PHYSICS-BASED FEATURE MINING FOR LARGE DATA EXPLORATION

One effective way of exploring large scientific data sets is a process called feature mining. The two approaches described here locate specific features through algorithms that are geared to those features' underlying physics.

Large-scale computational simulations of physical phenomena produce enormous data sets, often in the terabyte and petabyte range. Unfortunately, advances in data management and visualization techniques have not kept pace with the growing size and complexity of such data sets. One paradigm for effective large-scale visualization is browsing regions containing significant features of the data set while accessing only the data needed to reconstruct these regions. To demonstrate the feasibility of this approach, we are currently developing a prototype system, Evita—exploratory visualization, interrogation, and analysis.[1] The cornerstone of this visualization paradigm is a representational scheme that facilitates ranked access to macroscopic features in the data set.

We call the process of detecting those significant features *feature mining*, and in this article, we propose two paradigms for accomplishing this task. Our intent with both approaches is to exploit the physics of the problem at hand to develop highly discriminating, application-dependent feature detection algorithms and then use available data mining algorithms to classify, cluster, and categorize the identified features. We have also developed a technique for denoising feature maps that exploits spatial-scale coherence and uses what we call *feature-preserving wavelets*.

The large-data exploration methodology we describe can work for any data that can be transformed to a multiscale representation and consists of features that can be extracted through local operators and aggregated in spatial, scale, and temporal dimensions. The examples we present in this article, however, demonstrate our feature mining approach as applied to steady computational fluid dynamics simulations on curvilinear grids.

1521-9615/02/$17.00 © 2002 IEEE

DAVID S. THOMPSON, JAYA SREEVALSAN NAIR, AND SATYA SRIDHAR DUSI VENKATA
Mississippi State University
RAGHU K. MACHIRAJU, MING JIANG, AND GHEORGHE CRACIUN
The Ohio State University

Evita

The Evita system consists of three main components: an offline preprocessor, a server, and a client. The preprocessor takes the original data set and its associated grid to produce a compact representation. The compressed bitstream resulting from this offline preprocessing is produced under a fixed priority schedule that permits suitable visualization of the data set. The preprocessor organizes the bitstream in terms of *regions of interest*. Each ROI is a correlated spatiotemporal region that contains a physical feature with an associated ranking. Thus, the preprocessor is essentially a feature mining application that produces a significance map delineating the ROIs.

When an Evita user begins data exploration through the client system, the server initially transmits a background. Then, according to a feature-based priority schedule, it transmits information one feature at a time. Features appear on the client system according to the priority schedule and are incrementally refined over time, resulting in a monotonic improvement in the image quality. For each user-initiated change in the viewpoint, the server generates a new priority schedule and transmits new ROIs. While visualization is under way, the server accepts real-time information from the client and dynamically reorganizes the bitstream to produce the desired priority ranking.

The final component of the system, the client, decodes the reorganized bitstream arriving from the server and produces the visualization. The client can again extract features to gain further insights into the data.

The preprocessing stage requires the entire data set. Given the local nature of the wavelet and feature operators, the data can be accessed by blocks or piecemeal. The system can also use out-of-core and parallel execution to reduce excessive memory buffering of data during preprocessing. However, for the actual client-driven exploration, buffering the entire data set is not necessary. Only the data demarcated by the view frustum is necessary; features can be extracted from that. Thus, exploration does not require a large buffer space.

Several aspects of the Evita project have generated innovative research—for example, its embedded and progressive encoding, server-client architecture, and visualization interactors. In this article, however, we will focus exclusively on how Evita handles feature mining.

Other Feature-Mining Work

In many ways, our approach and framework parallel those of Kenneth Yip and Feng Zhao.[1] Spatial aggregation is the cornerstone of their approach. All points belonging to an identified region are aggregated to form a subdomain or a region of interest. They propose frameworks that facilitate imagistic reasoning and allow construction of frameworks for imagistic solvers. In our framework, features are extracted as simplicial entities (for example, points, lines, regions); we rely on the visceral potency of visualization algorithms to gain insights.

Yip and Zhao use aggregation and task-specific classification to create an explicit neighborhood relationship graph. Their approach uses redescription to conduct aggregation at higher levels of abstraction. On the other hand, we exploit aggregation, classification, and other tasks (denoising, tracking, and so on) to facilitate exploration of large data.

Furthermore, many of the examples described by Yip and Zhao are for 2D scalar fields; we've chosen our examples from more complex flow problems.

Reference

1. K. Yip and F. Zhao, "Spatial Aggregation: Theory and Applications," *J. Artificial Intelligence Research,* vol. 5, Aug. 1996, pp. 1–26.

