Structural Inferences from Massive Datasets

Kenneth Yip
MIT AI Lab
Cambridge, MA 02139

Abstract

High-level understanding of data must involve the interplay between substantial prior knowledge with geometric and statistical techniques. Our approach emphasizes the recovery of basic structural elements and their interaction patterns in order to summarize and draw inferences about the significant features contained in the data. As a testbed for modeling how scientists analyze and extract knowledge of structure morphogenesis from data, we examine the datasets obtained from numerical simulation of turbulence. We describe a program that automatically extracts 3D structures, classifies them geometrically, and analyzes their spatial and temporal coherence. Our program is constructed by mixing and matching the aggregate, classify, and re-describe operators of the spatial aggregation language. The research is a continuation of the effort to investigate the role of imagistic reasoning in human thinking.

Introduction

An essential survival skill for humans is the ability to perceive objects, classify them, and predict their behavior. Much of the neural machinery is devoted to these tasks. We believe that the machinery that forms the foundation of intelligence is to be found in the mechanisms that support, for example, the vision, the language, and the sensorimotor faculties. Each faculty has to solve the figure-ground problem: To identify, interpret, manipulate, and track salient objects.

It is important to understand how each such modality contributes to overall intelligence. Each modality draws on proprietary representations and problem-solving competences. Each takes inputs from internal sources as well as from the outside world. For example, a linguistically represented problem can require us to harness the power of our visual system so as to engage the strategies of "imagistic reasoning" or visual imagination. What appears to be a high-level cognitive ability emerges from cooperative problem-solving as peripherals work out parts of a problem that have been reexpressed in terms of peripheral-specific representations.

If humans do have and make use of these peripheral representations, what is the form of these representations? What are the processes that manipulate them?

In this paper we focus on the development and new applications of imagistic reasoning. In particular, we propose a model to reason about spatially distributed, temporally evolving 3D structures in the domain of fluid dynamics. Our model aims to capture the process by which human scientists interpret numerical simulation data. Our approach follows the spatial aggregation framework: Programs are implemented by mixing and matching a few high-level operators such as aggregate, classify, and redescribe. We report on our progress in implementing the performance model.

The research is inspired by the practical needs for analyzing large amount of data and the intellectual curiosity to understand how humans discover regularities, summarize significant events, and debug conceptual models. We use scientific datasets as a testbed in order to exploit the vast amount of accumulated specialized knowledge. Scientists' reasoning about fluid dynamical objects rely on the same intuitive concepts of objects: cohesiveness and continuity (Spelke *et al.*, 1995). An object is cohesive if it is internally connected, externally bounded, and moves while maintaining its connectedness and boundedness. The motion of an object is not arbitrary; it traces one connected path in space-time and leaves no gaps.

Describing complex fluid phenomena in terms of such intuitive notions of cohesive objects[1] and their interaction patterns is useful for qualitative understanding and almost necessary for developing new conceptual models of dynamical mechanisms.

Human understanding

The entire theory of incompressible fluid flow is contained in the Navier-Stokes equations. However, progress in analyzing the equations analytically has been slow. Much of the understanding of flow mechanisms comes from an interplay between numerical simulation and experimental data. To see an example of this kind of understanding that fluid dynamicists arrive at after detailed investigation of simulation data, let us quote a passage in the concluding section of (Robinson, 1991):

> The main conclusion is that the self-maintaining cycle of turbulence production in the boundary layer is driven by the formation or regeneration

[1] The term "coherent objects" is more commonly used in the turbulence research community.

of embedded vortical structures... A mature vortical arch gives rise to a trailing quasi-streamwise vortex... The quasi-streamwise vortex elements collect and lift low-speed near-wall fluid, leaving behind a persistent low-speed streak. Relatively high-speed fluid scrubbing the vortex-lifted low-speed fluid creates a shear layer which rolls up into a new vortical arch. The new arch grows outward by agglomeration and/or self-induction and circulation lift, and the cycle repeat itself.

See the diagram in Figure. 1 accompanying the explanation.

