Occam's razor and petascale visual data analysis

E. W. Bethel[1], C. Johnson[2], S. Ahern[3], J. Bell[1], P.-T. Bremer[5], H. Childs[1], E. Cormier-Michel[1], M. Day[1], E. Deines[4], T. Fogal[2], C. Garth[4], C. G. R. Geddes[1], H. Hagen[7], B. Hamann[4], C. Hansen[2], J. Jacobsen[1], K. Joy[4], J. Krüger[2], J. Meredith[3], P. Messmer[6], G. Ostrouchov[3], V. Pascucci[2], K. Potter[2], Prabhat[1], D. Pugmire [3], O. Rübel[1], A. Sanderson[2], C. Silva[2], D. Ushizima[1], G. Weber[1], B. Whitlock[5], K. Wu[1]

[1]Lawrence Berkeley National Laboratory
[2]University of Utah
[3]Oak Ridge National Laboratory
[4]University of California, Davis
[5]Lawrence Livermore National Laboratory
[6]Tech-X Corporation
[7]Technische Universität Kaiserslautern, Germany

E-mail: `ewbethel@lbl.gov`

Abstract. One of the central challenges facing visualization research is how to effectively enable knowledge discovery. An effective approach will likely combine application architectures that are capable of running on today's largest platforms to address the challenges posed by large data with techniques with visual data analysis techniques that help find, represent, and effectively convey scientifically interesting features and phenomena.

1. Introduction

The principle *Occam's Razor*, which is attributed to the 14th century English logician and Franciscian Friar William of Ockham, is roughly translated from Latin as "entities must not be multiplied beyond necessity"[1]. The principle suggests that when there are multiple competing hypotheses, that the simplest one is usually the correct explanation. Another interpretation, and one we focus on here, is that "less is more." Within the context of modern computational science and visual data exploration and analysis, applying this principle yields a fruitful course of research and development: simulations and experiments produce larger and more complex data, yet the sheer size and complexity of these results confounds attempts to gain insight; insights are drawn from simple yet elusive observations.

Our colleagues in the computational and experimental sciences are using increasingly powerful computational technology to their advantage. The advantage for science is that there is the opportunity to model scientific phenomena with increasing physical realism and to collect and store data at increasingly finer spatiotemporal resolution. Data from both simulations and experiments exhibits increasing spatial resolution: climate models, for example, are moving towards sub-kilometer resolution, resulting in data files that are unwieldy at best and that at

[1] `http://en.wikipedia.org/wiki/Occam's_razor`

worst confound existing analysis and visualization algorithms and implementations. The grid or sample values are evolving from simple scalars or vectors to more complex types like tensors, functions, and distributions. For some of these data types, there do not exist any satisfactory method for visual data exploration. Large and distributed systems enable larger ensemble runs, which produce many more datasets. The process of studying the relationships between many datasets and input parameters grows increasingly difficult due to the sheer volume of data and the potential complexity of the analysis. More powerful platforms enable codes to model phenomena for a longer duration in simulation time, thereby increasing temporal complexity.

From a visual data analysis perspective, the resulting challenges are numerous and profound and provide the basis for much of the current research in our community. For example, [1] concludes that our ability to generate data far exceeds our ability to gain insight from such data using existing techniques, even if scaled up to run on large, parallel platforms. This theme restates the findings of an earlier work [2] pointing out that the sheer size and complexity of scientific data is itself an impediment to scientific insight.

Insight comes in many forms and via many different paths. From a visualization perspective, there are three broad visualization use modalities [3]. "Presentation visualization" is where you know what is there and want to show it to someone else. "Analytical visualization" is where you know what you are looking for. "Discovery visualization" occurs when you have no idea what you're looking for. Discovery visualization is characterized as an "undirected search," or "unconstrained navigation" through visualization or rendering parameter space. One might expect that most "unexpected" discoveries would occur in discovery and analytical visualization, with these unexpected results clearly communicated to others through presentation visualization techniques.

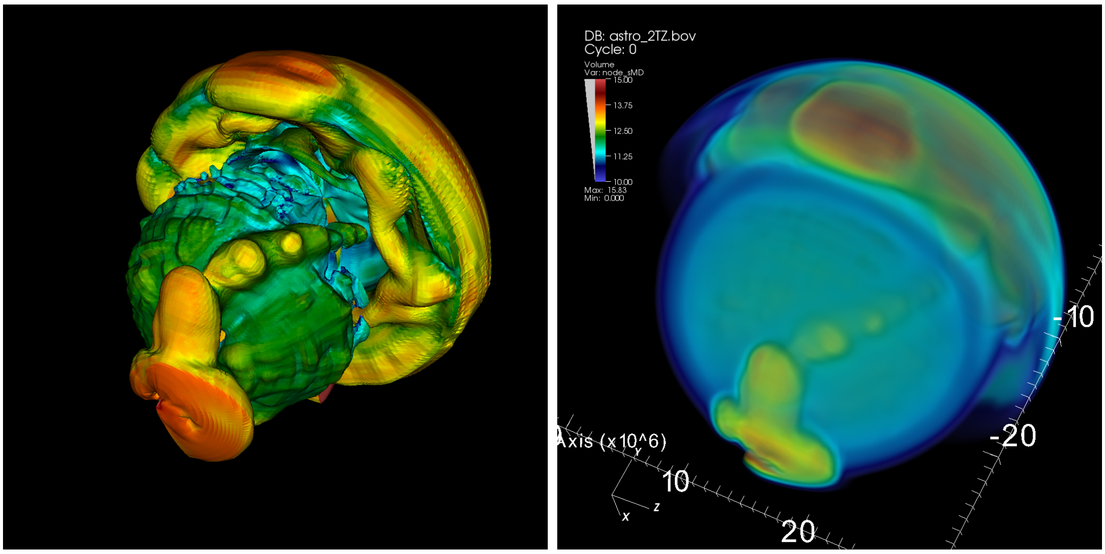
One central thesis of this article is that focusing visualization research on techniques for discovery and analytical visualization with a special emphasis on the challenges posed by science in the petascale era will yield the greatest impact, or potential for enabling scientific knowledge discovery. One conclusion we hope to convey here is that simply making higher resolution images of higher resolution data exacerbates the problems pointed out in earlier reports, e.g., [2], where the scientist is burdened with more work. Instead, our approach is to realize an effective implementation of Occam's Razor, where petascale visual data analysis produces less work for the scientist, thereby increasing the likelihood of scientific insight.

We discuss this theme in several different dimensions in this article. First, we, like many other scientific fields, seek to leverage increasingly powerful computational platforms to tackle ever larger problems. We present results in Section 2 where we show effective use, from a visual data analysis perspective, of today's largest platforms on problem sizes of unprecedented scale. While this result is noteworthy in its own right, Section 3 takes a different approach by combining advances in visualization and data management to tackle a hero-sized problem in accelerator modeling to replace a manual process that previously required hours with yesterday's "small" datasets with one that takes seconds on today's "hero-sized" datasets. Next in Section 4, we present work that uses advances in computational geometry to discover elusive and difficult to quantify features in scientific data to produce insights never before possible using traditional visualization or analysis technologies. Finally, Section 4 presents advances in visualization technology that help to convey deeper understanding into flow-based data.

2. Petascale Visual Data Exploration and Analysis

2.1. Introduction

As the by-product of technology advances is "more and more data," one issue facing the visualization and analysis community is the feasibility of using today's largest computational platforms for knowledge discovery. To gain better insight into this issue, VACET researchers recently conducted a series of experiments aimed at fostering a better understanding of functional and performance limits that might be encountered when running a production-quality visualization application at extreme levels of concurrency on datasets of unprecedented size. The results, which we discuss in this section, suggest this approach is viable and that visualization research and development efforts have produced technology that is today capable of ingesting and processing tomorrow's datasets.



(a) Isocontouring of two trillion zones on 32K Cray cores of jaguarpf, a Cray XT5 at OLCF/ORNL.

(b) Volume rendering of two trillion zones on 32K cores of franklin, a Cray XT4 at NERSC/LBNL.