What is a feature?

Perhaps the most appropriate response to this question is, "It depends on what you're looking for." In general, a feature is a pattern occurring in a data set that is of interest and that manifests correlation relationships between various components of the data. For instance, a shock in a supersonic fluid flow would be considered a significant feature: when a shock occurs, the pressure increases abruptly in the direction of the flow, and the fluid velocity decreases in a prescribed manner. A significant feature also has spatial and temporal scale coherence. In most cases, an adequately resolved feature spans several discrete spatial or temporal increments.

For many applications, generic data mining techniques such as clustering, association, and sequencing can reveal statistical correlations between various components of the data.[2] Returning to the shock example, we could use statistical mining to ferret out associations, but it might be difficult to attach precise spatial associations for the rules discovered. A fluid dynamicist, however, would like to locate features with a rather high degree of certainty. Such qualitative assertions alone will not suffice.

This is where our approach to feature mining comes in: we take advantage of the fact that, for simulations of physical phenomena, the field variables satisfy certain physical laws. We can exploit these kinematic and dynamic considerations to locate features of interest. The resulting feature detection algorithms, by their very nature, are highly application-specific. However, the fidelity improvements garnered by tailoring these highly discriminating feature detection algorithms to the particular application far outweigh any loss of generality. In this context, feature detection is, in essence, a data mining task.

The state of the art in feature detection and mining in simulation data is similar to what existed for image processing when edge detection methods were the main techniques. Much more is now understood, and mining in images is often done in terms of the features, namely edges. This suggests that a blend of data and feature mining methods might have the potential to reduce the burdensome chore of finding features in large data sets.

Feature detection algorithms

In this section, we describe two distinct feature detection paradigms. The common thread is that both are bottom-up feature constructions with underlying physically based criteria. The two perform essentially the same steps, but in different order. As will become evident, it is unlikely that non-physics-based techniques would provide the fidelity needed to locate complex flow field structures.

The feature we'll focus on is the vortex. (For an excellent review of vortex detection techniques, see Martin Roth.[3]) We all have an intuitive, informal understanding of what a vortex is—a swirling flow pattern around a central point. However, the primary challenge associated with vortex detection is that there is no clear definition of a vortex. Here is one of the literature's clearest:

> A vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern, when viewed from a reference frame moving with the center of the vortex.[4]

Unfortunately, this definition is self-referential—you have to know certain properties about a particular vortex to be able to detect it. For steady 3D flow, the vortex's orientation is, in general, unknown; for unsteady flows, the velocity of the "reference frame moving with the center of the vortex" is likewise unknown. The underlying difficulty with this definition lies in its global nature. To use this definition, you must know the orientation and velocity of a planar surface moving through space. Furthermore, you must be able to deduce streamline patterns.

Now we'll show how to apply our two different feature detection paradigms to vortical flows.

Point classification techniques

The first feature detection paradigm, which we call *point classification*, requires several operations in sequence:

1. Detection by application of a local sensor at each point in the domain
2. Binary classification (verification) of points based on some criteria
3. Aggregation of contiguous regions of like-classified points
4. Denoising to eliminate aggregates that are of insufficient extent, strength, and so on
5. Ranking based on feature saliency

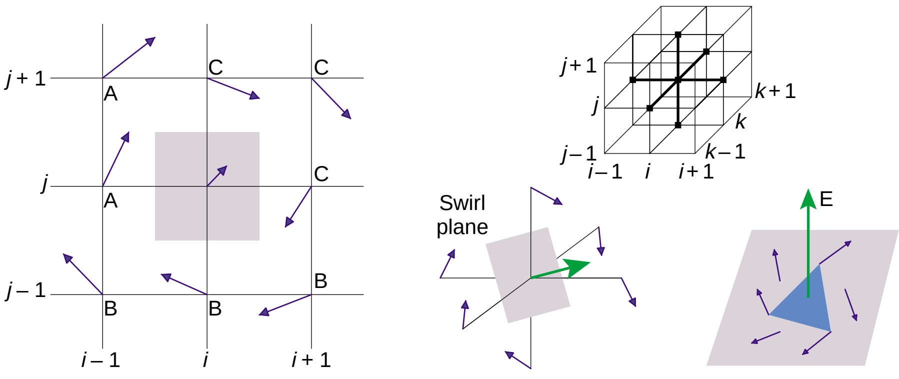
This approach identifies individual points as belonging to a feature and then aggregates them to identify regions that are features. The points are obtained from a tour of the discrete domain and can be in many cases the vertices of a physical grid. The sensor used in the detection phase and the criteria used in the classification phase are *physically based point-wise characteristics* of the feature of interest. (We could also track these regions in the temporal dimension, but in this article we'll restrict our attention to feature extraction.)

