Scalable Lagrangian-based Attribute Space Projection for Multivariate Unsteady Flow Data

Hanqi Guo [1,2] Fan Hong [1] Qingya Shu [1] Jiang Zhang [1,2] * Jian Huang [3] † Xiaoru Yuan [1,2] ‡

1) Key Laboratory of Machine Perception (Ministry of Education), and School of EECS, Peking University
2) Center for Computational Science and Engineering, Peking University
3) Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville

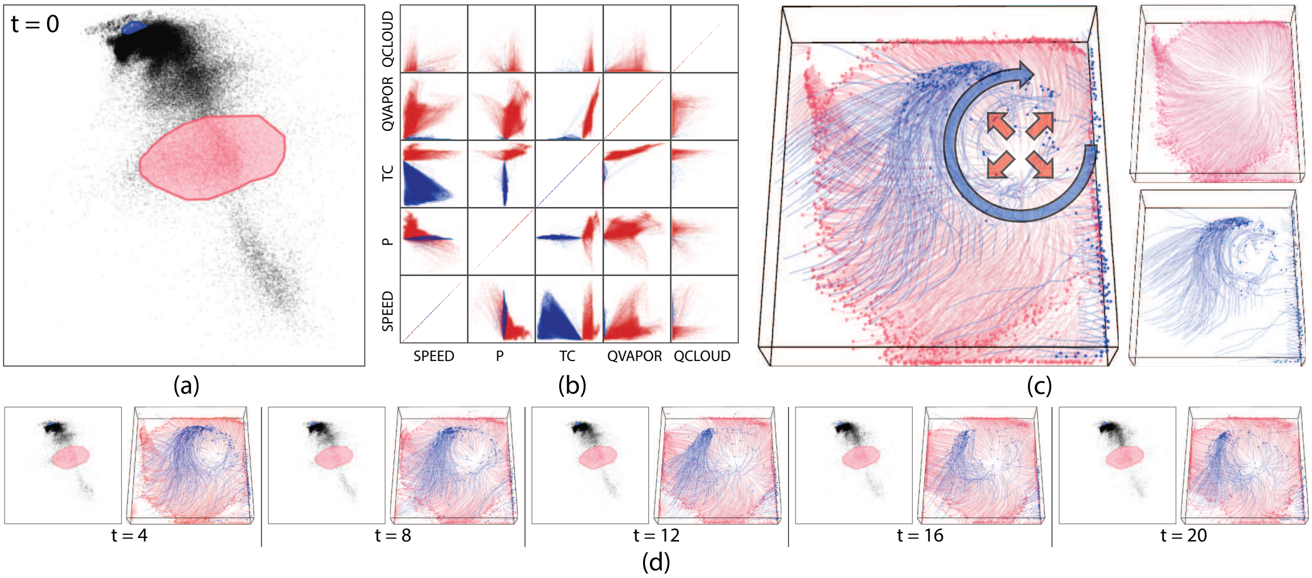


Figure 1: The visualization results for Hurricane Isabel dataset. (a)(b)(c) show results at time step 0. (a) On the projection view, two groups of points are selected. Each point represents the pathline starting from a spatiotemporal point. (b) In the attribute matrix, the selected pathlines are projected in individual scatterplots in the matrix. Each selected group in (a) appears to be clustered in attribute space. (c) In the spatial view, the spatial distribution of the corresponding pathlines are visualized. (d) shows the projection view and spatial view of corresponding pathlines at time step 4, 8, 12, 16, and 20.

Abstract

In this paper, we present a novel scalable approach for visualizing multivariate unsteady flow data with Lagrangian-based Attribute Space Projection (LASP). The distances between spatiotemporal samples are evaluated by their attribute values along the advection directions in the flow field. The massive samples are then projected into 2D screen space for feature identification and selection. A hybrid parallel system, which tightly integrates a MapReduce-style particle tracer with a scalable algorithm for the projection, is designed to support the large scale analysis. Results show that the proposed methods and system are capable of visualizing features in the unsteady flow, which couples multivariate analysis of vector and scalar attributes with projection.

Keywords: Flow visualization, attribute space projection, parallel processing

1 Introduction

Visualization and analysis on flow field data has been long studied in the community, such as texture-based [19] and geometry-based [21] flow visualization. However, it is still challenging to visualize the insights into large scale unsteady flow data. For example, the visual clutter problem often exists in geometry-based flow visualization, which requires careful seed placement strategies. The same problem also exists in texture-based 3D flow visualization.

Alternative to traditional flow visualization methods, interactive feature selection techniques have been developed to help users to identify and extract interesting features in the flow field. Among them, projection methods can help users to identify and select features in both projection space and spatial space. For example, sample values in 2D flow are mapped into 2D space, thus important spatial structures can be extracted interactively on the projection space [7]. Streamlines can also be projected according to their geometry distances for visual exploration [28].

However, there are certain problems in the current practice on the projection of flow field data, especially for unsteady flow dataset with multivariate scalar fields. Traditional projection techniques, which are originally proposed in the light of multivariate data, do not sufficiently take the flow advection into account. For example, if l^2-norm is used to evaluate the distance of two velocities, it is equivalent to the differences of velocity magnitude, thus the directional information is totally lost. Even if feature descriptors are applied to compute the distances, e.g. λ_2 or vorticity magnitude, the evolution of the unsteady flow is still ignored, as the particles advect in the field.

In this work, we propose Lagrangian-based Attribute Space Projection (LASP), which tightly couples multivariate analysis and

*e-mail: {hanqi.guo,fan.hong,qingya.shu,jiang.zhang}@pku.edu.cn
†e-mail: huangj@eecs.utk.edu
‡e-mail: xiaoru.yuan@pku.edu.cn

flow advection. A parallel system is also developed to support large scale analysis. With LASP, sample points in unsteady flow are embedded into a lower dimensional space using Multi-Dimensional Scaling (MDS) with the Lagrangian-based distance metric. The metric, not only accounts for the multivariate attributes on the current location, but also accumulates the attribute values along the particle advection locations. Formally, we use Lagrangian instead of Eulerian specification to compute the distances. The Pivot MDS algorithm [2], which is with low complexity, is used to project the samples, and we further develop the out-of-sample extension to Pivot MDS for efficient multiresolution analysis of large scale data.