Figure 1. Our functional performance experiments consist of loading extremely large datasets and executing visualization algorithms at extreme levels of concurrency. These imares were produced as part of those experiments.

Another purpose of these runs was to prepare for establishing VisIt's credentials as a "Joule code," or a code that has demonstrated scalability at a large number of cores. VisIt is the first and only visual data analysis code that is part of the ASCR Joule metric, which aims to track code performance (scalability) over a period of time.

The team's experiments consisted of running the VisIt software application [4] on several of the nation's largest computing platforms and on dataset sizes ranging from 500 billion (two terabytes per scalar) to 2 trillion cells (eight terabytes per scalar) and at concurrency levels ranging from 8K to 32K cores. Each experiment consisted of running VisIt in parallel: loading in data, performing two common visualization tasks (isosurfacing and volume rendering), and producing an image (Figure 1).

2.2. Parallel Processing of Data

As there is some variation in how tools process data in parallel, we will describe VisIt's approach for parallel processing. VisIt employs a client-server model: the VisIt server is parallelized using an MPI-based communication model and can run on a large, parallel platform; and the VisIt client, which runs on the user's workstation, communicates with the VisIt server over an IP-based connection. To begin, the end user sets up the visualization or analysis task from the VisIt client-side interface. The client then communicates this information to MPI rank 0 of the server. MPI rank 0, in turn, communicates this information to the other processors through a broadcast command. Each processor then sets up a data flow network, which consists of modules to read data, process data, and ultimately to render data to images. VisIt's parallelization approach uses domain decomposition so that each MPI process, which runs data I/O, processing, and rendering, operates only on its data subset. The partitioning occurs by operating on meta-data, for example by partitioning pre-defined domains.

Once the data flow networks are set up on each MPI process, execution begins. First, each process sets up its reader to load only the data for its subset of the larger data set. Next, each process executes one or more algorithms (e.g., isocontouring) with user-specified parameters parameters (e.g., isocontouring level). For processing sequences that includes rendering, each processor renders its portion of the problem into an image, then the resulting images from all processes are assembled into a final image. The resulting image is then transferred to the client and displayed to the user. The processing and rendering for volume rendering is somewhat different, because volume rendering algorithm is not an embarrassingly parallel algorithm. That approach is described in more detail in [5].

2.3. Experiment Description

One goal of this experiment was to demonstrate the viability of these techniques across diverse supercomputing environments, in terms of operating system, I/O performance, FLOPs, and network bandwidth. We performed these tests on on Crays (OLCF/ORNL's Jaguar & NERSC/LBNL's Franklin), a Sun Linux machine (TACC's Ranger), a CHAOS Linux machine (LLNL's Juno), and an AIX machine (LLNL's Purple).

Because each machine has a different number of cores, we performed a weak scaling study. We started with a single data set, which we then upsampled to an appropriate resolution. This data set was from a core-collapse supernova simulation done by the CHIMERA code, on a curvilinear mesh of more than three and one half million cells[2]. We chose the upsampling approach since we are not aware of any current datasets containing 2T cells and since the primary objective for our studies is to better understand the performance and functional limits of parallel visual data analysis software. These objectives can be achieved using upscaled data.

The upsampling process involved interpolating a scalar field onto a high resolution rectilinear mesh and then writing the data out as compressed binary data (gzipped). There were ten files for every core used, and every file contained 6.25 million data points, for a total of 62.5 million data points per core. In our experience, the visualization tool often has one tenth (or less) of the resources (e.g. cores) as the simulation code and simulations codes often write out one file per core. Hence, having multiple files per core was our best approximation at emulating these common real world conditions.

We ran with 16000 cores on each machine visualizing one trillion cells[3]. On the Jaguar and Franklin machines, we ran additional, larger-sized problems consisting of two trillion cells on 32000 cores. Although times varied from machine to machine, I/O was the dominant factor (taking two or more minutes at 16000 cores), with contouring taking approximately ten seconds and rendering taking one to ten seconds.

2.4. Issues Discovered During Scaling Study

Although the majority of VisIt's infrastructure scaled well to a large number of cores, we ran into several obstacles along the way:

- VisIt's MPI rank 0 was collecting status information from all other processors through point-to-point communications, which is non-scalable. We worked around this issue for our study and plan on implementing a complete solution for a future release of VisIt.

[2] Sample data courtesy of Tony Mezzacappa (ORNL), Bronson Messer (ORNL), Steve Bruenn (Florida Atlantic University) and Reuben Budjiara (University of Tennessee).

[3] We ran with only 8000 cores and one half trillion cells on Purple, because the full machine has only 12208 cores, and only 8000 are easily obtainable for large jobs.

- VisIt's volume rendering algorithm was attempting an optimization for sample point communication that required a buffer that had an $O(nProcs \times nProcs)$ space requirement. This "optimization," while appropriate for low levels of concurrency, proved to cause problems at high levels of concurrency. The workaround for these tests was to remove this "optimization." We are evaluating the best course of action for a solution for a future release of VisIt.
- We observed that the loading of shared libraries took quite a long time at scale (as much as five minutes) and VisIt's plugin model may need to adapt for this case, likely by switching to a static binary with precompiled plugins.

2.5. Lessons Learned

The primary objective for these experiments was to gain a better understanding of functional and performance limits when running visual data analysis applications at extreme levels of concurrency and and problem sizes. We encountered a couple of minor problems that will be rectified and appear in a future public VisIt release.

From a visual data analysis perspective, these problem sizes and concurrency levels are a "first." The successful completion of these functional and performance tests show progress towards petascale computing by demonstrating that today's technology is capable of ingesting and processing tomorrow's datasets.

The performance data the team collected during the experiments reveals insights into potential bottlenecks and opportunities for performance optimization on different machine architectures at high levels of concurrency and ultrascale datasets. Future work will include a more detailed, end-to-end performance study of several different visualization algorithms to better understand performance limits and opportunities for VisIt, a production-quality visual data analysis software application.

3. Accelerator Modeling

3.1. Background

Laser wakefield simulations model the behavior of individual particles as well as the behavior of the plasma electric and magnetic fields. Output from these simulations can become quite large: today's datasets, such as the ones we study here, can grow to be on the order of 200GB per timestep, with the simulation producing ≈ 100 timesteps. The scientific challenge we help address in this study is first to quickly find particles that have undergone wakefield acceleration, then trace them through time to understand acceleration dynamics, and perform both visual and quantitative analysis on the set of accelerated particles. Visualization and analysis of *all* particles modeled and output by the simulation would not directly help answer scientific questions.

To solve this problem, we focused on combining and extending two different but complementary technologies aimed at enabling rapid, interactive visual data exploration and analysis of contemporary scientific data [6]. To support highly effective visual data exploration, knowledge discovery and hypothesis testing, we have adapted and extended the concept of parallel coordinates, in particular binned or histogram-based parallel coordinates, for use with high performance query-driven visualization of very large data. In the context of visual data exploration and hypothesis testing, the parallel coordinates display and interaction mechanism serves multiple purposes. First, it acts as a vehicle for visual information display. Second, it serves as the basis for the interactive construction of compound Boolean data range queries. These queries form the basis for subsequent "drill down" or data mining actions. To accelerate data mining, we leverage state-of-the-art index/query technology to quickly mine for data of interest as well as to quickly generate multiresolution histograms used as the basis for the visual display of information. This combination provides the ability for rapid, multiresolution visual data exploration, and implements an easy-to-use interface for composing multivariate and multidimensional queries in a high performance, query-driven visualization and analysis system [7].

We apply this new technique to large, complex scientific data created by a numerical simulation of a laser wakefield particle accelerator. In laser wakefield accelerators, particles are accelerated to relativistic speeds upon being "trapped" by the electric fields of plasma density waves generated by the radiation pressure of an intense laser pulse fired into the plasma. These devices are of interest because they are able to achieve very high particle energies within a relatively short amount of distance when compared to traditional electromagnetic accelerators. The VORPAL [8] simulation code is used to model experiments, such as those performed at the LOASIS facility at LBNL [9], and is useful in helping to gain deeper understanding of phenomena observed in experiments, as well as to help formulate and optimize the methodology for future experiments.