Now let's look at a vortex detection technique that uses the point classification approach. This technique uses the eigenvalues of the local velocity gradient tensor.[5] Under a limited set of conditions, swirling flow is characterized by regions where the eigenvalues of the velocity gradient tensor are complex. Carl Berdahl and David Thompson defined a swirl parameter that estimates "the tendency for the fluid to swirl about a given point."[5] The value of this parameter is

$$\tau = \frac{\left|\mathrm{Im}\left(\lambda_{1,2}\right)\right| L}{V_{conv}},$$

where $\mathrm{Im}(\lambda_{1,2})$ is the imaginary part of the complex conjugate pair of eigenvalues, V_{conv} is the velocity in the plane whose normal is the real eigenvector, and L is the characteristic length of the swirling region. The swirl has a nonzero value in regions containing vortices and attains a local maximum in the core region.

In this point classification algorithm, the detection step consists of computing the eigenvalues of the velocity gradient tensor at each field point. The classification step consists of checking for complex eigenvalues and assigning a swirl value if they exist. The aggregation step then de-



(a) (b)

Figure 1. Topology-based vortex core region detection algorithm: (a)2D algorithm and (b) 3D algorithm.

fines the region containing the vortex.

This method's primary shortcoming is that it—and all eigenvalue-based vortex detection techniques—can generate false positives. Its local nature makes it unable to discriminate between locally curved streamlines and closed streamlines characteristic of a vortex. Other features, such as shocks, are more amenable to the point classification framework.

Aggregate classification techniques

We can best incorporate the global information needed to define a vortex into our second feature detection paradigm, the *aggregate classification* approach. Aggregate classification follows a somewhat different sequence of operations:

1. Detection by application of a local sensor at each point in the domain
2. Aggregation of contiguous regions of probable candidate points
3. Binary classification (verification) of each aggregate based on some criteria
4. Denoising to eliminate aggregates that are of insufficient extent, strength, and so on
5. Ranking based on feature saliency

This approach identifies individual points as being probable candidate points in a feature and then aggregates them. The classification algorithm is applied to the aggregate using *physically based regional criteria* to determine whether the candidate points constitute a feature.

We have recently developed an aggregate classification vortex detection technique.[6] We based its detection step on an idea derived from a lemma in combinatorial topology: Sperner's lemma states, "Every properly labeled subdivision of a simplex σ has an odd number of distinguished simplices."[7] In other words, given a convex set in n dimensions, triangulate it into subtriangles and assign to each vertex of the subtriangles a label from 1, 2, ..., $n + 1$. If the initial vertices of the convex set are completely labeled, then there exist an odd number of completely labeled subtriangles within the convex set. A subtriangle is completely labeled if it receives all $n + 1$ labels.

The idea behind Sperner's lemma is to deduce the properties of a triangulation based solely on the labeling of the vertices. In a dual fashion, this approach deduces the behavior of a vector field based solely on the labeling of the velocity vectors. In particular, velocity vectors around core regions exhibit certain flow patterns unique to vortices, and it is precisely these flow patterns that we search for in the computational grid. Not surprisingly, our approach is related to critical point theory. A critical point is a point at which the velocity is zero—that is, where the local slope of the streamline is undefined. However, critical points alone are not sufficient to detect a vortex.

For each grid point, our algorithm examines its four immediate neighbors in 2D—six in 3D—to see whether the neighboring velocity vectors point in three or more direction ranges (that is, to see if they form a complete triangle). 3D vortex core regions are much more difficult to detect than their 2D counterparts: to do so, we must identify a core direction (the normal to a plane) and apply our 2D algorithm to the neighboring vectors projected onto that plane. We call this plane the *swirl plane* because instantaneous streamlines projected onto it exhibit a swirling pattern.

