Volume Rendering with Multidimensional Peak Finding

Natallia Kotava*
University of Kaiserslautern, Germany

Aaron Knoll
Argonne National Laboratory, USA

Mathias Schott
University of Utah, USA

Christoph Garth
University of Kaiserslautern, Germany

Xavier Tricoche
Purdue University, USA

Christoph Kessler
University of Kaiserslautern, Germany

Elaine Cohen
University of Utah, USA

Charles D. Hansen
University of Utah, USA

Michael E. Papka
Argonne National Laboratory, USA

Hans Hagen
University of Kaiserslautern, Germany

ABSTRACT

Peak finding provides more accurate classification for direct volume rendering by sampling directly at local maxima in a transfer function, allowing for better reproduction of high-frequency features. However, the 1D peak finding technique does not extend to higher-dimensional classification. In this work, we develop a new method for peak finding with multidimensional transfer functions, which looks for peaks along the image of the ray. We use piecewise approximations to dynamically sample in transfer function space between world-space samples. As with unidimensional peak finding, this approach is useful for specifying transfer functions with greater precision, and for accurately rendering noisy volume data at lower sampling rates. Multidimensional peak finding produces comparable image quality with order-of-magnitude better performance, and can reproduce features omitted entirely by standard classification. With no precomputation or storage requirements, it is an attractive alternative to preintegration for multidimensional transfer functions.

Keywords: volume rendering, multidimensional transfer functions, peak finding

1 INTRODUCTION

Direct volume rendering (DVR) is a popular technique for visualizing spatial scientific data. A strong appeal of this method is its flexibility: through choice of transfer function, a user can dynamically classify and render different phenomena in the same data set. Multidimensional transfer functions allow the user to visualize volume data more expressively, through classification of derived values such as gradient or curvature [10]. For multifield volume data, multidimensional transfer functions offer insight into relations between variables and provide comparative analysis. Despite being more complex to design, multidimensional transfer functions are more powerful than 1D transfer functions and offer more control than automatic classifications.

While transfer functions offer flexibility, accurate volume rendering requires adequate sampling with respect to both the volume data and chosen transfer function. When either the volume data or transfer function possess high frequencies, high sampling rates are required to reproduce features without artifacts. The conventional solution has been to choose sufficiently smooth transfer functions; however, this limits classification. In multifield visualization, in particular, users are interested in analyzing relationships between two variables, such as regression lines. As shown in Figure 1, rendering narrow features is costly and often inaccurate with uniform spatial sampling.

By integrating the volume and transfer function domains separately, preintegration [6] reproduces high-frequency features that are omitted by standard classification. Peak finding employs a similar tactic, treating high-frequency values as discrete isovalues and solving for their location along the ray for shading [12]. Classifications with Dirac impulses or sharp features are effectively rendered as isosurfaces and shaded accordingly. More significantly, noisy volume data can be classified and rendered accurately with a lower sampling rate and better interactivity.

This paper extends peak finding to multidimensional transfer functions, and presents several novel approaches for finding and sampling at peaks. In general, we propose sampling dynamically in transfer function space, which is inexpensive compared to sampling in volume space, and allows for accurate integration across both domains. We investigate several mechanisms for doing this, using ray marching or scanline sampling on chord or spline parameterizations of the ray's image in transfer function space. These approaches require no precomputation, and extend trivially to N-dimensional transfer functions. Though simple, multidimensional peak finding has not been examined before, and offers clear qualitative and performance benefits over conventional volume rendering with multidimensional transfer functions.

We call our new technique multidimensional peak finding, because it pursues similar goals as the one-dimensional peak finding method, namely locating maxima in-between the samples in the transfer function space. However, in an N-dimensional setting we do not try to locate peaks of the transfer function, but find the local maxima along the approximated image of the ray. These points usually do not coincide with the maxima of the N-dimensional function. Kraus touches on this topic by dealing with peaks in a conventional sense [15].

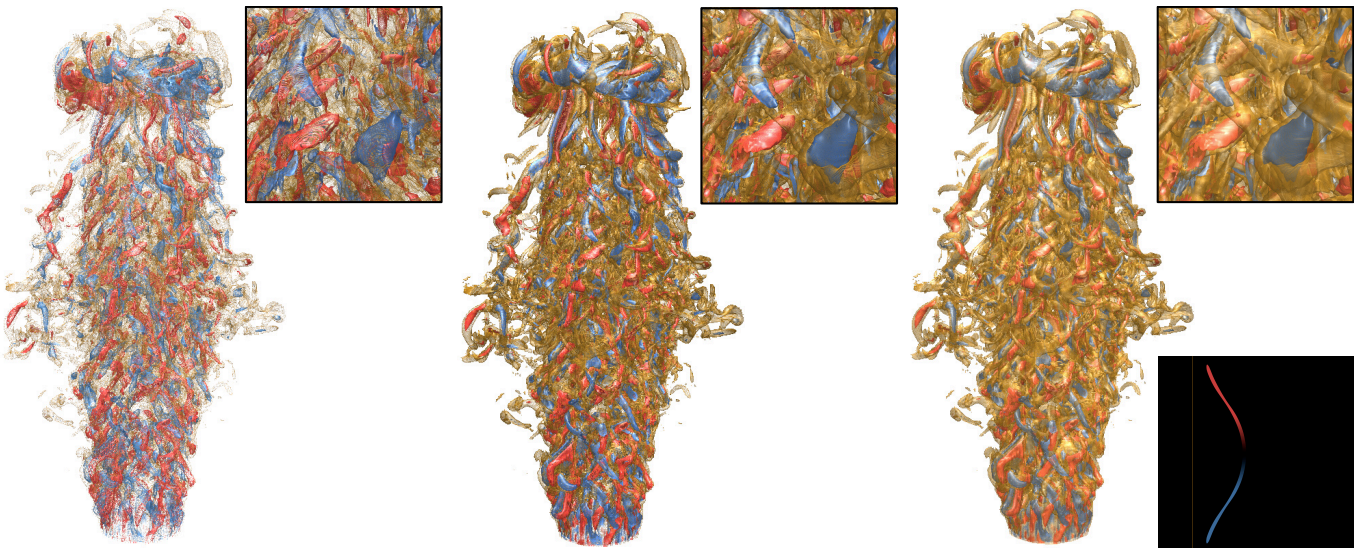
2 RELATED WORK

Direct volume rendering was introduced by the ray caster of Levoy [18]. Fast rasterization hardware made interactive DVR possible with slicing [4, 5]. While splatting [29] is feasible, ray casting [16, 24] has regained popularity due to its efficient implementation on current GPU hardware. Isosurface mesh extraction from structured volume data was first proposed by [20] and remains a common method for visualization. Direct ray casting of isosurfaces was proven on the CPU [23, 27] and later on the GPU [7]. Kraus [14] reformulated direct volume rendering as an integration of isosurfaces, showing that irradiance can be computed without normalizing the Riemann sum over the number of samples. Multifield isosurface rendering has been used to visualize particle astrophysics data. Navratil et al. [21] use marching cubes to extract separate meshes, while Linsen et al. [19] employ a particle reconstruction method to resample and render a single surface from multiple channels.