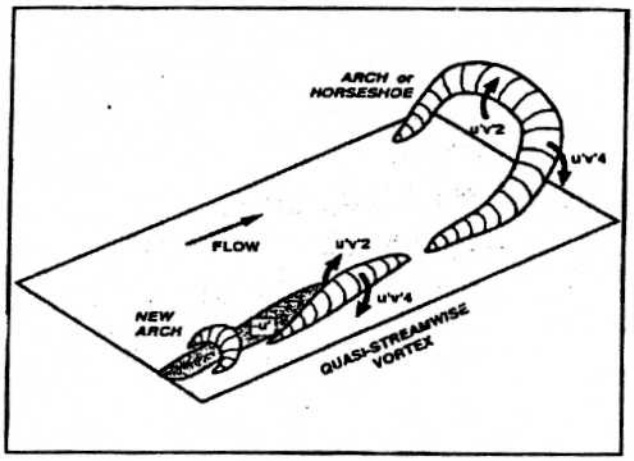

Figure 1: A conceptual model summarizing the morphogenesis of vortex structures near a turbulent boundary layer.

Much can be learned from this explanation. First, the explanation is conceptual: it does not provide any new equations. Second, the explanation refers to various kinds of structures: vortices [2], streaks, and shear layer. Third, the spatial character (e.g., arch-shaped, streamwise, persistent) of the structures are described and related. And finally the dynamical process is described in terms of structural morphogenesis: the growth and agglomeration of structures and their effect on others (e.g., lifting, rolling up).

The Spatial Aggregation Framework

In recent years, a computational framework, *spatial aggregation*, has been developed to unify the description of a class of imagistic problem solvers (Yip and Zhao, 1996). A program written in this framework has the following properties. It takes a continuous field and optional objective functions as input, and produces high-level descriptions of structure, behavior, or control actions. It computes a multi-layer of intermediate representations, called spatial aggregates, by forming equivalence classes and adjacency relations. It employs a small set of generic operators such as aggregation, classification, and localization to perform bidirectional mapping between the information-rich field and successively more abstract spatial aggregates. It uses a data structure, the neighborhood graph, as a common interface to modularize computations.

Our performance model is implemented in this style. In the following, we abstract away the details of the programming issues and focus on the incorporation of new aggregators and classifiers for manipulating 3D structures.

Performance Model

Objective and Strategies

The objective of the performance model is:

Given a sampled flow-field, construct a conceptual model to summarize the significant events implicit in the numerical data.

Numerical simulation of turbulent flow generates massive amount of data. Even a modest sampled flow-field consists of tens of gigabytes of data describing quantities such velocity, pressure, and vorticity at each sampled point. From observing how some scientists work, we propose a 5-step strategy to summarize data:

1. Isolate and characterize interesting structures.
2. Determine the spatial relationships among structures.
3. Track the morphogenesis of structures backward and forward in time.
4. Describe the statistical and dynamical significance of each structural event.
5. Note correlations among structural events and assign cause-and-effect relationships.

Not all of these steps have been automated. In the following we describe the pieces that have been implemented.

Aggregating field points

Many categories of structures can be defined as the isosurface of a scalar field. An isosurface of a scalar field F is defined as the set of points (x,y,z) such that $F(x,y,z) = c$ where c is a constant and is often called the threshold of the isosurface. Some examples are the low pressure region, the high-speed streak, and the eigenvalues of the velocity gradient.

A popular method to render isosurface is the marching cube algorithm (Lorensen and Cline, 1987). The algorithm divides the field into little cubes. The intersection of the surface with the cube is determined by a table-lookup based on the signs of the values $F(x,y,z) - c$.[3] The algorithm is reasonably fast, but does not

[2] Vortex is an intuitive notion denoting some compact region of swirling fluid motion; it is related to but not the same as the vector quantity vorticity, the curl of the velocity, at a flow field point.

[3] There are 2^8 ways a cube intersects a surface. The number can be reduced to 14 distinct configurations using rotation and reflection symmetries. In our implementation, we find it more convenient to use the entire table of 2^8 intersection patterns.

guarantee a topologically consistent surface when adjacent cubes share a face where the vertices on the two diagonals differ in sign.

Our aggregator extends the basic marching cube algorithm in four ways:

- *Recursive subdivision.* The aggregator recursively subdivides an ambiguous cube into 8 little cubes until the newly generated cubes are unambiguous. The subdivision method gives accurate results subject to the limit in sampling resolution. The values at the sub-sampled points are determined by polynomial interpolation using values from adjacent cubes.
- *Multiple isosurfaces.* It requires little overhead to compute surfaces corresponding to different thresholds during a single march over the entire flow field.
- *Connected components.* The connectivity of a surface patch intersecting a cube are precomputed and stored in each cube pattern. The global connected components of the surface are determined by aggregating individual patches.
- *Volume estimate.* The volume enclosed by an isosurface is computed from the contributions from all the cubes on and inside the surface. Each cube pattern is pre-partitioned into tetrahedra representing the volume enclosed by a surface patch.