One scientific impact of our work is that we have vastly reduced the duty cycle in visual data exploration and mining. In the past, accelerator scientists would perform the "trace backwards" step using scripts that performed a search at each timestep for a set of particles. Runtimes for this operation were on the order of hours. Using our implementation, those runtimes are reduced from hours to seconds.

In our implementation, a user will see an "overview" of all data in the multidimensional dataset, then select subsets for subsequent visual data exploration and analysis. The multidimensional presentation of data – both the "overview" as well as subset selections – is a statistical display known as a *parallel coordinates* plot. Examples of such visual data exploration and analysis are shown in Figure 2, and are described in more detail in [6].

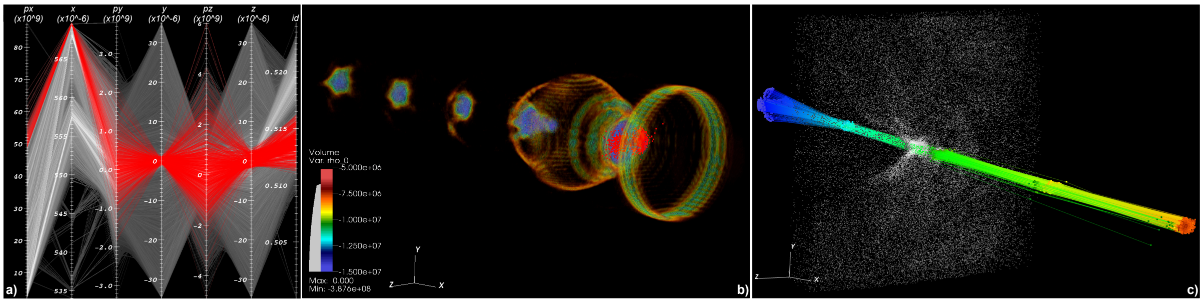


Figure 2. The interface on the left, which shows a multivariate view of particle distribution across all simulation variables, is a vehicle for selecting particles having high acceleration and spatial coherency. All particles are shown in gray, and selected particles shown in red. This selection interface is linked with other forms of visual data exploration, so that a physical view of selected particles (middle, red particles) helps the scientist to quickly gain insight into the relationship between statistical-space and physical-space features. Paths of "interesting" particles over a range of timesteps, shown on the right, are colored to indicate particle momentum. That image shows particles are constantly accelerated over this range of time. This new capability, developed as part of VACET's research and development portfolio, is now part of widely distributed, production-quality, parallel capable visual data exploration software infrastructure (VisIt). Image, courtesy O. Rübel et al., LBNL, generated from VORPAL simulation results, courtesy C. Geddes et al., LBNL.

3.2. Multidimensional Data Display: Histogram-Based Parallel Coordinates

Parallel coordinates, believed to be introduced in 1885[4], provide an effective interface for displaying multivariate data and for defining multi-dimensional queries (range selections) based on thresholding. The idea of combining parallel coordinates displays with physical space views of subset selections is not new. Applications range from multidimensional geometry [10, 11], GIS/remote sensing [12], data mining [13], currency and economic analysis [14]. These concepts have had success in the visual data exploration and analysis of particle-based data in life sciences data [15–17], and multiple teams concurrently and explored applying these concepts to fusion simulation data [18].

Using sliders attached to each axis of the parallel coordinates plot, a user defines range thresholds in each displayed dimension. By rendering the user-selected data subset (the focus view) in front of a parallel coordinates plot created from the entire data set (or in many cases a subset of it) (the context view), the user receives immediate feedback about general properties of the selection. Data outliers stand out visually as single or small groups of lines diverging from the main data trends. Data trends appear as dense groups of lines (bright colored bins in our case). A quick visual comparison of the focus and context views helps to convey understanding about similarities and differences between the two.

In practice, parallel coordinates have disadvantages when applied to very large datasets. First, each data record is represented with a single polyline that connects each of the parallel coordinates axes. As data size increases, the plot becomes more cluttered and difficult to interpret. Second, data records drawn later will occlude information provided by data records drawn earlier. Finally and worst of all is the fact this approach has computational and rendering complexity that is proportional to the size of the dataset. As data sizes grow ever larger, these problems become intractable.

To address these problems, which stem from the attempt to use an existing visualization technique on very large datasets, we developed an efficient rendering technique based on two-dimensional (2D) histograms. Rather than viewing the parallel coordinates plot as a collection of polylines, one per data record, we approach rendering by considering instead the relationships of all data records between pairs of parallel coordinate axes. That relationship can be discretized as a 2D histogram and then later rendered. This idea was introduced in earlier work [19], and we extended the basic idea to accomodate very large datasets through a combination of adaptive histograms, novel rendering, and rapid histogram computation.

Complementary to the histogram-based rendering technique is the need to quickly compute histograms from large data and user selections. Our approach is to leverage FastBit to quickly generate conditional histograms. This approach has proven useful in the past in a "hero-sized" cybersecurity visual analytics and data mining application [20]. The key here is to provide an implementation that runs sufficiently quickly as to enable *interactive* visual data exploration of large particle-based datasets. Our implementation, described in more detail in [6], achieves such performance levels through extensions to FastBit and use on parallel computing platforms to accelerate I/O and histogram computation.

3.3. Particle Paths and Visual Data Analysis

Once "interesting" particles are located, the next step in the knowledge discovery process is to learn something about the temporal characteristics of these particles. In our earlier work [6], we iteratively formulate complex multidimensional queries to isolate interesting particles and display them. Typically, the multidimensional query is applied at a timestep after particles have begun to undergo acceleration. From that query results a list of particles; we then use the FastBit machinery to find those named particles across all simulation timesteps and display

[4] See `http://en.wikipedia.org/wiki/Parallel_coordinates`

them.

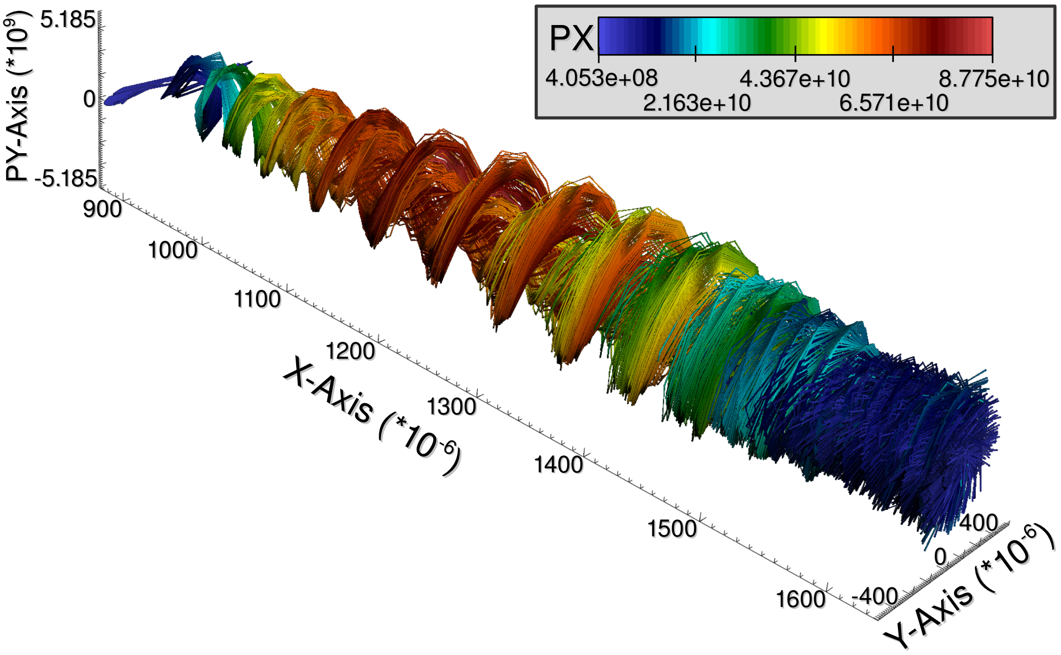


Figure 3. Particle paths colored by x-momentum so that color indicates the level of acceleration. The spiral structure of the paths illustrates the particles' oscillating motion over time. Image courtesy O. Rübel et al., LBNL.