Laidlaw [17] first advocated multidimensional transfer functions for improved classification of MRI data. 2D transfer functions with gradient magnitude [8,10] or curvature [9] can greatly improve classification flexibility, particularly for noisy scan data in biology and medicine. Kniss et al. [11] applied specially constructed Gaussian kernels to multifield volume visualization using an analytical integration method for improved visual quality. Simpler classifica-

*e-mail: kotava@rhrk.uni-kl.de

IEEE Pacific Visualization Symposium 2012
28 February - 2 March, Songdo, Korea
978-1-4673-0866-3/12/$31.00 ©2012 IEEE



(a) postclassification $\Delta t = 2$, 5fps (b) postclassification $\Delta t = 0.5$, 0.36fps (c) peak finding (ray march, chord) $\Delta t = 2$, 4.3fps

Figure 1: Volume rendering of a 2-channel fluid dynamics data set consisting of vorticity magnitude and normalized helicity. The transfer function (shown in the lower-right corner) is chosen to visualize surfaces of medium vorticity (yellow) and high-vorticity regions (red and blue). In the latter, normalized helicity is considered as a secondary variable to color strong vortical regions by direction of rotation. Multidimensional peak finding lets us quickly and accurately render multi-criteria vortex features without explicit mesh extraction.

tion and blending operations, such as maximum-intensity projection, can equally be used for rendering multifield data [26].

Preintegration [6, 25] integrates transfer function space using a separate Riemann sum. Irradiance on a ray segment can then be queried in a 2D lookup table. Multidimensional transfer functions can be preintegrated and rendered using a summed area table [15]. This is more costly to render, requiring frustum tracing to integrate over a 2D beam footprint in transfer function space. It is also expensive to preintegrate high-resolution 2D transfer functions, and this approach has not been extended to higher-dimensional classification. Peak finding [12] combines direct volume rendering with discrete isosurfacing by sampling directly at peaks in the transfer function. Ament et al. [1] detail a more robust method for DVR integration of discrete isosurfaces that removes scale-dependency entirely. However, it is expensive (requiring 3 samples per voxel as opposed to multiple voxels between samples for peak finding) and would not extend easily to multidimensional classification due to its reliance on lookup tables.

3 Background

Direct volume rendering is a numerical integration of discrete samples blended according to an emission-absorption model approximating the radiative transport equation [18]. On a ray segment, irradiance is represented continuously as:

$$I(a,b) = \int_a^b \rho_E(\mathbf{f}(s))\rho_\alpha(\mathbf{f}(s))e^{-\int_a^s \rho_\alpha(f(t))dt}ds \tag{1}$$

Here, ρ_E is the emissive term or color, ρ_α is the opacity of the transfer function; a, b are the segment endpoints, and $\mathbf{f}(t) = \mathbf{f}(\vec{R}(t))$ is the scalar field function evaluated in world space at t along the ray $\vec{R}$. Since the transfer function is applied after interpolation, $\rho(\mathbf{f})$ implies postclassification. To approximate Equation 1 discretely, we employ a Riemann sum,

$$e^{-\int_a^s \rho_\alpha(\mathbf{f}(t))dt} = \prod_{i=0}^{n} e^{-\Delta t\ \rho_\alpha(\mathbf{f}(i\ \Delta t))} = \prod_{i=0}^{n}(1-\alpha_i), \tag{2}$$

where Δt is the uniform sampling step, $n = (s-a)/\Delta t$, and

$$\alpha_i \approx 1 - e^{-\Delta t\ \rho_\alpha(f(i\ \Delta t))} \tag{3}$$

Discretizing the integral on $[a,b]$ yields the discrete summation

$$I \approx \sum_{i=0}^{N} \check{\rho}_E(i) \prod_{j=0}^{i-1}(1-\alpha_j), \tag{4}$$

where $\check{\rho}_E$ is approximated at discrete points along the ray as:

$$\check{\rho}_E(i) \approx \rho_\alpha(\mathbf{f}(i\ \Delta t))\rho_E(\mathbf{f}(i\ \Delta t)) \tag{5}$$

Preintegration employs a separate integral in transfer function space to estimate $\check{\rho}_E$ and ρ_a [6], specifically the Riemann sum of irradiance between two samples $\mathbf{f}_0 = \mathbf{f}(t_0)$ and $\mathbf{f}_1 = \mathbf{f}(t_1)$, assuming linear spacing of $\mathbf{f}$ values between these points. Typically, the colors $\check{\rho}_E(i)$ are associated, i.e. integrated alongside α_i.

$$\alpha_i \approx 1 - e^{-\int_0^1 \rho_\alpha((1-\omega)\mathbf{f}_0+\omega\mathbf{f}_1)d\ \mathrm{d}\omega}, \tag{6}$$

where $d = ||(\mathbf{f}_1 - \mathbf{f}_0)||$ is the length of a segment.

In order for this linear approximation to be accurate, preintegration assumes that transfer function space is continuous with bounded variation (specifically, Lipschitz) along the ray. However, it is often applied in scenarios where this is not the case. Peak finding [12] assumes the transfer function is potentially discontinuous, and that at sharp peaks α_i it is better approximated by the supremum:

$$\alpha_i \approx 1 - e^{\left(\ \sup_{\mathbf{f}\in(\mathbf{f}_0,\mathbf{f}_1)}\ \rho_\alpha(\mathbf{f})\ \right)} \tag{7}$$

Peak finding assumes the isosurface at that peak is always sampled with constant opacity regardless of the step size Δt. This approximation is only employed where peaks exist; all other samples are assigned α_i according to Equation 6 and integrated using standard postclassification. This has a biasing effect on the integral, but ensures peak features are sampled regardless of sampling rate. To determine if a peak exists, 1D peak finding uses a 2D lookup similar to a preintegrated table, storing the peak isovalue (or isovalues) between $[\mathbf{f}_0, \mathbf{f}_1]$. At rendering, if a peak υ exists in this

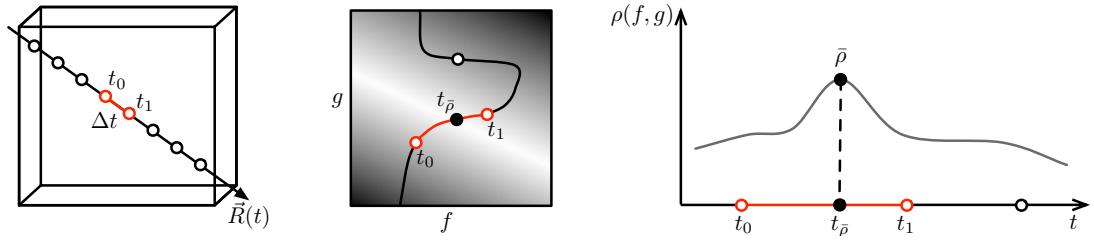


Figure 3: Classification of multivariate data, and locating peaks between segments. (a) - a straight ray $\vec{R}(t)$ in world space; (b) - the curved path of $\vec{R}(t)$ in transfer function domain, $\mathbf{f}(t) = (f(\vec{R}(t)), g(\vec{R}(t)))$; (c) - transfer function profile along the ray, $\rho \circ \mathbf{f}(t)$.

table, one solves for the spatial location t of the isosurface such that $\mathbf{f}(\vec{R}(t)) = \upsilon$, using the secant method.