Figure 1a shows the 2D algorithm on a 2D structured grid with the detected core region colored gray; Figure 1b shows the 3D algorithm, along with the swirl plane, and the complete tetrahedron A, B, C, E. Potential candidates for the core direction vector include the vorticity

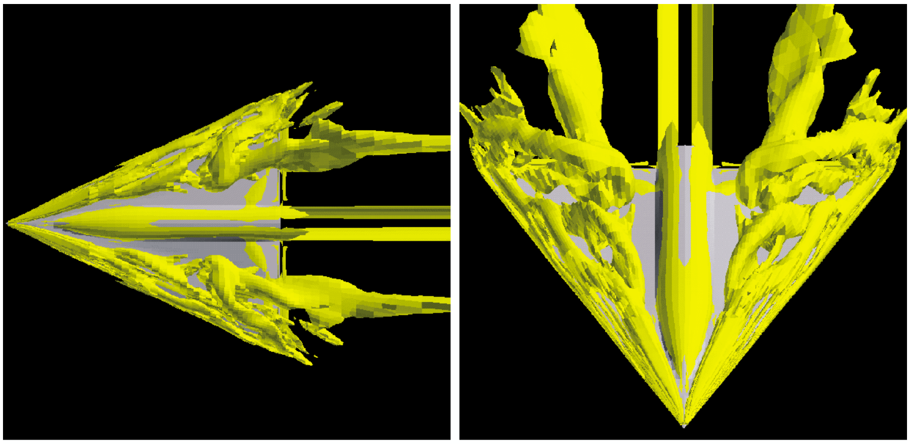


Figure 2. Point-based vortex detection algorithm applied to a delta wing with vortex burst.

vector and the eigenvector corresponding to the real eigenvalue of the velocity gradient tensor.

The final point to consider in the detection step is the issue of *direction quantization.* Direction quantization refers to the number of possible direction ranges in which a vector can point—that is, the number of possible labels a vector can receive. Given a continuous vector field defined on a discrete 2D grid, it is not always sufficient to use only three direction ranges to label the vectors. For the cases considered to date, four direction ranges have proved sufficient. An added benefit of direction quantization is that it makes the 3D algorithm relatively insensitive to the core direction, and approximate core directions can be just as effective as exact core directions.

Our technique segments candidate core regions by aggregating points identified from the detection phase. We then classify (or verify) these candidate core regions based on the existence of swirling streamlines surrounding them. (For features that lack a formal definition, such as the vortex, we must choose the verification criteria so that it concurs with the intuitive understanding of the feature. In this case, verifying whether a candidate core region is a vortex core region requires checking for any swirling streamlines surrounding it.) Checking for swirling streamlines is a global (or aggregate) approach to feature classification (or verification) because swirling is measured with respect to the core region, not just individual points within the core region.

In two dimensions, checking for swirling streamlines is fairly straightforward. Using operators from differential geometry, we can measure swirling by computing a streamline's winding angle with respect to a candidate core region. The winding angle is a measure of the total curvature, or the signed angle of rotation, of a planar curve with respect to a reference point. Therefore, a winding angle of 2π means that the planar streamline has completely swirled around the candidate core region, making it the natural choice for the classification criteria in two dimensions.

In three dimensions, however, checking for swirling streamlines is much more difficult, because we cannot extend the winding angle operator into higher dimensions, and vortices in three dimensions can exhibit geometries that bend or twist in various different ways. To address these issues, our verification algorithm computes the tangent vector and probe vector for each point along the streamline. The probe vector is oriented to point at or near the core region, and it retrieves the core direction vector from that location. The algorithm locally aligns the retrieved core direction vector with the z-axis and then applies the same transformation to the tangent vector. Essentially, the purpose of this alignment step is to locally straighten any curved vortices along the z-axis. The algorithm then projects the transformed tangent vector onto the (x, y)-plane; therefore, if the streamline is swirling, the projected tangent vectors make a complete revolution in the (x, y)-plane. Thus, the classification criterion in three dimensions is a signed angle of rotation of 2π, in the (x, y)-plane, by the projected tangent vectors.

Examples

Now we'll demonstrate the two techniques we've discussed by applying them to a complicated, delta-wing flow field that has undergone vortex burst—a condition characterized by the rapid expansion of the vortex. Figure 2 shows two views of the isosurfaces of the point classi-