Figure 3 shows recent results where we employ visual data analysis techniques to help reveal more scientifically interesting information. Figure 3 shows a subset of high-energy particles traced through time. The paths are shown in (x, y, py) space colored according px so that color indicates the level of acceleration of particles. The particles move along the paths from left to right. The spiral structure of the paths illustrates the oscillating motion of the particles in y. Along any given particle pathline, color transitions reveal interesting phenomena. First, particles exhibit low momentum in x when they first enter the simulation (blue at the left). Second, after being injected, the particle are constantly accelerated (transition from blue to red). Finally, later in the simulation – at $x \approx 1250\mu m$ – the particles "outrun" the wave and decelerate (transition from red to blue). Figure 3 shows the paths only until the particles have completed the deceleration process.

3.4. Knowledge Discovery: Traditional Analysis

Complementary to high performance visual data exploration technology, which helps with discovering characteristics of particles undergoing acceleration, are automated methods for finding and analyzing such particles of interest. Recent work [21] combines signal processing and machine learning techniques to locate particles undergoing a high degree of acceleration and that are spatiotemporally compact. Once individual particles of interest have been identified, they are grouped into compact "bunches" using unsupervised fuzzy clustering. Concurrently, the pathlines of the bunches are derived using a graph-based algorithm. This approach proved sufficiently robust to successfully identify high-energy beams in multiple datasets that exhibit markedly different characteristics and behavior. This type of approach holds promise for helping to accelerate scientific knowledge discovery by finding and analyzing "features" that are difficult to define and that vary from dataset to dataset.

4. Topologically Based Feature Detection and Tracking

4.1. Introduction: Topological Data Analysis

In recent years, topological structures have proven to be very useful for building very general "feature" definitions [22] in both visualization and quantitative data analysis. The success of these feature definitions stems from the fact that topological structures are based on fundamental properties of isosurfaces. An isosurface [23, 24] connects all locations in a domain where the scalar field assumes a specified isovalue and is an extremely versatile and ubiquitous visualization and analysis component. By varying the isovalue and observing changes to the corresponding isosurface, one gains considerable insights into properties of that scalar. Isosurfaces arise naturally in data analysis since distinct isosurfaces often have a useful physical interpretation directly related to the application domain. For example, in premixed combustion simulations, an isotherm (a surface of constant temperature) of an appropriate temperature is often associated with the location of the flame.

In the context of scalar field analysis, topological properties provide a quantifiable characterization of fundamental isosurface properties: the number connected isosurface components and the genus of an isosurface (i.e., the number of independent tunnels or "holes" in the surface). Morse theory [25] provides insight into the topological evolution of isosurfaces of scalar functions: isosurface topology changes at distinct isolated *critical points* in the scalar field that can be identified and analyzed analytically or combinatorially.

Critical points identified by the Morse theory constructions provide hints about the location of interesting behavior in a scalar field. However, as a set of individual disconnected points they are only of limited use to decipher global structures in a data set. Topological structures relate critical points to each other and provide a means for ranking topological features and simplifying the global topological structure [26, 27]. This simplification supports further "boiling down" of the data to essential features and extracting a simpler, overall coarser structure.

The *Reeb graph* [28] is one of these structures; it expresses the evolution of individual contours as a graph that is defined by these critical points and their relationships. The Reeb graph describes the evolution of contours, i.e., individual connected components of a level set, as the value for which we extract the level set changes. Essentially, the Reeb graph is a skeleton of the manifold. Edges of the Reeb graph correspond to families of topologically equivalent contours, and nodes correspond to critical points where the number of contours change. For simply connected domains, the Reeb graph is always a tree structure, called a *contour tree* [29].

The *Morse complex* and the *Morse-Smale complex* [30–32] comprise alternate means of relating critical points to each other. Assuming the function on the manifold is differentiable, and thus the gradient is defined at each location, it is possible to start at any location in the domain and follow a gradient line either to its origin or its destination, i.e., follow a line of steepest descent or ascent from that point. This line of steepest ascent/descent will end in a local maximum/minimum. The Morse complex is the segmentation of the domain into ascending or descending manifolds of the critical points. An ascending/descending manifold of a maximum/minimum is the union of all locations in the domain for which the path of steepest ascent/descent ends at that maximum/minimum. Superimposing ascending and descending manifolds results in the Morse-Smale complex.

Using contour tree simplification, it is possible to manipulate isosurfaces in terms of fewer, important connected components [26]. Furthermore, one can represent the contour tree in a hierarchical layout that makes it possible to explore the global structure of a data set and subsequently obtain a more detailed topological representation in regions of interest [33]. Finally, it is possible to represent the topological information contained in the contour tree in a more intuitive fashion [34]. Use of a multiresolution-representation of the Morse-Smale complex [27, 32] has also proven useful in analyzing scientific simulation data. For example, it is possible to specify features, such as bubbles, that are difficult to characterize using traditional feature detection methods, and use this definition for analyzing turbulent mixing simulations [35]. More recent work used the Morse-Smale complex for calculating clean distance fields in porous solids hit by a projectile [36] and used the results to calculate measures like filament density. Other topological structures, such as Jacobi sets, allow relating critical points between different functions/simulations [37]. Recently, Bremer et al. [22] gave an overview of topology-based feature definition and feature tracking.

4.2. Science Application: Analysis of Turbulent, Premixed Hydrogen Flames

Understanding combustion processes over a broad range of operational regimes is of great interest for a variety of applications like engine or power plant design. To this end, there has been considerable recent interest in developing premixed burners capable of stably burning ultra-lean hydrogen-air fuel mixtures. Such burners could, for example, be used as one component of a clean-coal power plant utilizing hydrogen extracted from coal gasification. Lean premixed systems are subject to a variety of hydrodynamic and combustion instabilities that render practical flame stabilization, and traditional approaches to flame analysis, extremely difficult. The flames burn in a cellular mode that is highly nonuniform, time-dependent, and difficult to characterize [38].

To study this combustion process, researchers at the Center for Computational Sciences and Engineering at Lawrence Berkeley National Laboratory performed numerical simulations of lean premixed hydrogen flames at three different levels of turbulence: no turbulence, weak turbulence, and strong turbulence. Each flame burns in a volume periodic in the x- and y-coordinate with pre-mixed fuel being injected through the plane $z = 0$. The flame simulations were carried out with a low Mach number model that incorporates a detailed description of the combustion kinetics and differential species transport. For more details of the model and its implementation, see [39]; for example analysis of lean hydrogen flames in this regime, see [40]. Each solution analyzed here is represented as a sequence of three-dimensional snapshots, taken at uniform intervals in time, where each snapshot consists of cell-centered data on a uniform Cartesian grid describing the temperature, chemical composition, and effective fuel consumption rate. The “flame” is represented as an isotherm extracted from the datasets using standard techniques. The local fuel consumption rate, interpolated to these isosurfaces, is used to divide the surface into burning “cells” separated by non-burning regions, as defined by a threshold in the consumption rate. To gain new insights into the combustion process, scientists are interested in two types of analysis: time-aggregate statistics and detailed analysis of cell evolution.

4.3. Characterizing Influence of Turbulence

The top row of Figure 4 shows isotherms of combustion simulations performed at the three turbulence levels under consideration. Mapping the fuel consumption rate to isotherm color, the cellular structure of the burning process becomes immediately obvious. Essentially we are interested in the influence of turbulence level on the size and evolution of these burning cells.