Studying flow fields with Lagrangian specification in visualization has been proved as a success in previous research [12, 29]. However, it is challenging to use Lagrangian specification in flow field analysis, because it is a both data- and task-intensive problem. The computation power of supercomputers is essential for such analysis on large scale unsteady flow. In our system design, two major routines are parallelized to support large scale data analysis, including the particle tracing and the projection. For the massive particle tracing, a MapReduce-like framework called DStep [17] is used. It is the most scalable solution for particle tracing up to date. For the massive projection, the scalable Pivot MDS [11] is used to accelerate the projection. We also develop the streaming extension to the existing Pivot MDS algorithm to decrease computation cost for multiresolution and progressive analysis. However, due to the MapReduce-like architecture design of DStep, the incorporation of the Pivot MDS is not straightforward. In the system, the two-pass procedures are designed to compute pathlines for Pivot MDS, and the projection is done in the reduce stage in parallel. More details are provided in following sections.

The contributions of this paper are three-fold: 1) Lagrangian-based attribute space projection featuring a new distance metric; 2) Out-of-sample extension for multiresolution analysis; 3) Scalable and parallel implementation.

In the remainder of the paper, we describe the background in Section 2. In Section 3, Lagrangian-based attribute space projection is introduced, and the parallel algorithms and system design are described in Section 4. Results are shown in Section 5, and conclusions are drawn in Section 6.

2 Background

Visualization of flow field data is challenging, and it has been studied for decades. In general, typical rendering techniques for flow field data include texture-based [19] and geometry-based [21] methods. The goal of our work better aligns with flow feature extraction and tracking [26]. In addition to the traditional flow visualization techniques, our method is more related to multidimensional projection techniques, Lagrangian and Eulerian flow analysis, and parallel particle tracing problems.

2.1 Multidimensional Projection Techniques

Multidimensional Projection techniques, which are widely used in various visualization applications, map a group of data instance into lower dimensions for data analysis. Multi-Dimensional Scaling (MDS) techniques are the most commonly used projection methods. They transform multidimensional data elements into lower and explorable dimension space according to the mutual distances. There are mainly three types of MDS techniques, namely the distance-scaling methods [10], optimization-based methods [18], and eigensolver-based methods. In our work, we focus more on the eignsolver-based techniques, because existing extensions are available to scale to large problem sets in parallel. Torgerson [32] proposed the first eigensolver-based MDS algorithm, classical MDS. The MDS is transformed to the eigensolver problem for the double-centered distance matrix. However, both the computation and storage complexities are often too high for real massive applications. Landmark MDS [8], which is an approximation to classical MDS, significantly reduces the complexities by only computing the eigen-decomposition of a few landmark elements. Pivot MDS [2] further improved the quality of landmark MDS. As the ever growth of the data scale, MDS algorithms are accelerated by parallelism. Parallel MDS algorithms are also accelerated on distributed and parallel systems [34]. In our system, we use Pivot MDS [2] and its parallel extension [11], which is efficient to the huge amount of spatiotemporal samples. Furthermore, the streaming extension is developed for multiresolution and progressive analysis in our approach.

In scientific visualization, multidimensional projection has been widely applied in various scenarios. Users can use brushing techniques to select multivariate features in 2D [14]. For example, multivariate transfer functions can be generated by selecting features on the projection plots [11]. For vector field data, distance metrics are the key to obtaining intriguing projection results [7]. In addition to individual sample points in flow field, it is also meaningful to embed field lines into lower dimensional spaces. Chen et al. [6] proposed a method to explore DTI fibers on 2D MDS based on mean distance, thus avoiding the cluttering problems in 3D space. In streamline embedding [28], the distances between seed points are defined as the Hausdorff distances between the corresponding streamlines. Different from previous study on streamline embedding, pathlines instead of streamlines are traced in unsteady flow, which is much more challenging on the computation and the data management. Furthermore, attribute space distance instead of geometry space distance is used to identify the flow features in Lagrangian perspectives.

2.2 Lagrangian and Eulerian Specifications

In fluid dynamics, there are basically two methods to describe unsteady flow field, namely the Eulerian specification and the Lagrangian specification. Both two specifications are useful in different scenarios. In Finite-Element study, Arbitrary Lagrangian-Eulerian techniques, which combine the benefits of the both specifications, are widely used in engineering simulations [31]. Similar approaches are applied in dense and texture-based flow visualization methods [19, 15]. Our focus is mainly on Lagrangian-based approaches, which incorporate massive computation of field lines. One typical example is Finite-Time Lyapunov Exponent (FTLE) [13], which indicates how particles diverge around the seed locations. Flow feature detection conducted by classifying pathline attributes [29, 30, 9], can also be categorized as Lagrangian analysis. Lagrangian specification is also used to compare the flow fields in ensemble runs in previous research [12]. As the computation cost of using Lagrangian specification is very high, parallelism is often used to accelerate the particle tracing and analysis.

Formally, in Eulerian specification, the attribute values are as the function of spatiotemporal location $\mathbf{x}^t$, where $\mathbf{x}$ is the location and t is the time. For example, the velocity, the temperature, and the pressure in a flow field can be written as $\mathbf{v}(\mathbf{x}^t)$, $T(\mathbf{x}^t)$, and $p(\mathbf{x}^t)$, respectively. On the contrary, in Lagrangian specification, the attributes are associated with the particles in the flow field, which are moving from the spatiotemopral location $(\mathbf{a}^{t_0})$. The Lagrangian specification of the above mentioned attributes are written as $\mathbf{v}(\mathbf{a}^{t_0+t})$, $T(\mathbf{a}^{t_0+t})$, and $p(\mathbf{a}^{t_0+t})$, respectively, where t is the elapsed time. The relationships between the two specifications are as follows:

$$\mathbf{v}(\mathbf{X}^{t_0+t}(\mathbf{a}^{t_0+t})) = \frac{\partial \mathbf{X}(\mathbf{a}^{t_0+t})}{\partial t}, \tag{1}$$

where $\mathbf{X}$ is the displacement. For convenience, the Lagrangian specification can be written as *flow map*, which is an implicit function of spatiotemporal locations $\mathbf{x}_0^{t_0}$ and elapsed time t:

$$\Phi : (\mathbf{x}_0^{t_0}; t) \mapsto \Phi_{t_0}^{t_0+t}(\mathbf{x}) = \mathbf{x}(\mathbf{x}_0^{t_0}; t). \tag{2}$$

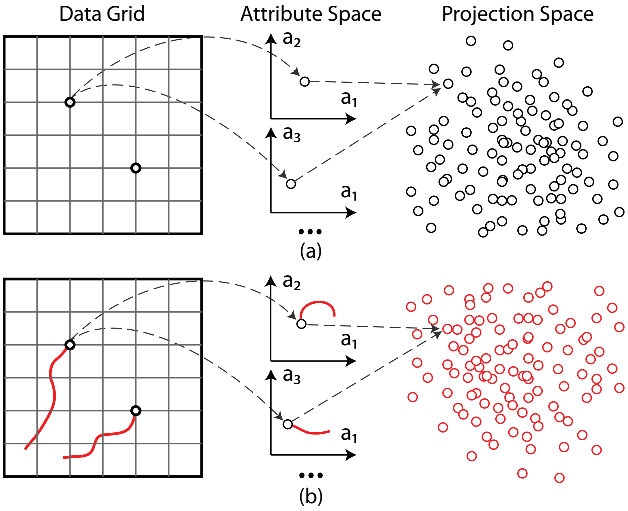


Figure 2: The Eulerian-based (a) and Lagrangian-based (b) attribute space projection for unsteady flow.