Peak finding and preintegration accomplish similar aims. Preintegration behaves nicely when both the volume and transfer function are Lipschitz and sampled adequately. Peak finding is preferable when one desires to sample the volume at a lower rate than the transfer function, in the case of extremely high-frequency transfer functions or noisy data.

4 Multidimensional Peak Finding

A multidimensional opacity transfer function is given by a $\rho : \mathbb{R}^M \rightarrow \mathbb{R}$, which classifies a multivariate scalar field $\mathbf{f} \in \mathbb{R}^M \times \mathbb{R}^N = \{f^0, f^1, , ..., f^M\}, f^i \in \mathbb{R}^N$. In theory we assume $\mathbf{f}$ is Lipschitz; but in practice the transfer function can be piecewise-constant (non C^0) with sufficiently high discretization.

Integrating the data and transfer function separably is more necessary for multidimensional classification than for the univariate case. Bergner et al. [2] have shown that for 1D classification the proper sampling rate of the composite function $\rho \circ \mathbf{f}$ depends on the frequency $\nu = \nu(\rho)\max|\mathbf{f}'|$. For multivariate data, the function signal contains all component variables of $\mathbf{f}$ and the band-limiting frequency is given by $\nu = \max_{\|\mathbf{l}\|=1}(\nu_{\mathbf{l}}\max(\mathbf{l}\cdot\mathbf{f}'))$, for $\mathbf{l}$ being all possible directions of the multidimensional composite function. Thus, multivariate postclassification is inherently higher-frequency, as illustrated in Figure 3. However, if we sample ρ and $\mathbf{f}$ separably, we need only sample at the maximum of those data frequencies $\sup\{\nu(\mathbf{f}')\}$ in world space, and separately at $\nu(\rho)$ in transfer space.

Extending 1D peak finding to multidimensional transfer functions is nontrivial, because peaks along a world-space ray almost never occur at maxima in higher-dimensional transfer space, as shown in Figure 2. In most cases, therefore, one cannot precompute a peak table and solve for an isovalue at each peak along a ray segment. When handling multidimensional transfer functions, different strategies must be used.

4.1 Separable Transfer Functions

When a multidimensional transfer function is separable into 1D transfer functions, one can use 1D peak finding with some modifications. Even when peaks are close together, it is most efficient to assume one peak per ray segment and employ one isosurface solving routine (as opposed to separate solvers for separate dimensions). Instead of using the secant method, we use bisection to determine whether a root for any dimension exists in the left or right half of a bracket. While this can yield shading ambiguities when roots exist in both subfields of $\mathbf{f}$, in practice the quality is good and performance is as fast as 1D peak finding. Unfortunately, this trivial extension only works for a small subset of multidimensional classifications.

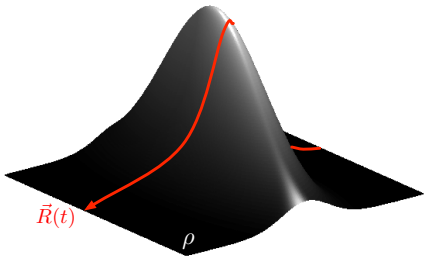


Figure 2: In 2D transfer function space, peaks along the image of the ray are in general not the same as peak points of ρ.

4.2 General Multidimensional Transfer Functions

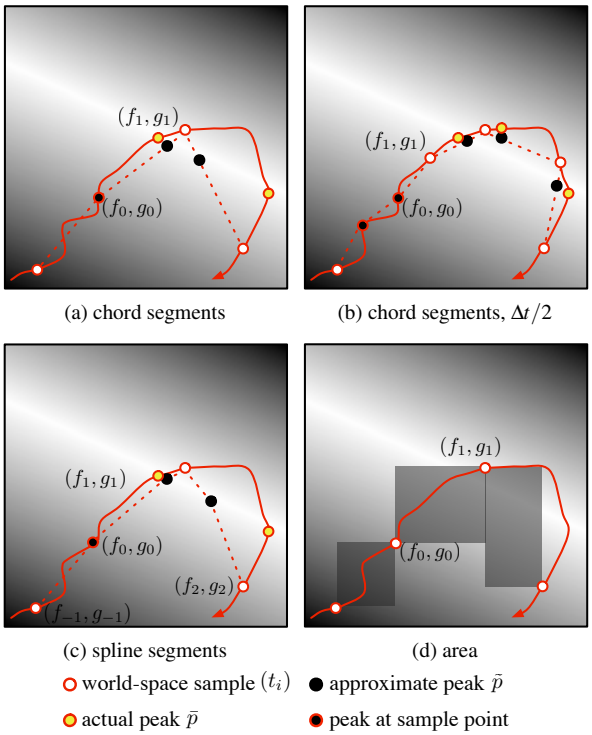
Peak finding in general multidimensional functions requires a different approach. Both enumerating peaks and solving at isosurfaces are difficult (if not intractable) for M-dimensional transfer functions, $M > 1$. Rather than solve for a specific peak value on a segment $[\mathbf{f}_0, \mathbf{f}_1]$ (Figure 3), we note that local maxima can be found dynamically along the ray with sufficient sampling, and that:

- Sampling in transfer space ρ is less expensive than sampling in world space $\mathbf{f}$.
- Since ρ and $\mathbf{f}$ are compact, a contraction in $\mathbf{f}$ yields a contraction in $\rho(\mathbf{f})$.
- Finding the correct local maximum $\bar{\rho}$ on a given $[\mathbf{f}_0, \mathbf{f}_1]$ is more important than accurate world-space location $t_{\bar{\rho}}$.

We are chiefly interested in the intersection of the image of the ray with peak manifolds in transfer function space. Sampling directly along the ray image is costly, as it requires an increase in the world-space sampling rate. However, it is inexpensive to sample along an *approximation* of that image in transfer space itself. Such an approximation can be parameterized from world-space points $\mathbf{f}_i$, and sampled directly in transfer space ρ.

As with 1D peak finding, though we must still sample at the Nyquist limit in ρ-space, we only need to sample $\mathbf{f}$ such that ρ is monotonic on each $[\mathbf{f}_0, \mathbf{f}_1]$ – a less stringent requirement. We thus propose to parameterize an approximation of a segment $[\mathbf{f}_0, \mathbf{f}_1]$ in transfer space, as illustrated in Figure 4. As $\Delta t = [t_0, t_1]$ contracts, the segment connecting $\mathbf{f}_0, \mathbf{f}_1$ contracts to approximate $\rho(\mathbf{f})$, as shown in Figure 4 (a) and (b). We can directly query the transfer function along these segments, identifying a local maximum and using that as our peak. In the rest of this section, we explore chord and spline parameterizations (Figure 4 (a-c), using ray marching or scanline sampling. We claim that curve approximations capture classified features better than area elements (Figure 4(d), [15]), and are less costly when dynamically computing maxima or integrals.