Other structures are defined by integral curves of a vector field. For example, a vortex line, the integral curve of the instantaneous vorticity field, is the basic building block of a vortex structure. However, it is important to distinguish vorticity lines (lines everywhere parallel to the vorticity vector) and vortices (regions of nearly circular motions in the plane normal to the core of the vortex observed at the core speed). In particular, vortices are much more cohesive and less noisy than vorticity lines. We adopt the following working definition for vortices:

Definition A *coherent vortex* is a compact bundle of adjacent, high-intensity vortex lines that are geometrically similar.

Aggregating vortex lines to form coherent structures is not straightforward. Previous researchers (Moin and Kim, 1985; Robinson, 1991) have found that vortex lines are sensitive to initial conditions. Nearby vortex lines can diverge rapidly. If the initial conditions are not chosen carefully, the resulting vortex lines are likely to resemble badly tangled spaghetti wandering over the whole flow field, making the identification of organized structure extremely difficult. This might explain why vortex lines have not been widely used for structure identification.

Our search algorithm for vortices exploits local geometric information (e.g., converging or diverging) of the vorticity lines to decide the direction and step size of integration. The algorithm has the following steps:

1. Find all grid points that are local extrema of vorticity magnitude and greater than a threshold. These are the seed points.
2. On the plane normal to the largest vorticity vector component at the seed point, find an isocontour centered at the point. The contour, discretized into points, represents the initial cross section of the surface.
3. Interleave the advancement of the cross section by integration and the tiling of the surface.
4. Use the geometry of the tiles to decide shrinking or expanding the cross section locally.
5. The cross section is periodically adjusted globally by computing its convex hull, and the diameter and width of the hull.
6. The forward integration terminates when the circulation on the cross section falls below certain threshold.
7. Reconstruct the surface by integrating the last cross section *backwards* until it reaches the initial cross section. No adjustment to cross section is needed in this step.
8. Remove the weak vortex lines.

Classifying surfaces

Structures are classified by a bidirectional search strategy that simultaneously transform stored prototype models and raw sampled signals for the purpose of matching and recognition (Ullman, 1996, Chap 10). Our shape models include:

- generalized cylinders (Binford, 1990), and
- superquadrics (Barr, 1984; Bajcsy and Solina, 1987; Gupta and Bajcsy, 1993)

A generalized cylinder consists of a spline, a cross-section, and a sweeping rule. It is useful for representing elongated slender structures which bends and turns (such as a streamwise vortex). The superquadrics are generalizations of the quadratic surfaces such as ellipsoids and toroids. They are less general than the generalized cylinders, but are more appropriate for blob-like structures (like pressure regions) and structures with holes (like a vortex ring).[4]

The superquadrics are a relatively coarse shape representation characterized by several intrinsic shape parameters such as the size parameters, the roundness parameters along the north-south and east-west axes, the hole parameter, the bent angle, and the tapering parameters. To fit a surface with superquadrics, one has to recover extrinsic parameters (3 position and 3 orientation parameters) as well as the intrinsic parameters. A nonlinear least square minimization procedure (such as the Levenberg-Marquardt method (Press *et al.*, 1992)) is commonly used to recover the superquadrics parameters.

Our experience with the Levenberg-Marquardt method suggests that the minimization result is quite sensitive to the estimate of the orientation parameters. Our strategy is to try multiple estimates of the parameters in parallel. These estimates correspond to rough

[4] Of course the two representations overlap in the classes of structures for which they are appropriate. Our classifier uses both of them. We believe the perceptual mechanisms uses many different representations.

guesses of the general shape of the object: Is it an ellipsoid? It is a toroid? Is it slender? Is it bent?

Bottom-up processes that extract features from raw signals cooperate to narrow down the range of parameters for matching. For example, the surface points are re-centered at the centroid of the points. They are re-oriented along directions that minimize or maximize or average the thickness of the distribution of points.[5] These directions provide a few canonical views to reduce the combinatorics in the number of model transformations that have to be tried. The thickness of the distribution along each direction gives an estimate for the intrinsic size parameters.

The decision for best match is based on a majority voting scheme. Each process – bottom-up or top-down – contributes to the overall decision.