In the analysis of the unsteady flow, the data structures of Eulerian and Lagrangian specifications are quite different. Unsteady flow datasets in Eulerian specification are often stored as 3D or 4D time-varying arrays. The Lagrangian specification is stored as a set of pathlines, and each point along the pathlines contains multivariate values from the dataset. It is highly difficult to compute and store the pathlines for analysis. For example, in a global climate simulation, if we record hourly positions and multivariate attributes in the pathline for two-month time scope (usually more than $1,000$ samples if the particle does not go out of the boundary), the intermediate data scale is about 10^3 times larger than the raw flow data, which is prohibitive in practice.

2.3 Parallel Particle Tracing

Analyzing unsteady flow with Lagrangian specification requires massive particle tracing in the field with scalability, which is a fundamental yet challenging problem in high performance visualization. The solution is either data- or task-parallelism. In data-parallel methods, the load-balancing highly relies on the data block distribution. Existing strategies include round-robin [25], hierarchical clustering [33], etc. Other strategies include partitioning base on flow features [5], and using flow-guided file layout to improve I/O performance [4]. In task-parallel methods, scheduling is the key to scalability. Studies on both dynamic [27] and static [24] load-balancing are available in existing literature. Hybrid method using on-demand strategy exists to reduce I/O and communication costs for overall performance [3]. Parallel particle tracing also accelerate the computation of FTLE [23].

DStep [17] is a MapReduce-like framework for particle tracing. While MapReduce only handles data-parallel, DStep can manage both data-parallel and task-parallel at the same time efficiently. It is reported that DStep has scaled to 64K cores in BlueGene/P. Guo et al. [12] proposed an improved system based on DStep to extract features as differences in ensemble flow data. Our system further extends the design for the applications, fully embraces the parallel Pivot MDS algorithm in the MapReduce-like workflow. More details and design rationales are provided in the following sections.

3 Our Method

In this work, we propose LASP for the coupled multivariate analysis and flow advection. As shown in Figure 2, the essential difference to traditional projection technique is the use of Lagrangian specification for distance computing. We provide three different views for the visual analysis on unsteady flow data with LASP, including the projection view, the attribute matrix, and the spatial view. The attribute matrix contains the scatterplots which presents the projection of pathlines in the attribute spaces. Users can identify and select features in the projection view, and observe both the attribute and the spatial distribution in the other two views.

Our method is capable of and necessary to scale with parallelism. First, the unsteady flow data is often overwhelmingly large without parallelism. Second, the massive pathlines can be traced in parallel. Third, the parallel MDS projection is also necessary.

3.1 Overview of LASP

The LASP method transforms the spatiotemporal samples in unsteady flow into a 2D screen space for feature identification and selection. The projection, in which similar samples are positioned saliently, is based on the Lagrangian-based distance metric. The new distance metric not only considers the multivariate attribute values on the fixed spatiotemporal positions, but also accounts the sample values on the flow trajectories that are traced from the above positions. Thus, we tightly couple the multivariate analysis of both flow advection and scalar attributes with projection.

As the amount of spatiotemporal samples is huge, we adopt Pivot MDS to reduce the complexity. Compared to traditional methods like classical MDS, it is not necessary to compute the distances between all spatiotemporal samples ($O(n^2)$), which is prohibitive in our framework. Notice that the distance computing is even more challenging than Eulerian-based method, because Lagrangian-based metric requires both tracing and comparison of pathlines. With Pivot MDS, we only need to compute the distances between all samples with a small set of randomly selected ones, so called pivot elements or pivots. Thus, the complexity is reduced to $O(kn)$, where k is the number of pivots. We further develop the out-of-sample extension to Pivot MDS, to support progressive and multiresolution analysis.

The logical pipeline of the projection is shown in Figure 3. From the raw data, the pathlines are traced from all spatiotemporal sample points. The distances between every pathline pairs are computed according to Lagrangian-based metric, then the projection plot is created by MDS techniques. Although the logical pipeline appears to be straightforward, it is challenging to achieve the goal in efficient and scalable way. First, the massive tracing of pathlines is very costly in I/O bandwidth, computation, and memory use. Second, the distance computation and projection is also costly. Although Pivot MDS with out-of-sample extension has largely reduced the complexity, the overall complexity is still too high for real applications in a serial manner. Further design on the parallel and scalable system is needed.

3.2 Distance Metric

The essence of LASP is to measure and show the distance between the spatiotemporal samples with the Lagrangian specification. The illustration of the Lagrangian-based distance metric is shown in Figure 2. The attribute values are collected along the movement of the particle, instead of only one sample.

Without the loss of generality, the distance between $\mathbf{x}^t$ and $\mathbf{x}'^{t'}$ is defined as follows:

$$d^2(\mathbf{x}^t,\mathbf{x}'^{t'};t_c) = \int_0^{t_c} \sum_k \omega_i^2 ||\mathbf{A}_k(\Phi_{t_0}^{t_0+\tau}(\mathbf{x})) - \mathbf{A}_k(\Phi_{t'_0}^{t'_0+\tau}(\mathbf{x}'))||^2 d\tau, \tag{3}$$

where Φ is the flow map, and ω_k is the weight for the attribute $\mathbf{A}_k$. t_c is the time window size, which defines the time scope for the comparison. Notice that t_c is less than the total time T of the dataset. In discrete form, the distance is written as:

$$d^2(\mathbf{x}^t,\mathbf{x}'^{t'};t_c) = \frac{1}{N}\sum_i^N \sum_k \omega_i^2 ||\mathbf{A}_k(\Phi_{t_0}^{t_0+i\Delta t}(\mathbf{x})) - \mathbf{A}_k(\Phi_{t'_0}^{t'_0+i\Delta t}(\mathbf{x}'))||^2, \tag{4}$$

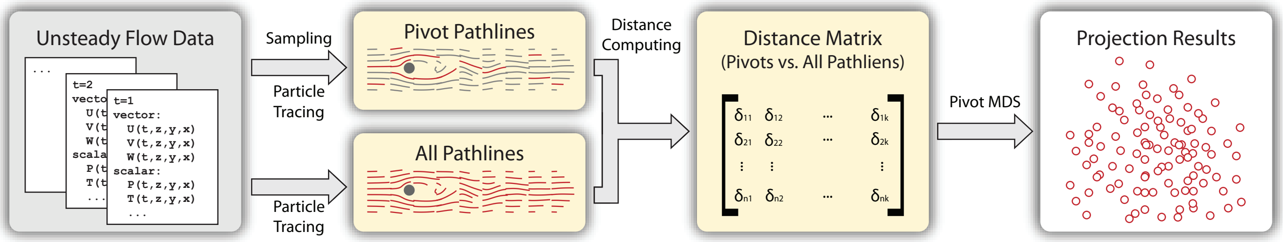


Figure 3: The logical pipeline of Lagrangian-based attribute space projection.

where N is the maximum common number of samples of the two flow maps (pathlines), and Δt is the sample distance on the time dimension. N is implicitly determined by the time window size t_c.