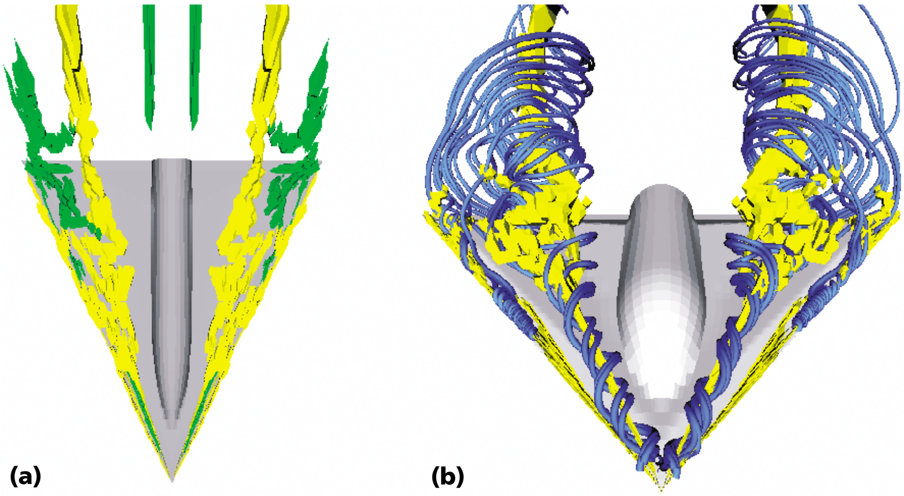


Figure 3. Aggregate-based vortex detection algorithm applied to a delta wing with vortex burst: (a) yellow candidate core regions are actual, green are false; and (b) verified core regions.

fication–based swirl parameter, with the complex structures of the burst vortex clearly evident.

Figures 3 and 4 show results generated using the aggregate classification technique. Figure 3a shows the candidate core regions. The yellow aggregated regions are those identified by the classification step as being actual cores; the green aggregated regions failed the verification step. Figure 3b shows the verified core regions enclosed in swirling streamlines. Figure 4 illustrates the verification technique. The cyan vectors in the upper image are the streamline tangents, and the orange vectors are the probe vectors. The probe vectors interrogate the core region for the local core direction vector. The bottom image shows that the projected tangent vectors satisfy the 2π swirling criterion.

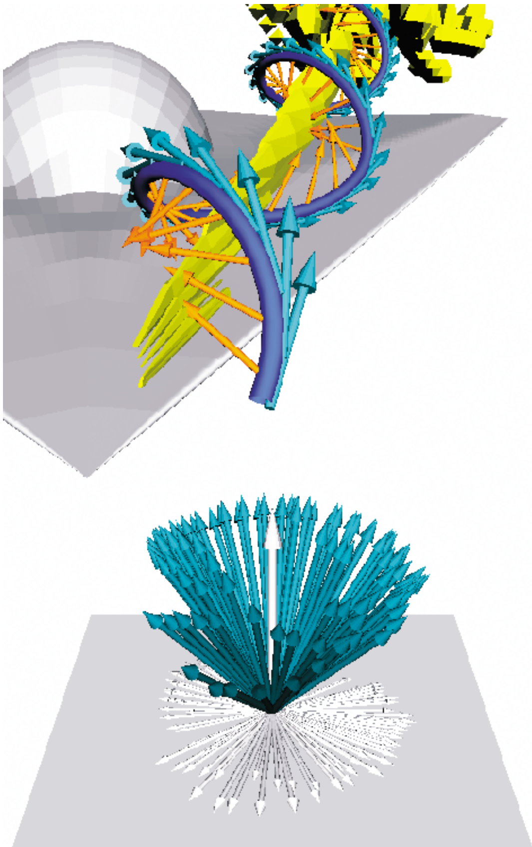


Figure 4. Verification process for primary vortex, aggregate classification technique.

Wavelet-based denoising

Once we have obtained a feature map, the next step is to filter and rank the ROIs systematically. The visualization process should not accord significance to features that are weak or of small spatial extent. In addition, certain types of features require several grid points for the simulation to adequately resolve them; the visualization should also ignore features that don't meet this criterion. Our basic strategy, which eliminates all these insignificant features, is to accord ROI status only to features that persist over several spatial scales. In other words, we exploit the scale coherence of significant features to determine whether a detected feature is, in fact, an ROI.

Once we have denoised the feature map, we can rank the remaining ROIs according to appropriate criteria—for example, size, strength, average strength, and so on. A scale space generated by the wavelet transform is even more attractive. Features of small spatial extent, possibly noise, populate the finer scales. However, true features populate several scales. Because our data exploration system employs the wavelet transform, using scale-based denoising was a natural choice. Another study, which used masks derived from the swirl operator to ascribe a saliency, showed that scale space denoising is more powerful than spatial methods that employ size or value as a criterion for rejecting regions as feature-poor.[8] Thus, we rank regions by measuring the persistence of features in the discrete scale space.

Multiscale representation

To obtain a multiscale representation of the feature map, we apply a discrete wavelet transformation (DWT) to the grid and the field data. Because we are interested in the presence of features in a multiscale representation of the data, we do not apply the wavelet transform directly to the feature maps but, instead, to the field data itself. We then apply the feature detection algorithm at each scale to generate scale-dependent binary maps. Then, we combine the binary maps at each scale to generate a single denoised binary map. Because many feature detection algorithms are based on gradients of the field variables, it is critical that the wavelet transform not introduce spurious features.

