image comprehension in direct volume rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1922–1931, 2011.
 - [38] N. Max, “Optical models for direct volume rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.
 - [39] L. Williams, “Casting curved shadows on curved surfaces,” *SIGGRAPH Computer Graphics*, vol. 12, no. 3, pp. 270–274, 1978.
 - [40] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.



Tobias Rapp received the PhD (Dr.-Ing.) degree in computer science from the Karlsruhe Institute of Technology, Germany. His research in the field of visualization focuses on the visualization of large scattered data and dynamical systems.



Christoph Peters received the PhD (Dr. rer. nat.) degree in computer science from University of Bonn, Germany. He has been a postdoctoral researcher at the Karlsruhe Institute of Technology, Germany, for four years and is now at Intel. His research focuses real-time rendering with an emphasis on mathematical methods.



Carsten Dachsbaucher received the PhD (Dr.-Ing.) degree in computer science from the University of Erlangen-Nuremberg, Germany. He is a professor of computer science at the Karlsruhe Institute of Technology, Germany, and heads the Institute for Visualization and Data Analysis. His research interests include computer graphics, light transport simulation, interactive and perceptual rendering, and scientific visualization.