When sampling in transfer space, time complexity is linear per ray segment in the worst case, as opposed to constant for postclassification, 1D peak finding, and both 1D and multidimensional preintegration. Unlike with 1D peak finding, we must look for a peak on each segment, since it is not known in advance if it exists in $[\mathbf{f}_0, \mathbf{f}_1]$ or not. However, due to the contractive behavior of $\rho(\mathbf{f})$, few samples in ρ are necessary when samples in $\mathbf{f}$ are close. Assuming ρ is



(a) chord segments (b) chord segments, $\Delta t/2$

(c) spline segments (d) area

world-space sample (t_i) approximate peak $\bar{p}$

actual peak $\bar{p}$ peak at sample point

Figure 4: Parameterizations for sampling in transfer space $\rho(f,g)$.

Lipschitz, the number of samples needed on $\rho(\mathbf{f})$ will be bounded, implying average-case constant time complexity per segment. In practice, sampling monotonic regions of ρ incurs small cost. More samples in world-space necessitate fewer samples in transfer space, and visa-versa; we are interested in finding a good equilibrium.

4.3 Chord Parameterization

The simplest method of searching in transfer function space is to parameterize the segment between $[\mathbf{f}_0, \mathbf{f}_1] \in \rho \subset \mathbb{R}^M$ as a line, and sample along that chord. Like 1D peak finding and preintegration, this requires fetching front samples and storing back samples $\mathbf{f}(t_1)$ and $\mathbf{f}(t_0)$, respectively. We compute a constant to normalize samples over this segment:

$$L_f = ||(\mathbf{f}_1 - \mathbf{f}_0)||$$
$$d_s = \Delta s \, W / L_f, \tag{8}$$

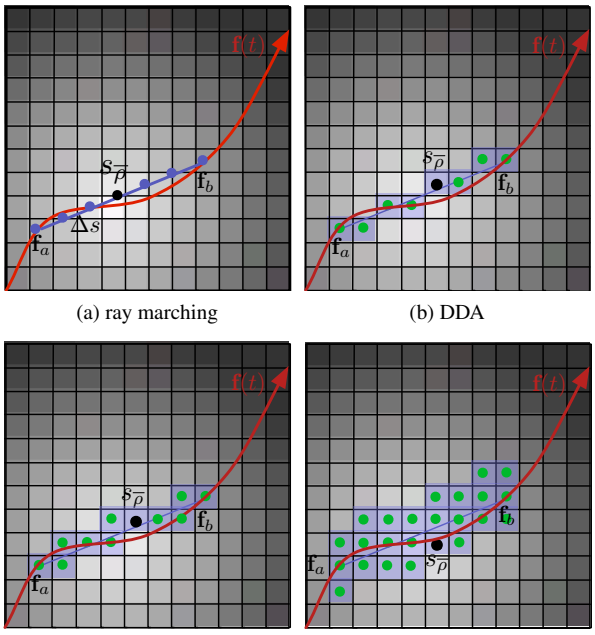
where Δs is our sampling step in ρ (pixels per sample in transfer space) and W is the discretization of the transfer function (W=1024 for a 1k^2 texture). We then parameterize the chord as a ray $\mathbf{F}(s)$, where $s \in [0,1]$,

$$\mathbf{d}_f = (\mathbf{f}_1 - \mathbf{f}_0) \tag{9}$$
$$\mathbf{F}(s) = \mathbf{f}_0 + s \, d_s \mathbf{d}_f$$

We find that better visuals and performance are achieved with relatively high-resolution transfer functions with smooth (non-pixelated) features and $\Delta s > 1$; we use $\Delta s = 4, W = 1024$ in Figure 1. For analytical transfer functions such as the one shown in Figure 5, we similarly set Δs based on the smallest desired feature size. Through this iteration, we find the $s_{\bar{\rho}}$ corresponding to the maximum $\bar{\rho}(\mathbf{F}(s))$ along the segment; then we interpolate to find the peak t:

$$t_{\bar{\rho}} = t_0 + \frac{s_{\bar{\rho}}}{t_1 - t_0} \tag{10}$$

Having found the peak on this segment, we proceed to shade (Section 4.6).



(a) ray marching (b) DDA

(c) Wu (d) slab

Figure 6: Scanline sampling approaches.

4.4 Spline Parameterization

Spline interpolation is a logical improvement over piecewise linear parameterization. To accomplish this, we must maintain a stencil of four world-space samples $\mathbf{f}_{-1}, \mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2$. We use a cubic Hermite spline formulation, as the four coefficients $h_{ij}(s)$ can be precomputed and efficiently accessed in a 1D texture on the GPU. As in Section 4.3, we use the chord length L_f to parameterize the segment with $s \in [0,1]$ and choose a suitable increment d_s. Although this is an imperfect metric, arc-length parameterization would be too costly. We then parameterize the curve as a Catmull-Rom spline:

$$\mathbf{F}(s) = \mathbf{f}_0 h_{00}(s) + (\mathbf{f}_1 - \mathbf{f}_{-1}) h_{10}(s)$$
$$+ \mathbf{f}_1 h_{01}(s) + (\mathbf{f}_2 - \mathbf{f}_0) h_{11}(s)$$

Interpolating splines should improve the adherence of our approximating segments to the image $\rho(\mathbf{f}(t))$, providing smoother results with fewer world-space samples. However, the added cost of maintaining a 4-point stencil and evaluating the spline makes this approach impractical for most 2D transfer functions, compared to simpler chordal parameterization with more samples. The technique begins to be useful when the data itself is extremely noisy and world-space samples are farther apart, such as in the example depicted in Figure 7.

4.5 Scanline Sampling

When $\mathbf{f}$ is quantized to 8-bit or lower precision and the user requires more precise classification, it is useful to employ low-resolution (256^2) piecewise-constant transfer functions. To cheaply and accurately find peaks in such functions, we employ a scanline algorithm instead of ray parameterization. We use a digital differential analyzer (DDA), namely Bresenham's scanline algorithm [3], to scan the chord from $\mathbf{f}_0$ to $\mathbf{f}_1$ in discretized $\mathbb{R}^M$-space. This better guarantees that features in ρ will not be missed. 2D DDA is similar to ray marching (Section 4.3), except we parameterize the distance between pixel centers, and march along either the X or Y axis, whichever is greater, incrementing the differential and terminating when we reach the endpoint on that axis. Instead of the position along the chord, we use the position along the major axis to determine $s_{\bar{\rho}}$ and again interpolate to find $s_{\bar{\rho}}$. This is illustrated in Figure 6(b).