It is possible to use the Morse complex to study the evolution of the cellular burning patterns observed in lean premixed hydrogen-air flames. One can construct a quantitative analysis through segmentation of this flame isotherm based on threshold values of the local burning rate. This thresholding divides the isotherm contour into individual burning cells that evolve in time. Such a definition allows a quantitative analysis of the influence of turbulence on the evolution of these flame structures, for example. However, the network of these burning cells (and its dual, the network of extinction regions) may depend strongly on the threshold that determines the segmentation, and there is no universally accepted threshold value. It is crucial then to examine the statistical characterization of these flame cells as a function of the threshold value to ensure that the conclusions drawn from the analysis do not depend on the detailed values of this rather arbitrary parameter [38]. The Morse complex [41] provides a framework to represent

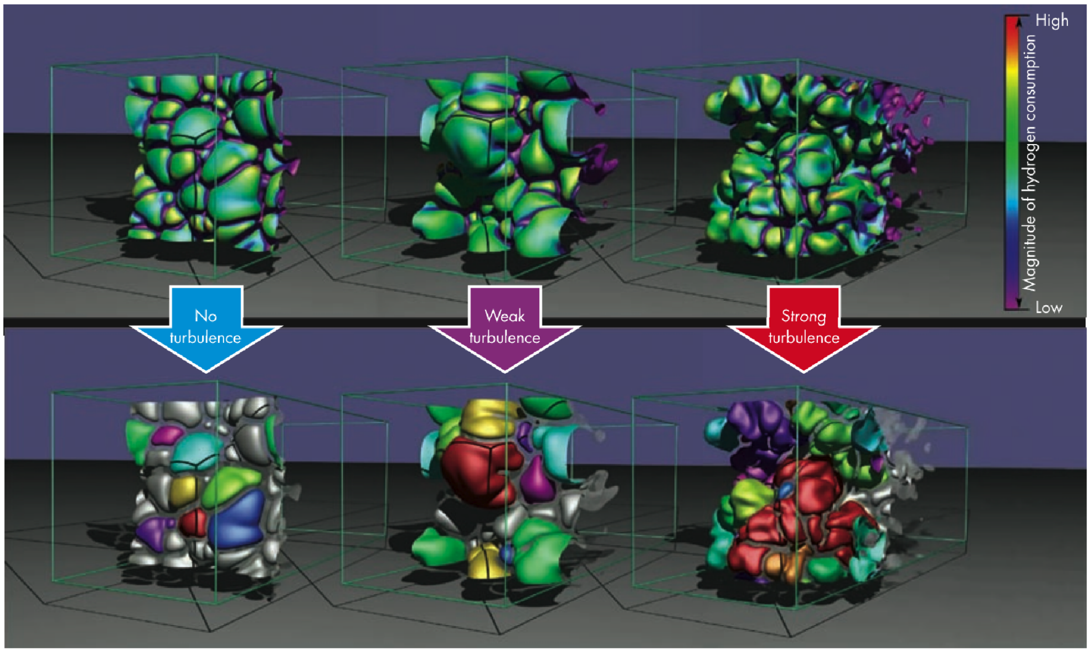


Figure 4. The top 3D panels show flame surfaces of a lean premixed hydrogen flame at different levels of turbulence colored by the local fuel consumption. In the bottom 3D panels a small set of burning cells are randomly colored to show the irregularity of the more turbulent cells.

segmentations for all possible thresholds and thus for evaluating the segmentation for a range of thresholds and determining the stability of these arbitrary parameters. Furthermore, it supports simplifying the topological structure of burning regions, merging very small burning regions into larger ones, thus simplifying analysis.

Once an appropriate choice of parameters has been determined we track the corresponding cells over time. A tracking graph can be constructed efficiently from a three-dimensional hypersurface (embedded in four-dimensional space-time) of the evolution of the flame isotherm over time. Based on this hypersurface and a segmentation in each time step, boundaries of burning regions as they evolve over time can be constructed. The Reeb graph of this system (Figure 5) can efficiently represent a complete tracking graph of the evolution of the burning regions over time, including their creation, destruction, splitting and merging behavior, and provides a basis for quantitative analysis of their evolution.

4.4. Lessons Learned

Our methods allow, for the first time, a quantitative analysis of the cellular burning structures (Figure 6), and yield important scientific insights: Contrary to common intuition, higher turbulence levels, which generally lead to increased energy in the finer length scales of the approach flow to the flame, apparently lead to *larger* cellular burning structures. Moreover, these larger cells tend to burn more intensely than would be expected from simple theories of flame propagation. The combination of these two effects dramatically increases the global propagation speed of the turbulent flames over the steady flat flame case.

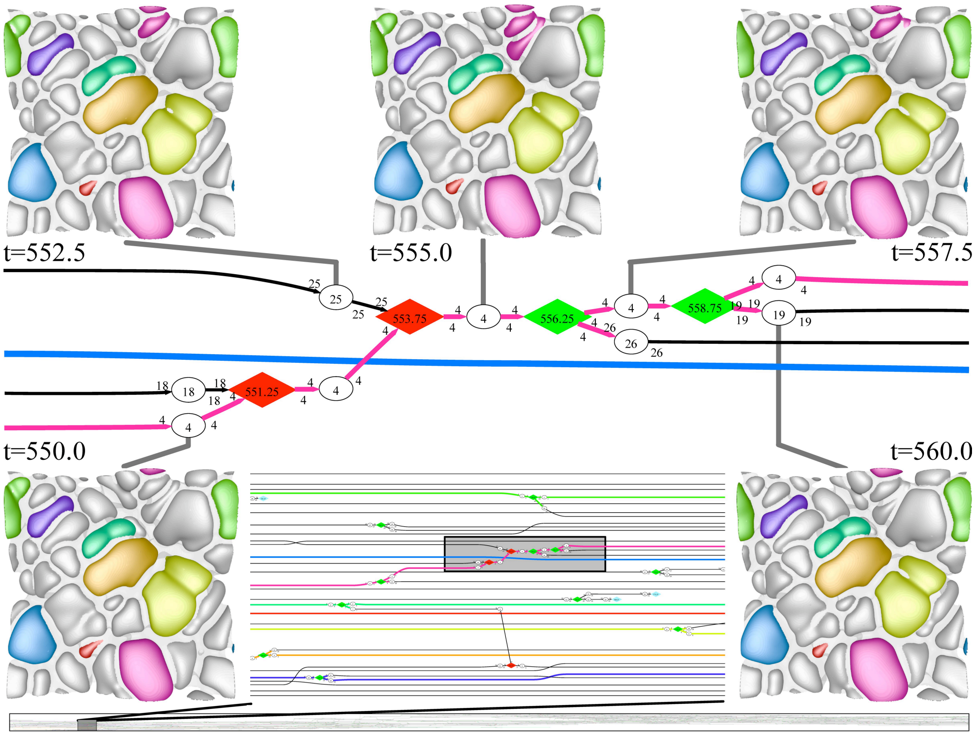


Figure 5. Subsection of the tracking graph of a combustion simulation of the "no turbulence" case that is compressed to only show two merge and two split events. Arc color corresponds to region color in corresponding segmentations. Round nodes correspond to cells explicitly segmented by the Morse complex, diamonds to topological events between time steps. Red signifies a merge, green a split, and turquoise a birth/death event. The figure shows three zoom levels of the graph: graph for the entire simulation (bottom graph), a portion corresponding to several subsequent time steps (middle graph) and a zoom into the two merge and split events annotated with colored segmentations of the isotherm.

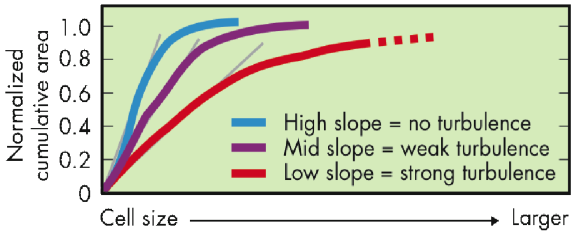


Figure 6. Graph showing the cumulative density function of cell area distributions show that more turbulence creates larger cells with a wider distribution of normalized surface areas indicating a more intense burning process.