There are several important parameters in the distance metric, including the attribute weights ω_i and the time window size t_c, etc. The impact on the distance from different attributes can be weighted more or less by tuning ω_i. For example, in the atmospheric simulation, users can set different weights to the attributes like pressure, temperature, etc., or completely remove certain attributes. t_c is important to investigate the distances in different time scales for various purposes. If t_c is set to zero, then the distance metric is degraded to an Eulerian-based one.

3.3 Out-of-Sample Extension to Pivot MDS for Multiresolution Analysis

The Pivot MDS, which is with low complexity, is used for embedding massive spatiotemporal samples by transforming the distances into 2D screen space. In addition, we develop an out-of-sample extension to Pivot MDS, thus users can progressively refine the granularity of analysis by changing the resolution coherently. Similar extensions to other projection techniques were studied in previous research [1]. New samples can be directly added into current results without changing the original layout, as shown in Figure 4.

Without loss of generality, the scaling process is based on the distances between every sample pairs in the space. In the projection result, the distances between points are approximately preserved by minimizing the error:

$$\min \sum_{i<j} (||\mathbf{p}_i - \mathbf{p}_j|| - \delta_{ij})^2, \qquad (5)$$

where $\mathbf{p}_i$ are the coordinates of the sample in the projection result, and $\delta_{ij} = d(\mathbf{x}_i^{t_i}, \mathbf{x}_j^{t_j}; t_c)$ is the distance between the two samples $\mathbf{x}_i^{t_i}$ and $\mathbf{x}_j^{t_j}$. With eigensolver-based methods, the optimal layout can be obtained by solving the largest eigenvectors of the double-centered distance matrix $\mathbf{B}$. Pivot MDS only requires the $n \times k$ sized squared distance matrix $\Delta^{(2)}$ instead of a complete $n \times n$ matrix, thus the complexity is reduced. The elements in the double-centered $n \times k$ matrix $\mathbf{C}$ are [2]:

$$c_{ij} = -\frac{1}{2}(\delta_{ij}^2 - \frac{1}{n}\sum_{r=1}^{n} \delta_{rj}^2 - \frac{1}{k}\sum_{s=1}^{k} \delta_{is}^2 + \frac{1}{nk}\sum_{r=1}^{n}\sum_{s=1}^{k} \delta_{rs}^2), \qquad (6)$$

and then Pivot MDS evaluates the eigenvalues and eigenvector of $\mathbf{C}^T\mathbf{C}$ instead of $\mathbf{B}$, thus greatly reduces the complexity of the eigensolver-based methods. The final coordinates $\mathbf{p}$ are obtained by multiplying $\Delta^{(2)}$ and the two largest eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

The out-of-sample problem of Pivot MDS can be described as follows. Given n' more spatiotemporal samples in addition to the existing n ones, compute the projection results of $\mathbf{p}$ and $\mathbf{p}'$. In our application, when the user increases the resolution for the analysis, new spatiotemporal samples need to be added into the projection results. Of course, this problem can be solved by recomputing the MDS result for $n+n'$ samples. The recomputing of the MDS requires recalculating the double-centered matrix $\mathbf{C}'$, the eigenvalues and eigen vectors of $\mathbf{C}'^T\mathbf{C}'$, which is obviously not efficient. Another problem is the coherency. Because the largest eigenvectors $\mathbf{v}_1'$ and $\mathbf{v}_2'$ are likely to change, original results $\mathbf{p}$ also differ from the original positions. We assume that the newly inserted samples in the multiresolution exploration are very similar to the original ones, so the inner products of $\mathbf{C}'$ are proportional to the original ones:

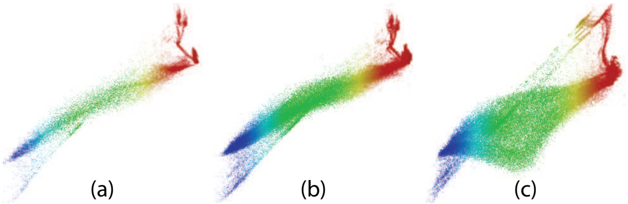


Figure 4: The out-of-sample extension to Pivot MDS for multiresolution analysis: (a) low resolution (34,560 samples), (b) high resolution, out-of-sample extension (138,240 samples), (c) high resolution, direct projection (the same number of samples as in (b)).

$$\mathbf{C}'^T\mathbf{C}' \approx \frac{n+n'}{n}\mathbf{C}^T\mathbf{C}. \qquad (7)$$

As the eigenvalues and eigenvectors are scale-invariant, we assume:

$$\mathbf{v}_1' \approx \mathbf{v}_1, \mathbf{v}_2' \approx \mathbf{v}_2. \qquad (8)$$

Thus, the new samples can be inserted into the projection results while keeping the original positions $\mathbf{p}$. The approximation by Eq. 8 can be interpreted as a “low-pass filter”, if the characteristics of new inserted samples differ. Figure 4 shows an example of multiresolution analysis on GEOS-5 simulation data. The sample distance in (a) is $(12,6,18,4)$ in the 4 spatiotemporal dimensions, and it is $(6,6,9,4)$ in (b) and (c), thus the samples in (a) are in a subset of the samples in (b) and (c). The out-of-sample extension provides consistent results in (b) as user increases the resolution for the analysis.

4 System Design with Scalability

The design goal of the parallel system is to fully incorporate particle tracing with dimension projection in a scalable manner. The reason for the integration is due to the volume of intermediate pathlines, which is often much larger than the raw data, as we discussed in Section 2.2. If the particle tracing and the projection are performed independently, the intermediate data is prohibitive for file systems to store in most cases.

The workflow of the parallel system is shown as Figure 5. We use the modified DStep framework for particle tracing, and the projection is tightly integrated into the framework. The coupling of the DStep and SPMDS is challenging, because MapReduce-like frameworks are quite different to traditional visualization pipelines [22]. Our method requires sharing essential data including the pivot elements, but the DStep design pattern requires “share-nothing” design of the algorithm. Several phases are configured to bypass the restrictions.