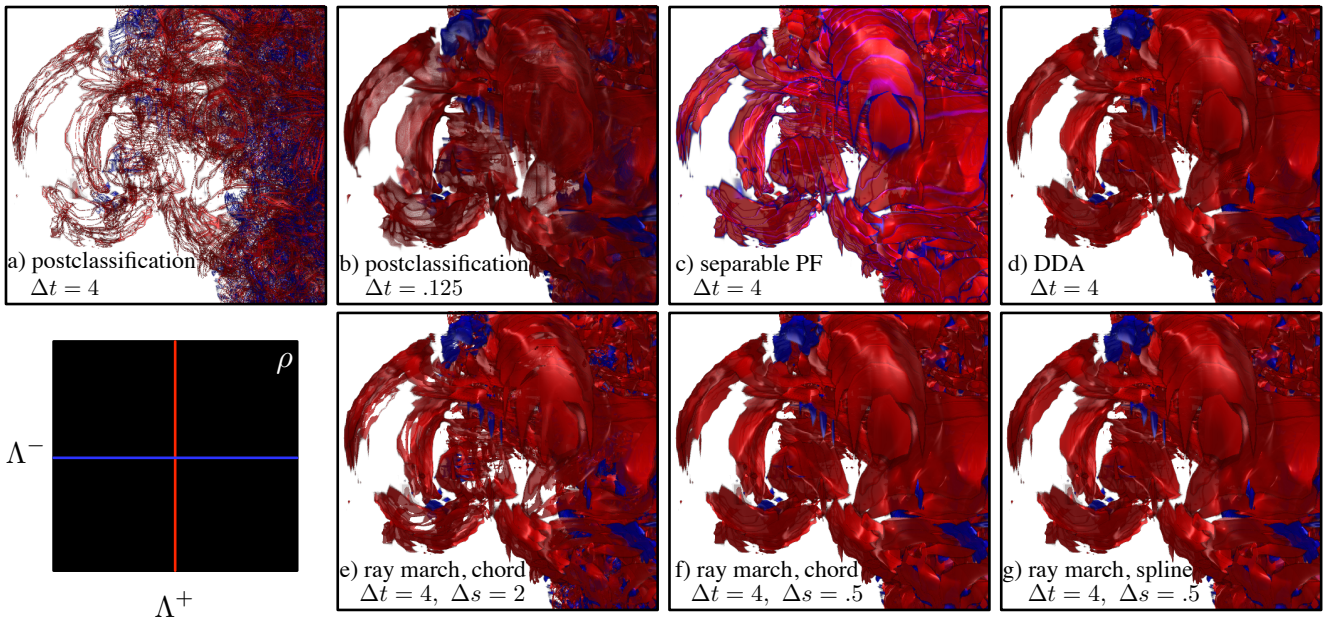


Figure 5: Comparison of various classification techniques on a close view forward (Λ^+) and backward (Λ^-) FTLE fields of a combustion dataset, classified using a sharp separable Gaussian 2D transfer function, evaluated analytically (sampled at a discretized resolution of 1024^2 in (d)). Rendering of frames (a-g) run at 33, 1.2, 35, 22, 32, 18 and 10 fps, respectively.

We implemented other scanline methods such as Wu's algorithm [30] (Figure 6 (c)), which guarantees every pixel between endpoints in ρ will be scanned. However, this approach was slower and did not provide better quality. This is likely because sampling along a chord itself is only an approximation. We also modified the DDA algorithm to rasterize slabs (Figure 6 (d)), which produces fewer artifacts but was more expensive, and can overestimate the number of peaks present. Overall, we find point sampling on chords is best for most cases. DDA can be useful for lower-resolution 2D transfer functions, or when the user does not wish to control Δs.

4.6 Integration and Shading

Since we are finding peaks in between every world-space sample, we do not need to choose a strategy for peak samples with samples from standard (postclassified) DVR integration, as done in 1D peak finding [12]. We use our $t_{\overline{\rho}}$ as the root of the isosurface along the world-space ray, and use $\vec{R}(t_{\overline{\rho}})$ as the position at which to shade.

Two principal options exist for choosing the gradient of a multifield volume $\mathbf{f}$ when shading:

- Shading multiple data gradients $\nabla f^0, \nabla f^1$, etc. separately, using multiple central-differences neighbor stencils.
- Computing the gradient $\nabla\rho(\mathbf{f})$ of the transfer function, classifying $\mathbf{f}$ at each point of a single central-difference stencil.

Both approaches are expensive, and are responsible for a significant share of the cost of multifield DVR regardless of whether peak finding is used or not. In our examples we opt for the first approach because $\nabla\mathbf{f}$ tends to exhibit higher frequency than separate individual gradients.

4.7 Implementation

All approaches presented were implemented in a GPU shader ray caster written in OpenGL and GLSL. To evaluate baseline performance, this renderer is not heavily optimized; it does not employ methods for multiresolution, empty space culling or adaptive sampling. Indeed, applying such techniques to multifield DVR is nontrivial. Simple methods such as precomputing gradients or adaptive sampling could improve performance; however we have opted for the simplicity, flexibility and reproducibility. We note that performance could be greatly improved with such optimizations.

5 Results

Benchmarks were conducted on a 3.0 GHz Intel Xeon and an NVIDIA 285 GTX GPU, at 512^2 screen resolution unless otherwise stated. Overall performance can seem slow, as more samples are required to render high-frequency data and transfer functions than low frequency ones. Generally, peak finding is modestly slower (usually 10-30%) than postclassified ray casting with the same number of world-space samples, but produces results equivalent to sampling at 4x-16x higher rates. In effect, peak finding is an order of magnitude faster for equivalent quality.

Generally, we recommend using 1024^2 2D textures and the chordal ray marching method with $\Delta t = 2, \Delta s = 2$. There is no major performance difference between peak finding with small and large 2D transfer functions. Aliasing in the transfer function domain is a major source of rendering artifacts; even at 256^2 it is easy to specify features in transfer function space that yield artifacts when undersampled in world space. We believe it is better to use peak finding to improve classification quality and performance, rather than to identify peaks at given pixels in transfer space. However, multidimensional peak finding with scanline sampling makes this approach feasible if it is desired. For 256^2 and 512^2 2D transfer functions such as the ones we used in our gradient magnitude classification examples (Section 5.2) we find the DDA method is slightly faster and better at finding peaks than ray marching. Not needing to control Δs can be seen as an advantage. For analytically constructed transfer functions, it is usually more efficient to use ray marching than sampling into a texture and applying DDA.