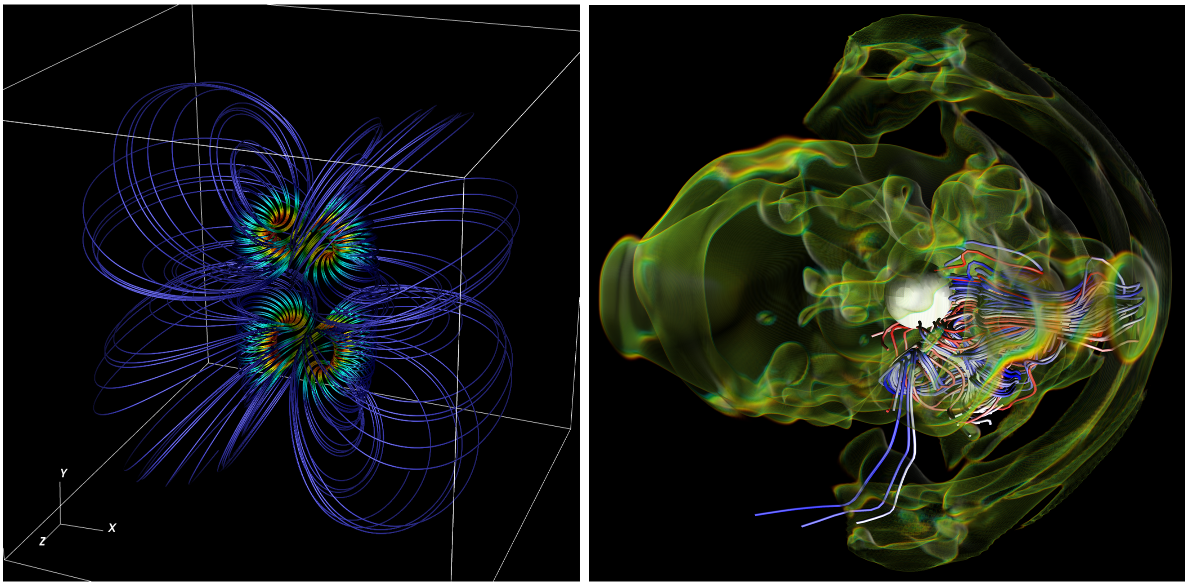


Figure 7. Integral curves illustrate the paths of massless particles in a vector field. The left image is generated from the numerical simulation of vortex merging (simulation courtesy ANAG, LLNL [43]). On the right, streamlines illustrate the magnetic field of a collapsing stellar core (simulation from Endeve et al., ORNL).

5. High Performance, Parallel, and Accurate Computation of Particle Trajectories, Integral Curves and Integral Surfaces

5.1. Introduction: Integration-Based Visualization

Modern methods for the analysis and visualization of vector fields, which play a key role in many application domains like flow simulation, astrophysics and fusion, are built on empirically studying the behavior of massless particles advected with the vector field. These so-called integration-based or Lagrangian methods offer unparalleled insight into vector field structures especially for time dependent vector fields [42].

One simple yet effective visualization approach for vector fields is the straightforward depiction of particle trajectories, as described by vector field integral curves. For a vector field $v(t, x)$, an integral curve started at a point x_0 at time t_0 is defined by the ordinary differential equation

$$\dot{\phi}(t) = v(t, \phi(t)), \quad \text{with initial condition } \phi(t_0) = x_0,$$

and is readily approximated using numerical integration schemes. The resulting curves can then be directly depicted graphically to allow an identification of general vector field behavior (see Figure 7).

5.2. Integral Surfaces

Recently, VACET scientists have developed a novel approach of grouping trajectories of particles emanating from a common curve seeding into so-called integral surfacess. Technically, an integral surface S corresponding to a seeding curve C a is defined by

$$\frac{\partial}{\partial t} S(s, t) = v(t, S(s, t)) \quad \text{where } S(s, t_0) = C(s).$$

The resulting visualizations are a significant improvement over visualizing individual curves due to the fact they offer the possibility for greatly improved visual understanding of complex 3D flow (see Figure 8).

To approximate integral surfaces, the continuum of integral curves over s is reduced to a finite number of such curves that represent a skeleton of the integral surface. However, due to the strong deformation by the flow a fixed discretization does not result in a good approximation of the surface; rather, the newly developed algorithm adapts the discretization to the observed distortion of the surface as it progresses through the vector field [44, 45]. This allows the generation of high-quality surfaces exhibiting fine detail using a vastly reduced number of integral curve computations in comparison to a naive approach. Moreover, care is taken to preserve temporal locality during the surface propagation, permitting computation of surfaces in large time-dependent vector fields in an out-of-core, streaming fashion.

5.3. Lagrangian Coherent Structures

The recently introduced notion of Lagrangian Coherent Structures (LCS) and the associated Finite-Time Lyapunov Exponents (FTLE) measure (cf. e.g. [46, 47]) have garnered much attention in both visualization and application science communities for their ability to visualize and analyze time-varying vector fields in terms of the convergence and divergence of neighboring particles. FTLEs are a finite-time analogue of Lyapunov exponents in dynamical systems; they measure the average exponential separation rate between neighboring particles over a time interval Δt via

$$\sigma := \frac{1}{|\Delta t|} \log \|D\phi\|_2,$$

where $\|D\phi\|_2$ denotes the spectral norm of the spatial derivative of the particle advection over Δt. LCS are then identified as ridge lines in the resulting scalar measure σ. It can be shown that these LCS represent particle trajectories of maximal hyperbolicity and hence form a “dynamic skeleton” of a vector field. Through these structures, it is possible to accurately visualize the fully dynamic nature of vector fields, as exemplified in Figure 9. In contrast, earlier techniques such as topological analysis are unable to easily capture time-varying structures [48]. Furthermore, FTLEs enable the representation of the structural dynamics in terms of scalar fields, opening up vector field structural analysis to a host of existing methods developed in the context of scalar fields. VACET scientists have delivered visualization tools that enable fast computation and interactive visualization of vector fields using these techniques. Furthermore, they continue to examine the value of FTLEs and closely related measures can be generalized to other applications domains beyond fluid flow such as dynamical systems and medical visualization (see Figure 10).

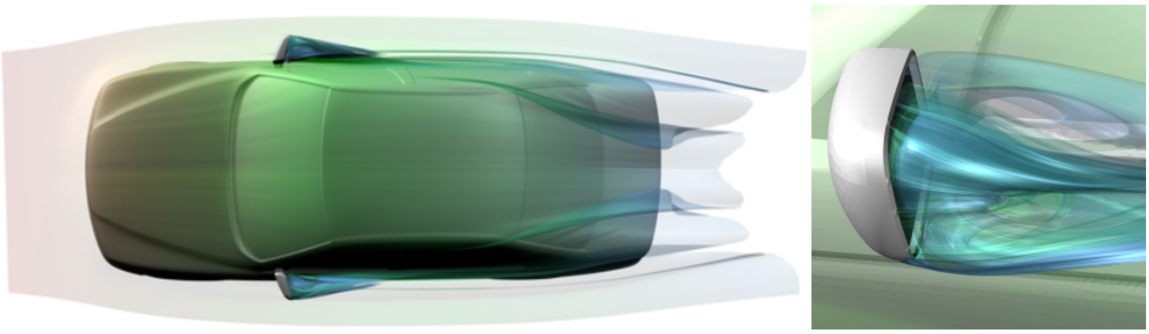
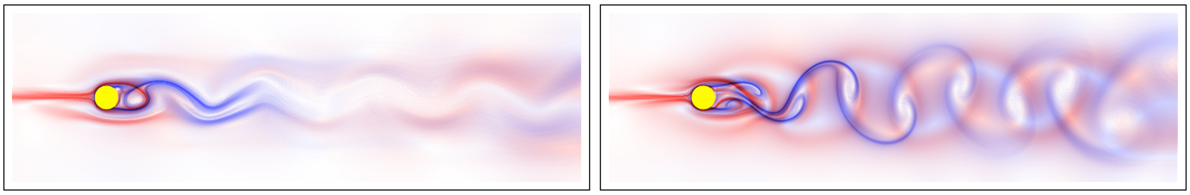


Figure 8. A path surface illustrates the flow of air around a car. The surface represents a dense sheet of massless particles, and the surface color coding provides insight into the lifetime of individual particles. The most notable features identified in this visualization are the vortices behind the rear-view mirror, where the particles are drawn into a vortex (close-up on the right). The path surface is constructed from several thousand integral curves (simulation courtesy M. Rütten, German Aerospace Research Center).



instantaneous	time-dependent

Figure 9. The two-dimensional flow behind a cylinder is visualized using a direct color coding of FTLE measures for forward (red) and backward (blue) advection. The left image shows an analysis of the instantaneous flow, whereas the right image is computed from the fully time-dependent flow field. In comparison, it is apparent that the instantaneous depiction does not adequately reveal the flow dynamics. The time-varying FTLE analysis reveals strongly localized ridge-like LCS that capture the characteristic vortex street downstream of the cylinder (cf. [49]).