4.1 Basics of Scalable Pivot MDS

Scalable Pivot MDS (SPMDS) projects massive multivariate samples into lower dimensions in parallel. Each process computes the

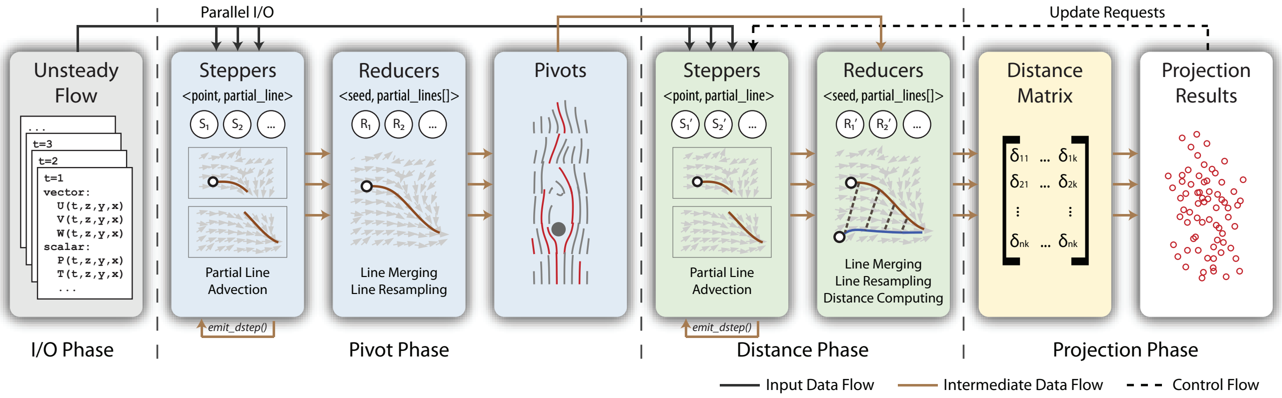
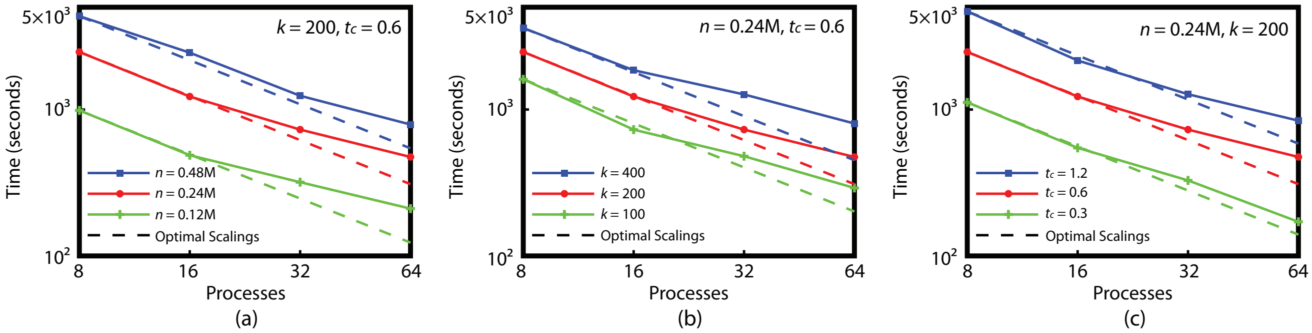


Figure 5: The parallel system design. The scalable Pivot MDS is tightly integrated into DStep framework. The pivot pathlines and all other pathlines are traced in the pivot phase and the distance phase, respectively. The projection is done after the distance matrices are computed.



(a)

(b)

(c)

Figure 6: Timings tested on parallel environment with different numbers of processes. In (a), (b), and (c), three parameters related to problem size, including number of samples n, number of pivots k, and the time window size t_c, are changed respectively.

distributed squared distance matrix $\Delta_p^{(2)}$ at first, where p is the index of the process. The column and row summation of $\Delta^{(2)}$ can be obtained by collective communication for the construction of the distributed double-centered distance matrix $\mathbf{C}_p$. To get the inner product of double centered dissimilarity matrix and its transpose, $\mathbf{C}^T\mathbf{C}$, we first calculate $\mathbf{C}_p{}^T\mathbf{C}_p$ in each process, then use summation reduction among all processes. At last, the first two eigenvectors are extracted for projection.

4.2 Basics of DStep

DStep, which is a MapReduce-like framework, is designed for simplified domain traversal, e.g. field line tracing. DStep hides explicit management of job scheduling and communication in a parallel environment. Developers only need to implement `step()` (`map()` equivalent) and `reduce()` functions with proper key-value pairs. In massive pathline tracing, partial pathlines are traced in the step stage, and they are merged into complete pathlines in the reduce stage. More details about the DStep framework and its memory footprint improvements are documented in [17, 11].

4.3 Incorporating DStep with Scalable Pivot MDS

As we discussed above, there are two major components that need to be parallelized, including the particle tracing and the Pivot MDS. DStep, which is a MapReduce-like framework, brings fine-granularity parallelism for particle tracing. Due to this design, the integration of the both components is not as straightforward as a pipelined model. The MDS projection needs to be tightly integrated into the DStep framework. Because DStep also requires "share-nothing" design, we avoid this restriction and share essential data by setting up several phases.

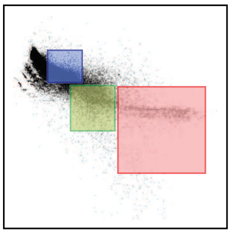
Our design includes 4 phases, namely the I/O phase, the pivot phase, the distance phase, and the projection phase. The sparse pivot pathlines and dense pathlines are seeded at the beginning of the pivot phase and distance phase respectively. The four phases are detailed as follows.

I/O Phase. The domain is partitioned into blocks, which are assigned to each stepper in round-robin order. The blocks are then loaded from the file system at once using BIL library [16].

Pivot Phase. The system randomly picks up k spatiotemporal points as the seeds for the pivot pathlines. The pivot pathlines are then computed in the DStep routines. In the step stage, the partial pathlines are traced. When a pathline goes out of the local block, the partial result is then sent to the reducers and emit a new step job to continue the pathline computation. A pathline is finished when it goes out of the global domain, or the time duration reaches t_c. In the reduce stage, the partial results are merged into complete pathlines, which are then parameterized for further distance computation. When the pivot phase completes, all pivot pathlines are distributed into all reducers for the next phase.