Although denoising the data might seem to be independent of the application, we contend that the denoising procedure itself should preserve certain feature characteristics. For generality, we use physical characteristics including position, shape, and strength (or geometrical characteristics) instead of the features' dynamical properties. Our objective is to employ what we term feature-preserving wavelets to generate a multiscale representation of the data. We consider a wavelet feature-preserving if it does not distort geometrical characteristics. Many commonly used wavelet functions do not exhibit favorable feature preservation characteristics. For example, deriving a coarser version of a function can introduce new extrema that can lead to the detection of spurious features.

Elsewhere, we've outlined the design of a family of functions that satisfy certain feature preservation properties in a multiscale setting.[9] We implement these functions as the low-pass analysis filter in a two-channel filter bank. We use a factorization method to determine the other components of the filter bank. Because our focus here is feature mining, we will not elaborate further. However, we do stress that it is important to choose appropriate transforms for feature mining applications. The two wavelets we chose for this application were the linear lifting scheme[10] and a newly developed lifting implementation of a feature-preserving total variation diminishing (TVD) wavelet.[9]

Significance map generation

We've already obtained the significance or feature map from either the point classification or the aggregate classification feature mining algorithm. Now we convert this map to a binary form, with a 1 signifying the presence of a feature at a grid point.

The multiscale procedure for generating a robust, denoised significance map is as follows. We update the map at the finer scale j using the map at the lower, or coarser, scale $j - 1$. Because of the 2D wavelet transformation's dyadic nature, each grid point (l, k) in scale $j - 1$ corresponds to point $(2l, 2k)$ in scale j. Therefore, if a cell in the coarser scale $j - 1$ is defined by grid points (l, k), $(l + 1, k)$, $(l + 1, k + 1)$, and $(l, k + 1)$, the corresponding cell in the finer scale j is defined by $(2l, 2k)$, $(2l + 2, 2k)$, $(2l + 2, 2k + 2)$, and $(2l, 2k + 2)$.

The rules for updating scale j using scale $j - 1$ are as follows:

- If the map at any of the grid points in scale $j - 1$ is a 0, make the map at the corresponding grid point in scale j a 0.
- If both the grid points on an edge of the cell in scale $j - 1$ are 0s, make the map at the midpoint on the corresponding edge in scale j a 0.
- If all four grid points in a cell are 0s in scale $j - 1$, make the map 0 at the midpoint of the corresponding cell in j.

Apart from these changes, the rest of the grid points on the map remain unmodified. Thus, features that did not percolate down to the lower scale are marked with 0s. Because we have added no 1s, we haven't created any new features. We apply this procedure recursively over two or more scales—that is, we use a denoised map at a lower scale to denoise the higher scale.

Examples

We implemented and tested a 2D version of this approach using a section of the Naval Layered Ocean Model Pacific Ocean data set.[11] Figure 5a shows the original binary swirl map. Figure 5b shows the swirl map obtained after denoising using the linear lifting wavelet; Figure 5c shows that obtained after denoising using a feature-preserving TVD wavelet.

Although it is evident that both filters remove pixels from the maps, visual inspection does not provide a clear indication of the denoising algorithms' relative merits. Of 339 original features, feature-preserving wavelet denoising removed 82 features in one level of transform, 116 features in two levels, and 158 features in three levels. On the other hand, the linear lifting wavelet

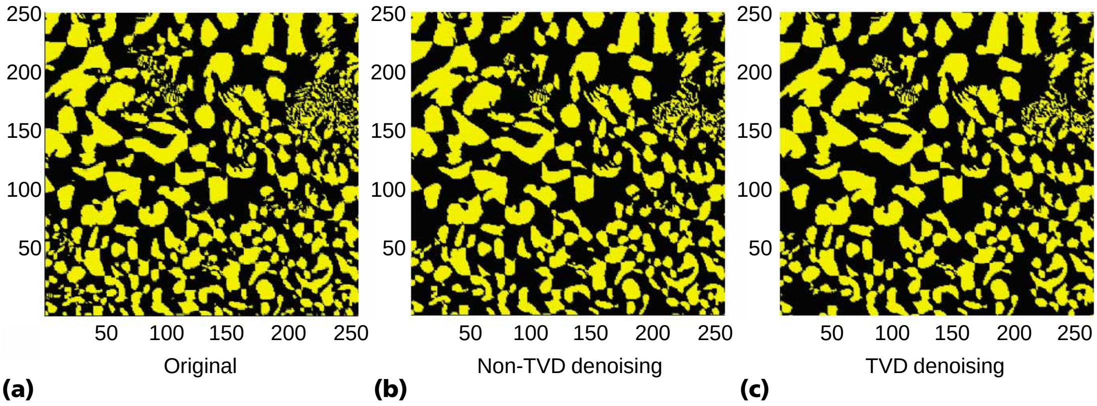


Figure 5. Wavelet-based denoising: (a) original binary swirl map, (b) map after denoising with the linear lifting wavelet, and (c) map after total variation diminishing denoising.