Unfortunately, the computation of FTLE measures entails computing a huge number of particle traces that densely cover the domain of interest. Even for medium-sized datasets, it is not uncommon that application of Lagrangian visualization techniques requires computing millions of trace particles. VACET researchers have proposed a variety of methods to address this problem. For smaller data, they leveraged the massive parallel nature of GPUs to allows a direct FTLE computation [49] and corresponding visualization with a 30× to $50x$ speedup compared to a CPU implementation, resulting in interactive computation and visualization. For large to very large data, an approach was presented that evaluates FTLEs adaptively by varying the density of seeded particles in dependence of the spatial variation of the FTLE measure. Depending on the vector field, this can reduce the number of required integral curve computations to between 5% and 20% of the naive, full implementation [51].

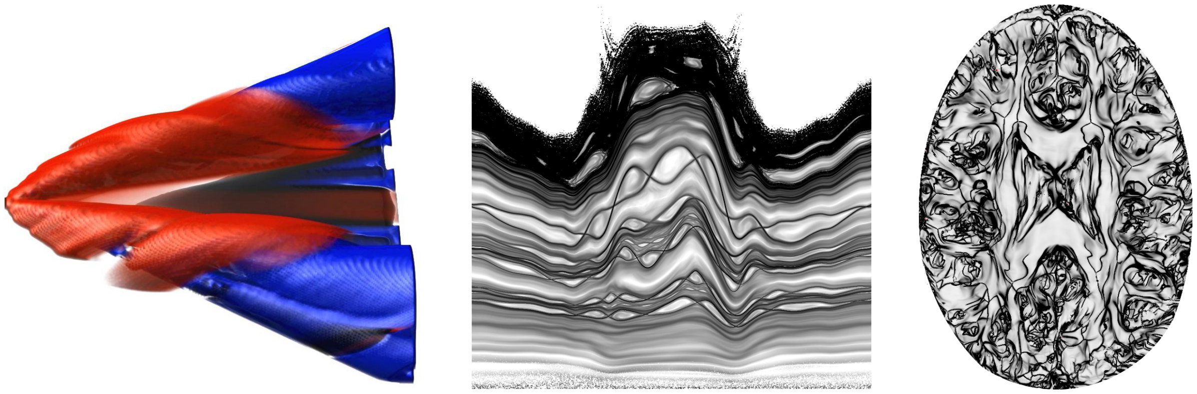


Figure 10. Lagrangian Visualization using FTLE measures is applicable over a wide variety of application domains. The right image visualizes the LCS over a Delta wing using a direct volume rendering of FTLE measures for forward (red) and backward (blue) advection (simulation courtesy M. Rütten, German Aerospace Research Center). The center image depicts an FTLE-like measure that captures coherent structures in a Poincaré section from a Tokamak fusion simulation (data from NIMROD project [50]). The rightmost image depicts a FTLE-like coherency measure for fiber bundles in medical DT-MRI data.

5.4. High-Performance, Scalable Integral Curves

For all integration-based visualization techniques, very large datasets represent a significant challenge. The non-local, non-linear nature of integral curve computation defies established parallelization paradigms and requires novel approaches to distribution of both data and computation. Furthermore, the sheer number of integral curves that must be computed to apply these techniques poses a significant challenge. Even though VACET researchers have successfully worked to reduce this number using adaptive approximation and other techniques, the computational effort required to apply Lagrangian methods to very large, petascale datasets is still significant. This problem will be addressed by the development of novel parallelization techniques that will allow integral curve computation to scale to large supercomputers. These new capabilities, together with the Lagrangian visualization techniques that are build on them, will be implemented in production-quality, parallel-capable visual data exploration software that runs on virtually all modern platforms, ranging from desktop-class machines to DOEs petascale computer systems. This work represent a major new capability for domain scientists concerned with vector field analysis, as it will enable broad use of modern visualization methods and allow the treatment of petascale datasets when run on large parallel computing platforms.

Acknowledgments

This work was supported by the Director, Office of Science, Office and Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET). This research used resources of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References

[1] 2007 Visualization and Knowledge Discovery: Report from the DOE/ASCR Workshop on Visual Analysis and Exploration at the Extreme Scale `http://www.sci.utah.edu/vaw2007/`

[2] Mount R 2004 The office of science data-management challenge. report from the doe office of science data-management workshops Tech. Rep. SLAC-R-782 Stanford Linear Accelerator Center URL `http://www.slac.stanford.edu/cgi-wrap/getdoc/slac-r-782.pdf`

[3] Butler D M, Almond J C, Bergeron R D, Brodlie K W and Haber R B 1993 Visualization Reference Models *VIS '93: Proceedings of the 4th conference on Visualization '93* pp 337–342 ISBN 0-8186-3940-7 (PAPER)

[4] Childs H, Brugger E S, Bonnell K S, Meredith J S, Miller M, Whitlock B J and Max N 2005 A contract-based system for large data visualization *Proceedings of IEEE Visualization 2005* pp 190–198

[5] Childs H, Duchaineau M A and Ma K L 2006 A scalable, hybrid scheme for volume rendering massive data sets *Eurographics Symposium on Parallel Graphics and Visualization* pp 153–162

[6] Rübel O, Prabhat, Wu K, Childs H, Meredith J, Geddes C G R, Cormier-Michel E, Ahern S, weber G H, Messmer P, Hagen H, Hamann B and Bethel E W 2008 High performance multivariate visual data exploration for extemely large data *SuperComputing 2008 (SC08)* (Austin, Texas, USA) lBNL-716E (to appear)

[7] Stockinger K, Shalf J, Wu K and Bethel E W 2005 Query-Driven Visualization of Large Data Sets *Proceedings of IEEE Visualization 2005* (IEEE Computer Society Press) pp 167–174 lBNL-57511

[8] Nieter C and Cary J R 2004 *J. Comput. Phys.* **196** 448–473 ISSN 0021-9991

[9] Geddes C, Toth C, van Tilborg J, Esarey E, Schroeder C, Bruhwiler D, Nieter C, Cary J and Leemans W 2004 *Nature* **438** 538–541 lBNL-55732

[10] Inselberg A and Dimsdale B 1990 Parallel coordinates: a tool for visualizing multi-dimensional geometry *VIS '90: Proceedings of the 1st conference on Visualization '90* (Los Alamitos, CA, USA: IEEE Computer Society Press) pp 361–378 ISBN 0-8186-2083-8

[11] Inselberg A 2008 *Parallel Coordinates Visual Multidimensional Geometry and Its Applications* (Springer-Verlag)

[12] Inselberg A 1984 Parallel coordinates for multidimensional displays *Spatial Information Technologies*

for Remote Sensing Today and Tomorrow, The Ninth William T. Pecora Memorial Remote Sensing Symposium (IEEE Computer Society Press) pp 312–324

[13] Inselberg A, Hauser H, Ward M and Yang L 2006 Modern parallel coordinates: from relational information to clear patterns, tutorial *IEEE Visualization*