5.1 Quality Comparison

In Figure 5, we compare results of postclassification, separable peak finding, ray marching and DDA with chordal parameterization, and ray marching with interpolating splines. To compare results we use a simple Gaussian analytical function $\rho_1(x) = e^{-2^{14}(x-.5)^2}$, yielding a separable 2D function $\rho(u,v) = \sup\{\rho_1(u), \rho_1(v)\}$. We see that postclassification requires high sampling rates to produce comparable renderings of these features. Separable 1D peak finding is fast, but shows ambiguities where peaks exist in both fields. The true 2D approaches show more correct results. We see that even for these sharp features, relatively coarse $\Delta t = 4$ and $\Delta s = 2$ generate good facsimilies. We observed

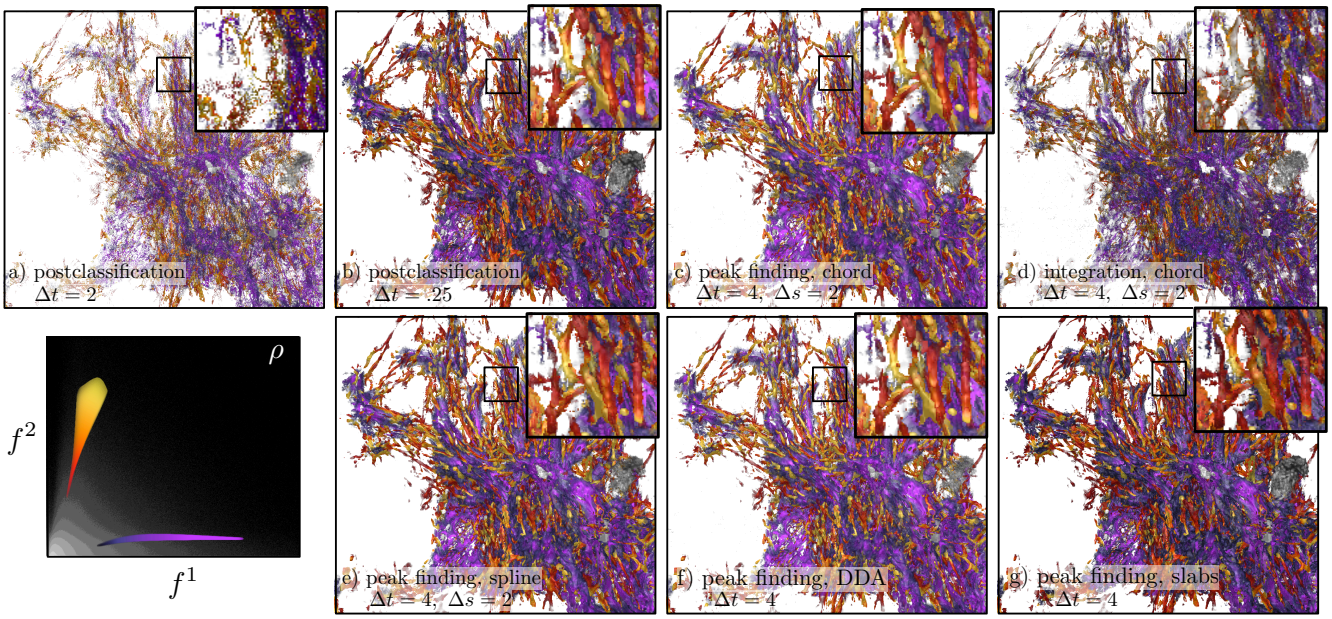


Figure 7: Classification of matter density (f^1) and dark matter density (f^2) in an Enzo computational astrophysics dataset [22]. The transfer function highlights ridges in the joint histogram to illustrate where one matter quantity is high relative to the other. We use a 1024^2 transfer function and render at 1024×768. Frames (a-g) run at 8.5, 1.2, 12.8, 10.6, 7.6, 12.5 and 8.4 fps, respectively.

that further decreasing Δt does not improve quality, but decreases performance. Although this transfer function is deliberately uninteresting, it shows these techniques essentially work, and deliver results comparable to increasing the world-space sampling rate, at significantly lower cost.

Peak finding exhibits similar behavior with non-separable transfer functions, like the one shown in Figure 1. This figure depicts a two-channel dataset resulting from a fluid flow application. The first variable, vorticity, encodes the magnitude of local rotation of the fluid, whereas the second variable, helicity, describes the alignment of the axis of rotation with the local flow direction. The transfer function shown in the lower right corner is aimed at illustrating vortical motion of the turbulent jet flow described in the simulation data. A vorticity isosurface (yellow) illustrates the larger region of turbulence that is the center of interest in this dataset. To extract individual vortex cores the red and blue lobes of the transfer function capture the rotational direction of the vortices in dependence of the rotational strength. This scenario is a typical representative of multi-variable volume rendering in flow analysis applications, where non-separable 2D transfer functions are employed to illustrate specific features of the flow. In [28] Tricoche et al. describe the benefits of using volume rendering with this type of transfer functions for flow visualization. In the example depicted in Figure 1 the high spatial frequency of the data in combination with the chosen transfer function necessitates the use of either very high sampling rates, or the increased fidelity provided by our method.

In Figure 7, we examine several approaches from Section 4.2 in classifying an Enzo computational astrophysics dataset [22]. Scientists use joint histograms to understand the statistical relationship between computational variables; multifield volume visualization allows us to show spatial correlation. The transfer function in this example conveys ridges in the joint histogram of density and dark matter density fields, illustrating regions where one quantity is high relative to the other. We note that the transfer function is relatively smooth, though the volume data are high-frequency. In comparing various peak finding modalities, we note that peak finding (c) delivers similar results to postclassification (b) with a 16x higher sampling rate, and at 11x the performance. Image (d) shows the effect of integrating color and opacity along the chord, similar in principle to preintegration. In this example and many others, peak finding delivers results closer to ground truth than preintegrated approaches. The bottom images (e,f,g) show results from non-chordal sampling of the transfer function domain. We see that interpolating splines (e) provide less aliased results, though with worse performance. DDA sampling (f) appears similar to sampling along the chord both in performance and quality, due to the high resolution (1024^2) of the chosen transfer function. DDA slabs (g) yield even better results, but with some artifacts due to detecting false peaks in the transfer function domain. While spline and slab methods might be appropriate in certain circumstances (such as low-resolution transfer functions) empirically we find that the simpler chord and DDA approaches yield better results.

5.2 2D Gradient Magnitude Classification

One drawback of 1D peak finding is that 1D transfer functions provide limited classification of noisy data from CT and biological sources. 2D functions mapping value and inverse gradient magnitude of univariate data offer better classification of material boundaries. Picking surface features in gradient space is an alternative to isosurfacing; but we find that peak finding is still useful in its ability to render specified features more accurately at lower sampling cost.

Figure 8 shows several examples. The backpack and Christmas tree are moderate-size CT scans with typical noise. With the backpack, peak finding clearly reproduces sharp features in the transfer function that are omitted by standard volume rendering at the same base sampling rate. Even when features are not particularly sharp in 2D TF space, as with the Christmas tree, peak finding frequently allows us to reproduce equivalent quality at a lower base sampling rate and faster overall frame rate. For medical and biological data, 2D peak finding delivers similar advantages as 1D peak finding: namely the ability to isolate and render surfaces from noisy data at higher quality with lower sampling rate.