Re-describing Surfaces

A classified surface is a compact parametric description of a large number of field points. Like a lambda abstraction, the re-describe operator encapsulates the classified surface as a primitive object for further processing. For example, the center of mass, volume, orientation, and inertia tensor of the object are computed. From these quantities we estimate the distribution of characteristic objects (e.g, streamwise slender vortices) in interesting regions of the field (e.g., near the surface of a free surface flow).

Object Cohesiveness

There is a degree of arbitrariness in defining structures in terms of isosurfaces. The shape of the object depends on the threshold chosen, but it is often unclear how to set the appropriate thresholds. For example, what is the threshold for a low pressure region? Intuitively we prefer structures that are stable against small changes in the threshold. We compute a range of thresholds (the lower and upper bound of the thresholds are typically known) for each scalar field. Geometric properties of the re-described objects are plotted as functions of the thresholds. We choose the mean value in the largest threshold region in which the structures are stable as the desired threshold.

Object Persistence

Structures typically persist over time. They evolve for a while and may grow or shrink or collide with other structures. Unlike structure recovery from 2D images, the analogous correspondence problem in recovery from 3D fields is much easier to solve. For example, we do not have to deal with occlusions.

Objects smoothly deform and move in connected paths. There is a lot of redundant information – size, shape, and relative positions – that carries over from one time slice to another. We track the largest structures (currently up to 50) and discard time slices which do not contain significant changes in the spatial character or distributions of the tracked structures. Conversely consecutive time slices that yield qualitatively different distributions are noted. These time slices are candidates for restart of numerical simulation to acquire better temporally resolved data.

Future extensions

We have only automated the extraction of kinematics and statistics of a few turbulence structures (vortices and pressure regions). Much more remains to be done:

- More structures. Ejection zones, dissipation regions, high stress areas, pockets, large-scale motions – these are some of the more important structures that need to be tracked.
- Spatial correlations. We need to have efficient ways to compute pairwise or even triple correlations of spatial distributions.
- Dynamics. A passive catalogue of distributions is not sufficient to determine a conceptual model that summarizes important dynamics and causal relationships among structures. We need to incorporate qualitative rules of interaction. For example, a vortex accelerates velocity (lifts) on one side and slows down on another. A bent vortex exerts a self-induced motion along its binormal and the effect is proportional to the curvature of the bent (Arms and Hama, 1965).

Experiment

As a testbed, we choose a direct numerical simulation[6] of free surface turbulence generated by a shear flow in a 128^3 rectangular box. The problem is important for several reasons. First, the mechanism for scalar transport into turbulent fluids across free-shear interfaces is important for the design and control of industrial equipments with free surfaces. Second, interest in environment and global warming raises the issue of how to accurately estimate mass transfer rate of CO_2 between atmosphere and ocean. Third, recent experiments have found evidence that low-speed streaks occur near free-slip surface under sufficient high shear. Visually these streaks resemble those found in wall-layer streaks. Without the complication of a solid boundary, studies in the formation of these streaks near free surface may shed additional light on the phenomenon.

It has been recently proposed that a definition of vortex in an incompressible flow in terms of the second eigenvalues of the symmetric tensor S^2 and Ω^2 where S and Ω are respectively the symmetric and antisymmetric parts of the velocity gradient tensor ∇u (Jeong and Hussain, 1995). The symmetric part determines the contribution to local pressure minimum due to rotational motion alone.

This proposed definition is attractive computationally because it refers to the isosurfaces of a scalar field, the second eigenvalues of the symmetric tensor. We compare this definition with our working definition of vortex in terms of compact bundles of vorticity lines.

[5] Mathematically the reorientation corresponds to the rotation of the surface points by the matrix of eigenvectors of the covariance matrix of the data points.

[6] A direct simulation proceeds from the Navier-Stokes equations without extraneous modeling assumptions.

We have examined dozens of structures according to both definitions. We found surprisingly substantial agreement between two definitions. Figure. 2 is a typical result. The figure shows a superposition of the vortex lines recovered by laborious integration and the surface patches quickly recovered by the extended marching cube algorithm (using a threshold of -0.35). Because of results like these, we now routinely use the scalar isosurfaces as surrogates for vortices to develop our matching and correlation algorithms.