[14] Inselberg A 2009 *Parallel Coordinates: Interactive Visualization for High Dimensions* (Springer-Verlag) Advanced Information and Knowledge Processing

[15] Rübel O, Weber G H, Keränen S V E, Fowlkes C C, Hendriks C L L, Simirenko L, Shah N Y, Eisen M, Biggin M D, Hagen H, Sudar J D, Malik J, Knowles D and Hamann B 2006 Pointcloudxplore: Visual analysis of 3d gene expression data using physical views and parallel coordinates *Data Visualization 2006 (Proceedings of EuroVis 2006)* ed Santos B S, Ertl T and Joy K (Aire-la-Ville, Switzerland: Eurographics Association) pp 203–210

[16] Rübel O, Weber G, Keränen S V E, Fowlkes C C, Hendriks C L L, Simirenko L, Shah N Y, Eisen M B, Biggin M D, Hagen H, Sudar D, Malik J, Knowles D W and Hamann B 2006 *PointCloudXplore: A Visualization Tool for 3D Gene Expression Data* (*GI Lecture Notes in Informatics* vol S-4) (Bonn, Germany: Gesellschaft fuer Informatik (GI)) pp 107–117 lBNL-62336

[17] Rübel O, Weber G H, Huang M Y, Bethel E W, Biggin M D, Fowlkes C C, Hendriks C L, Keränen S V E, Eisen M B, Knowles D W, Malik J, Hagen H and Hamann B 2008 *IEEE Transactions on Computational Biology and Bioinformatics* LBNL-382E, to appear

[18] Jones C, Ma K L, Sanderson A and Myers L 2007 *Journal of Physics, Conference Series, Proceedings of SciDAC 2007* **78**

[19] Novotný M and Hauser H 2006 *IEEE Transactions on Visualization and Computer Graphics* **12** 893–900

[20] Stockinger K, Bethel E W, Campbell S, Dart E and Wu K 2006 Detecting Distributed Scans Using High-Performance Query-Driven Visualization *SC '06: Proceedings of the 2006 ACM/IEEE Conference on High Performance Computing, Networking, Storage and Analysis* (IEEE Computer Society Press)

[21] Ushizima D, Rübel O, Prabhat, Weber G, Bethel E W, Aragon C, Geddes C, Cormier-Michel E, Hamann B, Messmer P and Hagen H 2008 Automated Analysis for Detecting Beams in Laser Wakefield Simulations *2008 Seventh International Conference on Machine Learning and Applications, Proceedings of IEEE ICMLA'08* lBNL-960E

[22] Bremer P T, Bringa E M, Duchaineau M A, Gyulassy A G, Laney D, Mascarenhas A and Pascucci V 2007 *Journal of Physics: Conference Series (Proceedings of SciDAC 2007)* **78** doi:10.1088/1742-6596/78/1/012007

[23] Lorensen W E and Cline H E 1987 *Computer Graphics (Proceedings of ACM SIGGRAPH 87)* **21** 163–169 ISSN 0097-8930

[24] Nielson G M 2003 *IEEE Transactions on Visualization and Computer Graphics* **9** 341–351

[25] Milnor J W 1963 *Morse Theory* (Princeton, New Jersey: Princeton University Press) ISBN 0691080089

[26] Carr H, Snoeyink J and van de Panne M 2004 Simplifying flexible isosurfaces using local geometric measures *IEEE Visualization 2004* (IEEE) pp 497–504

[27] Gyulassy A, Natarajan V, Pascucci V, Bremer P T and Hamann B 2005 Topology-based simplification for feature extraction from 3D scalar fields *IEEE Visualization 2005* pp 535–542

[28] Reeb G 1946 *Comptes Rendus de l'Acadèmie des Sciences de Paris* **222** 847–849

[29] Boyell R L and Ruston H 1963 Hybrid techniques for real-time radar simulation *Proceedings of the 1963 Fall Joint Computer Conference* (IEEE) pp 445–458

[30] Edelsbrunner H, Harer J and Zomorodian A 2003 *Discrete & Computational Geometry* **30** 87–107

[31] Edelsbrunner H, Harer J, Natarajan V and Pascucci V 2003 Morse-smale complexes for piecewise linear 3-manifolds *Proceedings of the 19th ACM Symposium on Computational Geometry* pp 361–370

[32] Bremer P T, Edelsbrunner H, Hamann B and Pascucci V 2004 *IEEE Transactions Visualization and Computer Graphics* **10** 385–396

[33] Pascucci V, Cole-McLaughlin K and Scorzelli G 2005 Multi-resolution computation and presentation of contour trees Tech. Rep. UCRL-PROC-208680 Lawrence Livermore National Laboratory preliminary version appeared in the proceedings of the IASTED conference on Visualization, Imaging, and Image Processing (VIIP 2004), 2004, pp.452-290.

[34] Weber G H, Bremer P T and Pascucci V November/December 2007 *IEEE Transactions on Computer Graphics, Proceedings of Visualization 2007* **13** 1416–1423 ISSN 1077-2626 lBNL-63763

[35] Laney D, Bremer P T, Macarenhas A, Miller P and Pascucci V 2006 *IEEE Transcations on Visualization and Computer Graphics* **12** 1053–1060

[36] Gyulassy A, Natarajan V, Duchaineau M, Pascucci V, Bringa E M, Higginbotham A and Hamann B November/December 2007 *IEEE Transactions on Computer Graphics, Proceedings of Visualization 2007* **13**

[37] Edelsbrunner H, Harer J, Natarajan V, and Pascucci V 2004 Local and global comparison of continuous

functions *Proceedings of IEEE Visualization 2004* pp 275–280

[38] Day M S, Bell J B, Bremer P T, Pascucci V and Vince Beckner M L 2009 *Combustion and Flame* **156** 1035–1045 doi:10.1016/j.combustflame.2008.10.029

[39] Day M S and Bell J B 2000 *Combust. Theory Modelling* **4** 535–556

[40] Bell J B, Cheng R K, Day M S and Shepherd I G 2007 *Proc. Combust. Inst.* **31** 1309–1317

[41] Bremer P T, Weber G H, Pascucci V, Day M S and Bell J B 2009 *Transactions on Visualization and Computer Graphics* Accepted with minor revisions

[42] McLoughlin T, Laramee R S, Peikert R, Post F H and Chen M 2009 Over Two Decades of Integration-Based, Geometric Flow Visualization *Eurographics STAR - State of The Art Report* ed Pauly M and Greiner G pp 73–92

[43] Martin D F, Colella P and Graves D 2008 *Journal of Computational Physics* **227** 1863–1886

[44] Garth C, Tricoche X, Salzbrunn T and Scheuermann G 2004 Surface techniques for vortex visualization *Proc. Eurographics - IEEE TCVG Symp. on Visualization*

[45] Garth C, Krishnan H, Tricoche X, Tricoche T and Joy K I 2008 *IEEE TVCG* **14** 1404–1411 ISSN 1077-2626

[46] Green M, Rowley C and Haller G 2006 *J. Fluid Mech.* **to appear**

[47] Lekien F, Shadden S C and Marsden J E 2007 *Journal of Mathematical Physics* **48**

[48] Sadlo F and Peikert R 2008 Visualizing Lagrangian Coherent Structures and Comparison to Vector Field Topology *Topology-Based Methods in Visualization II* ed Hege H C, Polthier K and Scheuermann G (Springer) pp 15–30 ISBN 978-3-540-88605-1

[49] Garth C, Li G S, Tricoche X, Hansen C D and Hagen H 2007 Visualization of coherent structures in transient 2d flows *Topology-Based Methods in Visualization 2007* Mathematics + Visualization (Springer)

[50] Sovinec C, Glasser A, Gianakon T, Barnes D, Nebel R, Kruger S, Plimpton S, Tarditi A, Chu M and the NIMROD Team 2004 *J. Comp. Phys.* **195** 355

[51] Garth C, Gerhardt F, Tricoche X and Hagen H 2007 *IEEE Transactions on Visualization and Computer Graphics* **13** 1464–1471 ISSN 1077-2626