5.3 Higher-Dimensional Multifield Data

Ray marching reduces our search in transfer space to 1D regardless of the dimension of our classification. This makes it particularly attractive for handling higher-dimensional transfer functions. In Figure 9, we have classified a 4-dimensional CFD combustion simulation [13], plotting entropy against volume mixture in one 2D

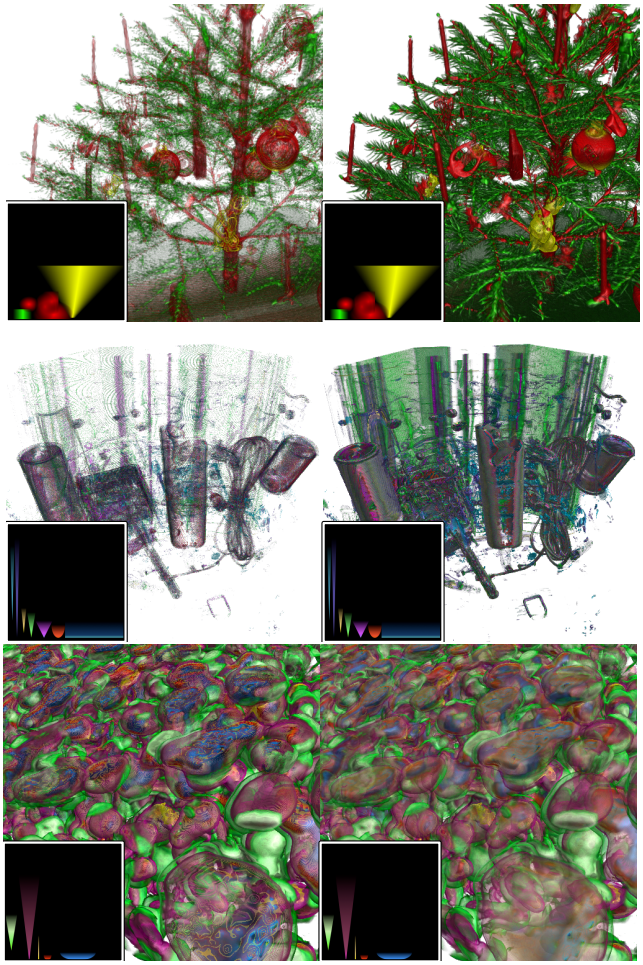


Figure 8: 2D classifications of value and inverse gradient magnitude, without (left) and with (right) peak finding. From top left to bottom right, these render at 8.0, 7.1, 5.3, 4.0, 12.3 and 9.8 fps, respectively, using scanline DDA sampling.

transfer function $\nu(f^0, f^1)$, and vorticity and a mixture fraction on another 2D function $\mu(f^2, f^3)$. We map our transfer functions on a subset of joint histograms from each data channel. To create a 4D function, we convolve ν and μ using $\rho((f)) = \rho_\alpha(f^0, f^1, f^2, f^3) = \nu\mu$. We then use chordal ray marching to peak-find directly in this 4D space.

Due to convolution of multiple variables, high frequencies are more common with multidimensional classification. As seen in Figure 9 (top), standard DVR neglects contributions from sharp isolines, and exhibits noise even at a high sampling rate (8 times the voxel Nyquist limit). Peak finding succeeds in detecting more of these features at the same sampling rate. Though some features may appear to be noise, they do not disappear with a higher peak-finding sampling rate, which indicates that they are actual features specified in the transfer function.

An even stronger argument can be made for peak finding with relatively low-frequency transfer functions in higher dimensions. With convolution of 4 variables, even the large block functions shown in the bottom examples of Figure 9 can begin to exhibit high frequencies. Again, these go unnoticed without an explicit algorithm for detecting them, and our multidimensional peak finding method excels at reproducing these features.

6 Conclusions and Future Work

Multidimensional peak finding is as good as 1D peak finding qualitatively, and useful for the same goals: rendering noisy volume data and specifying more precise transfer functions. For equivalent sampling rates, it ensures better sampling of pertinent features and produces results closer to ground truth than preintegration or separable 1D peak finding (when possible). We have applied this method to 2D gradient-magnitude classification of scalar volume data, and to direct volume rendering of regression-line features in joint histograms of multifield data. Ray marching reduces peak finding in any dimension to a 1D search, making this method applicable to 3D and higher-dimensional classifications. We have demonstrated this for 4D multifield classification, and higher dimensions are possible. We believe this technique can be a powerful tool for comparative volume visualization.

The main disadvantage of multidimensional peak finding is that it is not needed if classification is sufficiently smooth. However, as we have seen, high frequencies occur even more easily in multidimensional space than in 1D scalar fields. Since isosurfaces cause occlusion, scale-invariant volume rendering with peak finding may not be the best modality for all visualizations. In instances, post-classified rendering could be more useful than the peak-finding. Fortunately, our method provides some control over this behavior via the transfer function space sampling rate Δs. Lastly, unlike in 1D, multidimensional peak finding must sample transfer space between every world-space segment. This is expensive and unnecessary wherever $\mathbf{f}$ is monotonic. However, it does ensure scale-invariance, and in practice the cost of peak finding everywhere is small compared to its benefits.

In future work, a user interface for modeling 2D and higher-dimensional transfer functions from joint histograms would be useful. While this paper improves the efficiency of multifield volume rendering, performing meaningful multidimensional classification is at least equally important. We are interested in automatic or semi-automatic means of classifying joint histograms. Lastly, to avoid unnecessary peak finding on monotonic ray segments, topological methods might prove useful in further accelerating these techniques.

Acknowledgements

This work was supported by the International Research Training Group at the University of Kaiserslautern and the Computational Postdoctoral Fellowship at Argonne National Laboratory under the American Reinvestment and Recovery Act. We thank Rick Wagner at the San Diego Supercomputing Center for the Enzo data, Jacqueline Chen at Sandia National Laboratory for the combustion data, and Tolga Tasdizen at the University of Utah for the zebrafish data.

References

[1] M. Ament, D. Weiskopf, and H. Carr. Direct Interval Volume Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1505–1514, 2010.

[2] S. Bergner, T. MŽller, D. Weiskopf, and D. J. Muraki. A spectral analysis of function composition and its implications for sampling in direct volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12:2006, 2006.

[3] J. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

[4] B. Cabral, N. Cam, and J. Foran. Accelerated Volume Rendering and Tomographic Reconstruction using Texture Mapping Hardware. In *VVS '94: Proceedings of the 1994 Symposium on Volume Visualization*, pages 91–98, New York, NY, USA, 1994. ACM Press.

[5] T. J. Cullip and U. Neumann. Accelerating Volume Reconstruction With 3D Texture Hardware. Technical report, University of North Carolina at Chapel Hill, 1994.

[6] 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*, pages 9–16. ACM New York, NY, USA, 2001.

[7] M. Hadwiger, C. Sigg, H. Scharsach, K. Bühler, and M. Gross. Real-Time Ray-Casting and Advanced Shading of Discrete Isosurfaces. *Computer Graphics Forum*, 24(3):303–312, 2005.