eliminated 67, 105, and 150 features for one, two, and three levels of transform.

Obviously, the feature-preserving wavelet eliminated more features than the linear lifting wavelet. What is interesting, however, is the manner in which elimination occurred. Figures 6a and 6b show the distribution of features in the original data and after one, two, and three levels of denoising for the linear and feature-preserving wavelets. We categorized features using their average swirl (using $\log_{10}\tau$). In general, both wavelets do a good job of eliminating weaker features ($\tau < -2.5$). However, even though the feature-preserving wavelet eliminates more features, it preserves more of the strongest features as measured by average swirl ($\tau > -1.5$). Our interpretation is that the feature-preserving wavelets do a better job of preserving the significant features. We can now rank the remaining features using a criterion such as average swirl.

The basis of our efforts to improve feature mining algorithms is the assertion that, for physics-based simulations of complex phenomena, we should exploit the inherent relationships between the various components of the data. Traditional data mining algorithms alone cannot guarantee success. In cases where we understand the underlying physics, at least at some basic level, it makes sense to exploit the known correlations whenever possible. Taken together, application-specific feature detection algorithms and application-independent techniques from traditional data mining provide an arsenal that offers much promise in solving the problem of effective exploration of large data sets.

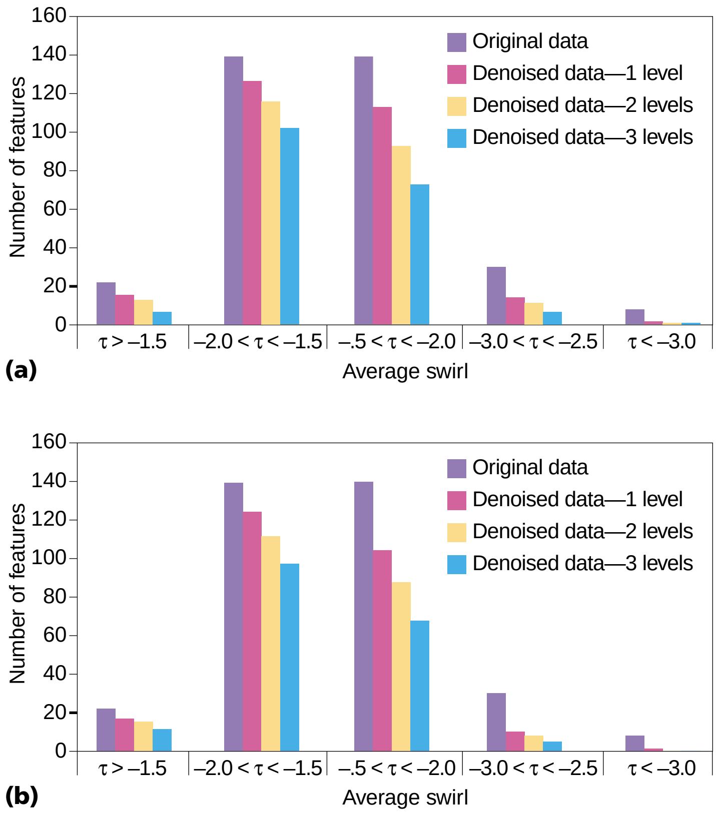


Figure 6. Denoising effectiveness based on average swirl: (a) linear lifting wavelet and (b) feature-preserving total variation diminishing wavelet.

Acknowledgments

We gratefully acknowledge the valuable contributions of our collaborators on the Evita project—James E. Fowler and Bharat Soni of Mississippi State University and Will Schroeder of Rennselaer Polytechnic Institute. This work is partially funded by the NSF under the Large Data and Scientific Software Visualization Program (ACI-9982344), the Information Technology Research Program (ACS-0085969), and an Early Career Award (ACI-9734483). Additional support was provided by a grant from the Army Research Office (DAA D19-00-1-0155). We also thank the anonymous reviewers for many useful suggestions.