Distance Phase. The distance matrices are computed by comparing all pathlines and pivot pathlines. The pathlines are traced online with DStep and then dumped right after the distance is computed, in order to reduce the memory footprint. The step stage is almost the same as the pivot phase. The reduce stage incorporates the merging and the parameterization of the pathlines, as well as the distance computation. Every new generated pathline is compared with all pivots, and the squared distance is then stored into the distributed squared distance matrix $\Delta_p^{(2)}$. At the end of the distance phase, every reducer keeps the distributed distance matrix $\Delta_p^{(2)}$ for the parallel MDS projection. Because of the nature of MapReduce-like design, the numbers of rows of $\Delta_p^{(2)}$ are almost the same, which ensures the load-balance of the projection phase.

Projection Phase. We follow the scalable Pivot MDS algorithm to obtain the projection results from the distributed squared dis-



Region	#Pathlines	% of all Pathlines	Computing Time (s)
🟥	2,579	2.3%	0.2
🟩	5,485	4.8%	0.7
🟦	18,791	16.4%	3.3
All	114,423	100%	22.0

Table 1: Timings for interactive feature selection on the LASP plot (Isabel dataset, $t = 0$, $t_c = 24$, 32 processes). Three different regions are selected for on-demand pathline computation.

tance matrices $\Delta_p^{(2)}$. All computations are conducted on the processes which contain reducer workers, as $\Delta_p^{(2)}$ is computed by each reducer. First, the distributed squared distance matrix $\Delta_p^{(2)}$ is transformed into the double-centered matrix $\mathbf{C}_p$ in parallel. Second, the inner product matrices $\mathbf{C}_p^T\mathbf{C}_p$ are computed and then reduced into $\mathbf{C}^T\mathbf{C}$. The two eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$ with largest eignenvalues are then solved, and then used to project all elements into 2D.

4.4 Performance

We evaluate the performance and scalability in a parallel environment. The platform is an 8 node PC cluster. Every node is equiped with two Intel Xeon E5520 CPUs which operate at 2.26GHz and with 48GB main memory. The inter-node connection is InfiniBand with 40Gbps theoretical bandwidth.

The benchmark timings of the system with different number of cores and different problem sizes are shown in Figure 6. We tested three parameters related to problem size, that is number of pivots, number of samples, and t_c. Our system maintains good efficiency as the number of processes increasing. Although the full-range analysis seems to be time-consuming, users can either use more computing resources, or reduce the problem size by reducing the sampling rate or enabling the out-of-sample extension.

The feature selection with LASP is interactive. When a user queries samples in the projection plot, the parallel system (server side) recomputes the corresponding pathlines on-demand, which are further send to the spatial view (client side). The feature selection timings (Table 1) are roughly proportional to the number of selected samples. As there are often small portions selected, users only need to wait for a short while to get the query results.

5 Results and Discussion

We applied our system to two dataset, including Hurricane Isabel simulation and GEOS-5 simulation.

5.1 Hurricane Isabel

Hurricane Isabel data is from an atmospheric simulation, which consists 9 scalar variables and the wind field. The spatial resolution of this data set is $500 \times 500 \times 100$, which represents a physical scale of 2,139km×2,004km×19.8km. There are 48 time steps, which are saved per hour during the simulation. The overall size of the data set is about 59 GB.

In this case, the wind speed vector field (U, V, and W) and five scalar variables are considered, namely the wind speed magnitude (SPEED), the pressure (P), the temperature (TC), the water vapor mixing ratio (QVAPOR), and the total cloud moisture mixing ratio (QCLOUD). These attributes are considered as most important attributes for analyzing hurricanes, as suggested by domain experts. We set timescope t_c to 20 hours, so that we can discover clusters in a relative long timescope. The visualization results are shown in Figure 1. Two regions are chosen for our interests. To validate the projection results, we map samples into the attribute matrix. From

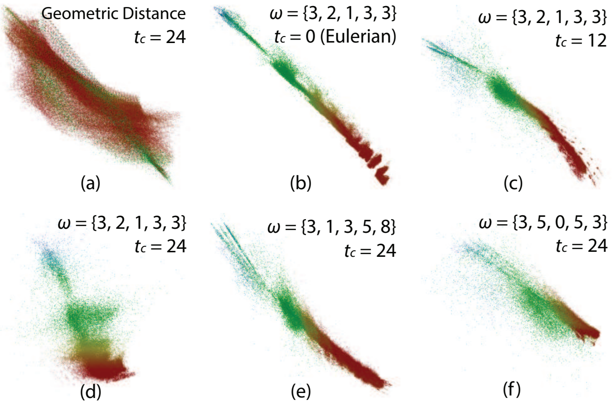


Figure 7: The projection results of Hurricane Isabel dataset with different distance metrics and parameters. Weights of attributes ω and t_c are shown with results except (a) use distance metric in geometric space. Pseudo-color is used to visualize the difference between the projection results.

the attribute matrix, we can observe that these two clusters are bundled by attributes such as TC, QVAPOR, and QCLOUD. At the same time, the two clusters are separated clearly in attribute space, for example by attribute QVAPOR and TC. We then map samples back to pathlines in the spatial view, trying to find their physical meaning. Orange pathlines are traversing from the center of hurricane to the peripheral part near the surface, while blue ones come from peripheral part, then have circular traces around the center at a high latitude. The orange cluster demonstrates transportation process of water vapor from the eye of hurricane to the periphery. At the beginning, orange pathlines convey relatively large amount of water vapor. As particles traverse to peripheral area, water vapor ratio decreases to a relative low level. Meanwhile, temperature and pressure also drops as orange particles go outwards. As for blue pathlines, since they come from outside of hurricane, they do not carry much water vapor. However, the temperature rises as blue particles come near the center.

The projection results with different distance metrics and parameters are shown in Figure 7. In Figure 7, we choose result (d) as baseline. From the results, we can observe geometry metric have totally different projection with others, since geometry metric does not consider any information in attribute space. Between Eulerian specification and Lagrangian specification, the main difference is their shape of clusters. For example, samples colored blue are centralized in (b), and become more and more dispersed as the timescope increasing in (c) and (d). These samples correspond to the seeds from highest level of hurricane center. They are similar in attribute space when using Eulerian specification. However, as we consider larger timescope, these pathlines traverse to regions which are significantly different in attribute space. As a result, these samples become more and more diverged in Lagrangian specification. With careful observation, we can also discover lots of mixing of samples with similar color in Eulerian specification. In LASP, users can also adjust attribute weights to comprehend the roles of individual attributes, as shown in Figure 7 (e)(f). The weights also provide mechanisms to navigate sub-dimensional attribute spaces.