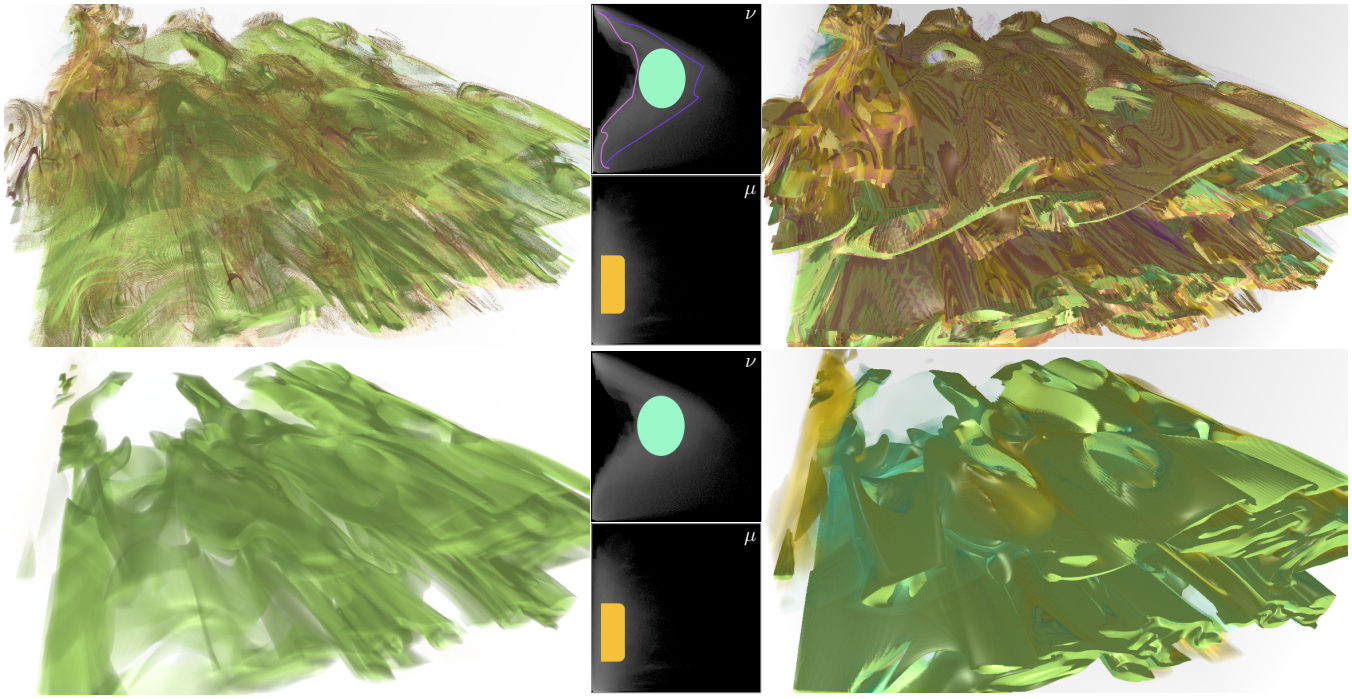


Figure 9: Visualization of 4-channel combustion simulation data [13] using a 4D transfer function modeled by analytically convolving two 2D transfer function textures ν and μ. Postclassified renderings are on the left and peak finding renderings are on the right. The top row shows a transfer function with peaks, purple and pink isolines. The bottom row shows results with the low-frequency components of the same function. With $\Delta t = 0.5$ for the postclassified examples and $\Delta = 1, \Delta s = 2$ for peak finding with chordal ray marching, these frames render at 0.8, 0.75, 1.6 and 1.6 fps, respectively at 1200×600.

[8] G. Kindlmann. Transfer Functions in Direct Volume Rendering: Design, Interface, Interaction. *Course notes of ACM SIGGRAPH*, 2002.

[9] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller. Curvature-based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 67. IEEE Computer Society, 2003.

[10] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional Transfer Functions for Interactive Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

[11] J. Kniss, S. Premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization 2003*, pages 497–504, October 2003.

[12] A. Knoll, Y. Hijazi, R. Westerteiger, M. Schott, C. Hansen, and H. Hagen. Volume Ray Casting with Peak Finding and Differential Sampling. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1571–1578, Nov-Dec 2009.

[13] Knollman, J. W. Chen, and H. Im. Combined Pdf-Sdf Approach to Partially Premixed Turbulent Combustion. In *Proceedings of the 28th Symposium on Internal Combustion, Edinburgh, Scotland*, 2000.

[14] M. Kraus. Scale-Invariant Volume Rendering. In *Proceedings of IEEE Visualization 2005*, pages 295–302. IEEE, 2005.

[15] M. Kraus. Pre-Integrated Volume Rendering for Multi-Dimensional Transfer Functions. *IEEE/ EG Symposium on Volume and Point-Based Graphics*, pages 1–8, 2008.

[16] J. Krüger and R. Westermann. Acceleration Techniques for GPU-based Volume Rendering. In *Proceedings IEEE Visualization*, pages 287–292, 2003.

[17] D. Laidlaw. *Geometric Model Extraction from Magnetic Resonance Volume Data*. PhD thesis, California Institute of Technology, 1995.

[18] M. Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[19] L. Linsen, T. Long, P. Rosenthal, and S. Rosswog. Surface Extraction from Multi-field Particle Volume Data using Multi-dimensional Cluster Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1483–1490, 2008.

[20] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics (Proceedings of ACM SIGGRAPH)*, 21(4):163–169, 1987.

[21] P. Navratil, J. Johnson, and V. Bromm. Visualization of Cosmological Particle-based Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1712–1718, 2007.

[22] M. L. Norman, G. L. Bryan, R. Harkness, J. Bordner, D. Reynolds, B. O'Shea, and R. Wagner. Simulating Cosmological Evolution with Enzo. *ArXiv e-prints*, May 2007.

[23] S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P.-P. Sloan. Interactive Ray Tracing for Isosurface Rendering. In *Proceedings of IEEE Visualization '98*, pages 233–238, October 1998.

[24] S. Roettger, S. Guthe, D. Weiskopf, T. Ertl, and W. Strasser. Smart Hardware-accelerated Volume Rendering. In *Proceedings of the Symposium on Data Visualisation (VISSYM)*, pages 231–238, 2003.

[25] S. Röttger, M. Kraus, and T. Ertl. Hardware-Accelerated Volume and Isosurface Rendering based on Cell-Projection. In *Proceedings of IEEE Visualization '00*, pages 109–116. IEEE Computer Society Press Los Alamitos, CA, USA, 2000.

[26] J. Schulze and A. Rice. Real-time volume rendering of four channel data sets. In *Proceedings of IEEE Visualization '04*, pages 598–34. IEEE Computer Society, 2004.

[27] M. Sramek. Fast Surface Rendering from Raster Data by Voxel Traversal Using Chessboard Distance. *Proceedings of IEEE Visualization 1994*, pages 188–195, 1994.

[28] X. Tricoche, C. Garth, G. Kindlmann, E. Deines, G. Scheuermann, M. Ruetten, and C. Hansen. Visualization of intricate flow structures for vortex breakdown analysis. *Proceedings of IEEE Visualization '04, 2004*, pages 187–194, 2004.

[29] L. Westover. Footprint Evaluation for Volume Rendering. In *SIGGRAPH '90: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, pages 367–376, New York, NY, USA, 1990. ACM Press.

[30] X. Wu. An Efficient Antialiasing Technique. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, pages 143–152. ACM, 1991.