References

1. R. Machiraju et al., "EVITA—Efficient Visualization and Interrogation of Tera-scale Data," *Data Mining for Scientific and Engineering Applications*, R. Grossman et al., eds., Kluwer Academic Publishers, Norwell, Mass., 2001, pp. 257–279.
2. J. Han and M. Kamber, *Data Mining*, Morgan Kaufmann, San Francisco, 2001.
3. M. Roth, *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization,* doctoral thesis, Swiss Federal Institute of Technology, Dept. Computer Science, Zurich, 2000.
4. S.K. Robinson, "Coherent Motions in the Turbulent Boundary Layer," *Ann. Rev. Fluid Mechanics*, vol. 23, 1991, pp. 601–639.
5. C. Berdahl and D. Thompson, "Education of Swirling Structure using the Velocity Gradient Tensor," *American Inst. of Aeronautics and Astronautics J.,* vol. 31, no. 1, Jan. 1993, pp. 97–103.
6. M. Jiang, R. Machiraju, and D. Thompson, "Geometric Verification of Features in Flow Fields," submitted for publication, *Proc. IEEE Visualization 2002,* IEEE CS Press, Los Alamitos, Calif., 2002.
7. M. Henle, *A Combinatorial Introduction to Topology,* Dover, New York, 1979.
8. B. Nakshatrala, R. Machiraju, and D. Thompson, "Ranked Representation of Vector Fields," *Proc. Dagstuhl 2000 Seminar on Scientific Visualization,* Kluwer Academic Publishers, Norwell, Mass. (to appear).
9. G. Craciun et al., "A Framework for Filter Design Emphasizing Multiscale Feature Preservation," *Proc. Army High-Performance Computing Research Center and Center for Applied Scientific Computation/Lawrence Livermore National Laboratory 3rd Workshop on Mining Scientific Data Sets,* Soc. for Industrial and Applied Mathematics, Philadelphia, 2001, pp. 105–111.
10. W. Sweldens, "The Lifting Scheme: A Construction of Second Generation Wavelets," *SIAM J. Mathematical Analysis*, vol. 2, no. 29, 1997, pp. 511–546.
11. A.J. Wallcraft, *The Naval Layered Ocean Model Users Guide,* tech. report 35, Naval Oceanographic and Atmospheric Research Laboratory, Bay St. Louis, Miss., 1999.

David S. Thompson is an associate research professor at the Center for Computational Systems at Mississippi State University. His research interests include feature mining in large data sets, mesh generation, and aircraft icing. He received a BS and an MS in aerospace engineering from Mississippi State and a PhD in aerospace engineering from Iowa State University. He is a member of the IEEE. Contact him at Box 9627, Mississippi State, MS 39762-9627; dst@erc.msstate.edu.

Raghu K. Machiraju is an assistant professor of computer and information science at the Ohio State University. His research interests include large data visualization and analysis, modeling, and image synthesis. He received his PhD from the Ohio State University. Contact him at 395 Dreese Labs, 2015 Neil Avenue Mall, The Ohio State University, Columbus, OH 43210; raghu@cis.ohio-state.edu.

Ming Jiang is a PhD candidate in computer science at the Ohio State University. His research interests include feature detection, flow visualization, and computer graphics. He received his BS in computer science from the Ohio State University. Contact him at 395 Dreese Labs, 2015 Neil Avenue Mall, The Ohio State University, Columbus, OH 43210; jiang@cis.ohio-state.edu.

Jaya Sreevalsan Nair is a graduate student completing her MS in the computational engineering program at Mississippi State University. She plans to continue her education in the PhD program in computer science at the University of California, Davis, in the Fall. Her research interests include wavelets and visualization. She received a BS in aerospace engineering from the Indian Institute of Technology at Madras. Contact her at Box 9627, Mississippi State, MS 39762-9627; jaya@erc.msstate.edu.

Gheorghe Craciun is a PhD candidate in the Mathematics Department at Ohio State University. He is interested in dynamical systems, wavelet design, and chemical and biological applications. He has a BS in mathematics from Bucharest University and an MS in mathematics and an MS in computer science from Ohio State University. He is a member of the American Mathematical Society. Contact him at Department of Mathematics, The Ohio State University, Columbus, OH 43210; gcraciun@math.ohio-state.edu.

Satya Sridhar Dusi Venkata recently completed his MS in computational engineering at Mississippi State University. His research interests include mesh generation and feature detection. He received a BS in mechanical engineering from the Indian Institute of Technology at Madras. Contact him at 89 Beechwood Place, Plano, TX 75075; sridhar@erc.msstate.edu.