5.2 GEOS-5 Simulation

The Atmospheric General Circulation Model (AGCM) of Goddad Earth Observation System, Version 5 (GEOS-5) from NASA Goddard Space Flight Center is developed for meteorological research and weather prediction [20]. The model is at a spatial resolution of $1^\circ \times 1.25^\circ$ lat-lon grid with 72 vertical pressure layers. It computes various attributes, including wind speed, humidity, tempera-

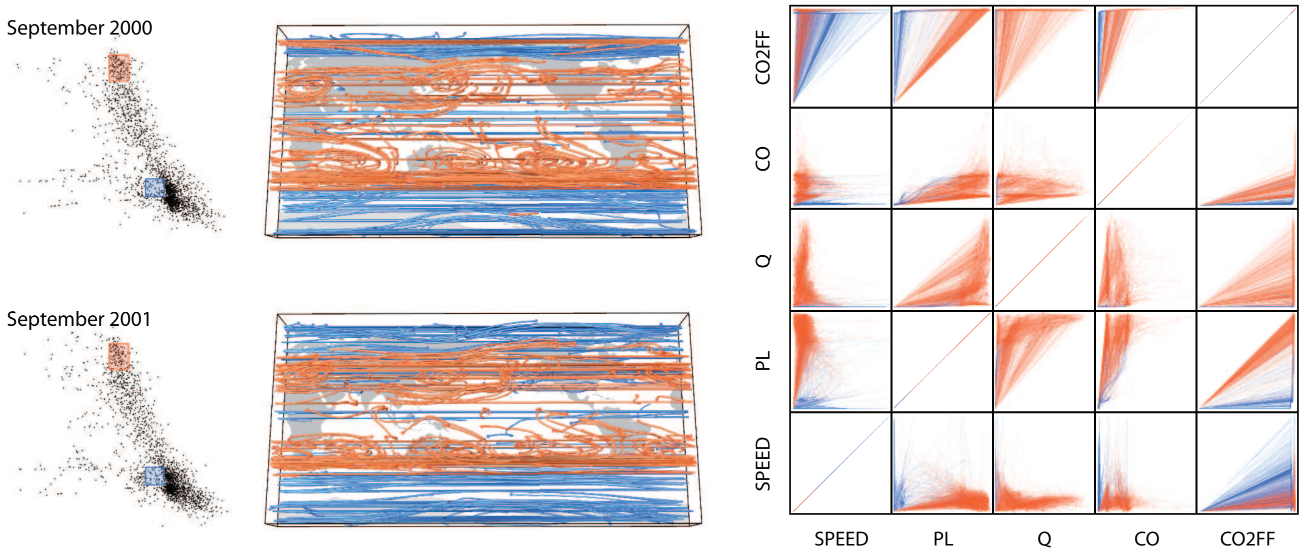


Figure 8: The visualization of GEOS-5 simulation dataset with the proposed method. Two clusters are selected in the projection view, and the corresponding pathlines are shown in the spatial views. Pathlines are also mapped in the attribute matrix to show the numerical distributions of the selected features.

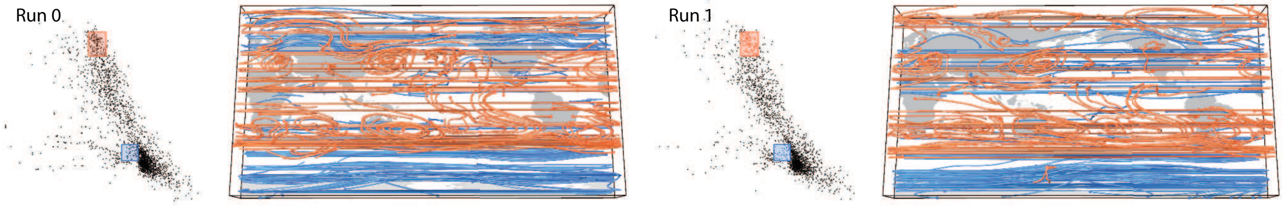


Figure 9: The LASP results and the spatial views for two runs from GEOS-5 ensembles at May 2000. Slight differences can be observed in projection results. By selecting two regions in projection plot, corresponding pathlines are shown in spatial views. Significant divergence exists in northern hemisphere for these two runs, while they are similar in southern part.

ture, atmospheric concentration of carbon dioxide, etc. Output of the simulation is stored in hybrid-sigma pressure grid. An eight-member ensemble simulation was performed previously. We use the monthly average data from January 2000 to December 2001. The data consists of 24 time steps with 35 variables in floating-point precision.The overall data size is about 76GB.

In this case, in addition to the wind field, five scalar attributes are considered, including wind speed magnitude (SPEED), mid-level pressure (PL), specific humidity (Q), global carbon monoxide (CO), and carbon dioxide fossil fuel (CO_2FF). Two interesting regions of samples are identified, which are highlighted in colored areas (Figure 8). From the spatial rendering, pathlines in two clusters are roughly separated by their spatial distribution. Pathlines with orange color are near the equator, while blue ones are more closer to polar region. By inspecting attribute space of the pathlines, we found pathlines in orange cluster usually travel regions which have much larger pressure, humidity, and concentration of CO, while these attributes of blue pathlines are in a very narrow range.

In another case, we try to investigate the difference between ensemble runs, which were conducted with slightly different initial conditions. The visualization results are shown in Figure 9. In projection plot, these two ensemble runs show little differences. In spatial views, we can observe that pathlines in these clusters have significant divergence in northern hemisphere, while they are similar in southern part.

5.3 Discussion

The coupled multivariate analysis and flow advection with LASP brings a novel perspective into unsteady flow datasets. Compared to Eulerian-based methods, LASP is capable of showing more insightful features in some applications. The Eulerian-based method is a degradation of Lagrangian-based method in theory. In real applications, unsteady flow datasets often contain both vector and scalar field data simultaneously, yet few existing methods are capable of incorporating the analysis of the both data types effectively. We have also received positive feedback on our visualization results from scientists from climate research. Further quantitative user study will be conducted to evaluate and improve our work.

There are a few limitations. First, the projection result can only show the similarities between samples, instead of specific attribute properties. Users need to identify the features by trial-and-error. Second, the parameter reconfiguration requires the recomputation of all pathlines. Due to the memory limit, all pathlines are dumped immediately after the distances are computed.

6 Conclusions and Future Work

In this work, we present a novel visualization method, which tightly couples multivariate analysis and flow advection for unsteady flow datasets with LASP. A scalable and parallel system, which combines the MapReduce-like DStep framework and scalable Pivot MDS, is designed for LASP to support large-scale analysis. In the parallel system, massive pathlines are traced in DStep framework with scalability, and the distances are computed before they are projected in parallel. The results show that the selected features in the projection show the groups of the pathlines in the attribute space, which are further visualized in the spatial view.

In the future, we would like to extend our work in several ways. Irregular and unstructured grids will be supported in the future for more applications. Currently, we uniformly sample the spatial domain, which is not efficient enough on certain occasions. We would also like to provide flexible mechanism for attribute selection and sub-dimensional space exploration. Some derived attributes, e.g. λ_2, vorticity magnitudes, can be used to help on identifying more interesting features. Dynamic strategies are going to be used to reduce the computational cost of LASP.

Acknowledgements

The authors wish to thank the anonymous reviewers for their comments. This work is supported by NSFC No. 61170204, NSFC Key Project No. 61232012, and partially by the "Strategic Priority Research Program - Climate Change: Carbon Budget and Relevant Issues" of the Chinese Academy of Sciences Grant No. XDA05040205. The authors would like to thank Dr. Xiaoguang Ma for his suggestive comments.

References

[1] Y. Bengio, J.-F. Paiement, and P. Vincent. Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering. Technical report, Departement d'Informatique et Recherche Opërationnelle, Universite de Montrëal, 2003.

[2] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *GD'06: Proc. International Conference on Graph Drawing*, pages 42–53, 2007.

[3] D. Camp, C. Garth, H. Childs, D. Pugmire, and K. I. Joy. Streamline integration using MPI-hybrid parallelism on a large multicore architecture. *IEEE Trans. Vis. Comput. Graph.*, 17(11):1702–1713, 2011.

[4] C.-M. Chen, L. Xu, T.-Y. Lee, and H.-W. Shen. A flow-guided file layout for out-of-core streamline computation. In *Proc. IEEE Pacific Visualization Symposium 2012*, pages 145–152, 2012.

[5] L. Chen and I. Fujishiro. Optimizing parallel performance of streamline visualization for large distributed flow datasets. In *Proc. IEEE Pacific Visualization Symposium 2008*, pages 87–94, 2008.

[6] W. Chen, Z. Ding, S. Zhang, A. MacKay-Brandt, S. Correia, H. Qu, J. A. Crow, D. F. Tate, Z. Yan, and Q. Peng. A novel interface for interactive exploration of DTI fibers. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1433–1440, 2009.

[7] J. Daniels, E. W. Anderson, L. G. Nonato, and C. T. Silva. Interactive vector field feature identification. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1560–1568, 2010.

[8] V. de Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *NIPS 2002: Proc. Neural Information Processing Systems*, pages 705–712, 2002.

[9] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VisSym 2003: Proc. Joint Eurographics - IEEE TCVG Symposium on Visualization*, pages 239–248, 2003.

[10] P. D. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42(11):149–160, 1984.

[11] H. Guo, H. Xiao, and X. Yuan. Scalable multivariate volume visualization and analysis based on dimension projection and parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 18(9):1397–1410, 2012.

[12] H. Guo, X. Yuan, J. Huang, and X. Zhu. Coupled ensemble flow line advection and analysis. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2733–2742, 2013.

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

[14] H. Jänicke, M. Böttinger, and G. Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1459–1466, 2008.

[15] B. Jobard, G. Erlebacher, and M. Y. Hussaini. Lagrangian-Eulerian advection of noise and dye textures for unsteady flow visualization. *IEEE Trans. Vis. Comput. Graph.*, 8(3):211–222, 2002.

[16] W. Kendall, J. Huang, T. Peterka, R. Latham, and R. Ross. Visualization viewpoint: Towards a general I/O layer for parallel visualization applications. *IEEE Comput. Graph. Appl.*, 31(6):6–10, 2011.

[17] W. Kendall, J. Wang, M. Allen, T. Peterka, J. Huang, and D. Erickson. Simplified parallel domain traversal. In *SC11: Proc. ACM/IEEE Conference on Supercomputing*, pages 10:1–10:11, 2011.

[18] J. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[19] R. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. Post, and D. Weiskopf. The state of the art in flow visualization: dense and texture-based techniques. *Comput. Graph. Forum*, 23(2):203–222, 2004.

[20] S.-J. Lin. A "vertically Lagrangian" finite-volume dynamical core for global models. *Monthly Weather Review*, 132(10):2293–2307, 2004.

[21] T. McLoughlin, R. Laramee, R. Peikert, F. Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum*, 29(6):1807–1829, 2010.

[22] K. Moreland. A survey of visualization pipelines. *IEEE Trans. Vis. Comput. Graph.*, 19(3):367–378, 2013.

[23] B. Nouanesengsy, T.-Y. Lee, K. Lu, H.-W. Shen, and T. Peterka. Parallel particle advection and FTLE computation for time-varying flow fields. In *SC12: Proc. ACM/IEEE Conference on Supercomputing*, pages 61:1–61:11, 2012.

[24] B. Nouanesengsy, T.-Y. Lee, and H.-W. Shen. Load-balanced parallel streamline generation on large scale vector fields. *IEEE Trans. Vis. Comput. Graph.*, 17(12):1785–1794, 2011.

[25] T. Peterka, R. B. Ross, B. Nouanesengsy, T.-Y. Lee, H.-W. Shen, W. Kendall, and J. Huang. A study of parallel particle tracing for steady-state and time-varying flow fields. In *IPDPS11: Proc. The International Parallel and Distributed Processing Symposium*, pages 580–591, 2011.

[26] F. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Comput. Graph. Forum*, 22(4):1–17, 2003.

[27] D. Pugmire, H. Childs, C. Garth, S. Ahern, and G. H. Weber. Scalable computation of streamlines on very large datasets. In *SC09: Proc. ACM/IEEE Conference on Supercomputing*, pages 16:1–16:12, 2009.

[28] C. Rössl and H. Theisel. Streamline embedding for 3D vector field exploration. *IEEE Trans. Vis. Comput. Graph.*, 18(3):407–420, 2012.

[29] T. Salzbrunn and G. Scheuermann. Streamline predicates. *IEEE Trans. Vis. Comput. Graph.*, 12(6):1601–1612, 2006.

[30] K. Shi, H. Theisel, H. Hauser, T. Weinkauf, K. Matkovic, H.-C. Hege, and H.-P. Seidel. Path line attributes - an information visualization approach to analyzing the dynamic behavior of 3D time-dependent flow fields. In *Topology-Based Methods in Visualization II*, pages 75–88. Springer, 2007.

[31] N. Takashi and T. J. Hughes. An arbitrary Lagrangian-Eulerian finite element method for interaction of fluid and a rigid body. *Computer Methods in Applied Mechanics and Engineering*, 95(1):115–138, 1992.

[32] W. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[33] H. Yu, C. Wang, and K.-L. Ma. Parallel hierarchical visualization of large time-varying 3D vector fields. In *SC07: Proc. ACM/IEEE Conference on Supercomputing*, pages 24:1–24:12, 2007.

[34] B. Zhang, Y. Ruan, T.-L. Wu, J. Qiu, A. Hughes, and G. Fox. Applying twister to scientific applications. In *CloudCom 10: Proc. IEEE Cloud Computing Conference*, pages 25–32, 2010.