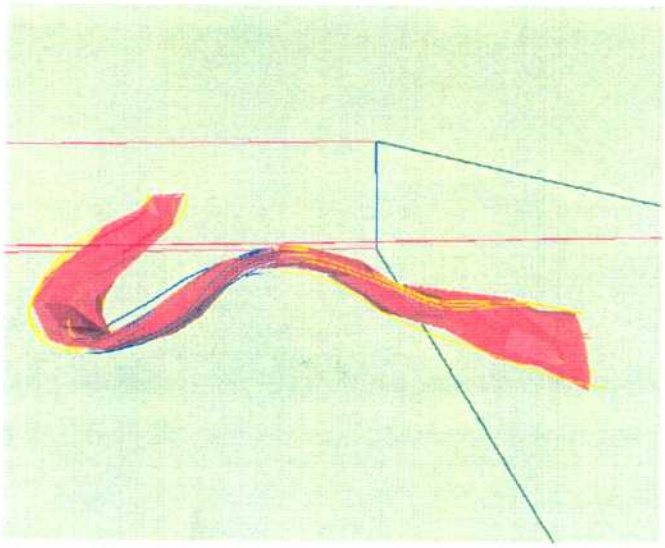

Figure 2: A superposition of vortex lines obtained by adaptive integration and the surface patches recovered by the marching cube algorithm. The agreement is surprisingly good.

Related Work

Our research shares some of the objectives with the visualization work in Rutgers (Silver and Zabusky, 1993). The Rutgers group is interested in better interactive tools for helping scientists visualizing the datasets. Our effort emphasizes more on building a computational model of the process by which scientists understand those data. We ask questions about how scientists think visually and how this visual thinking is connected to the rather primitive knowledge of object cohesiveness, persistence, and continuity.

Commonsense reasoning about fluids has been proposed as central problem in naive physics (Hayes, 1985b; Hayes, 1985a). The problem is hard because fluids do not conveniently divide into discrete pieces that can be easily combined. Ken Forbus and his group have extended and partially implemented Pat Hayes' ideas for representing fluids using both the contained-liquid and piece-of-stuff ontologies (Forbus, 1984; Collins and Forbus, 1987). The work described here is closer to the scientific end of the formalization spectrum.

Our work builds on previous work by several groups on imagistic reasoning and spatial aggregation (Yip and Zhao, 1996). These groups have designed problem solvers that achieve a high level of performance in many different domains: control and interpretation of numerical experiments (Nishida *et al.*, 1991; Zhao, 1994), kinematics analysis of mechanisms (Joskowicz and Sacks, 1991), design of controllers (Bradley and Zhao, 1993), analysis of seismic data (Junker and Braunschweig, 1995), and reasoning about fluid motion (Yip, 1995)

Discussion/Conclusion

We have described a framework for understanding flow fields and progress in implementation. The specific technical contributions are threefold:

- We show how aggregation algorithms (based on marching cube and adaptive vortex line tracing) and classification algorithms (based on recovery of generalized cylinders and superquadrics) can be used to extract and characterize interesting 3D fluid structures.
- In collaboration with the programs, we found substantial agreement between a recently proposed definition for vortex and the more intuitive definition of vortex as compact bundles.
- We extend the library of spatial aggregation routines by incorporating new aggregators and new classifiers.

We believe the current work also sheds light on the nature of human intelligence. Even in sophisticated scientific applications, some scientists appear to think visually and rely on commonsense intuitions of object cohesiveness, persistence, and continuity – not unlike those shown in recent cognitive psychology literature. This shared core of knowledge attests to the importance of recycling the peripheral machinery (such as vision) for solving otherwise symbolically-posed problems. Traditional AI has searched for the secrets of central intelligence for years. Perhaps the central is the peripherals.

References

Arms, R and Hama, F.R. 1965. Localized-induction concept on a curved vortex and motion of an elliptic vortex ring. *Physics of Fluids* 8.

Bajcsy, Ruzena and Solina, Franc 1987. Three-dimensional object representation revisited. *IEEE Proc. ICCV.*

Barr, A.H. 1984. Global and local deformations of solid primitives. *Computer Graphics* 18(3).

Binford, T.O. 1990. Generalized cylinder representation. In *Encyclopedia of Artificial Intelligence.* Wiley.

Bradley, E and Zhao, F 1993. Phase space control system design. *IEEE Control Systems Magazine* 13.

Collins, John and Forbus, Kenneth 1987. Reasoning about fluids via moecular collections. In *Proceedings AAAI-87.*

Forbus, Kenneth D 1984. Qualitative process theory. *Artificial Intelligence* 24.

Gupta, Alok and Bajcsy, Ruzena 1993. Volumetric segmentation of range images of 3d objects using superquadric models. *CVGIP: Image Understanding* 58.

Hayes, Patrick 1985a. Naive physics 1: Ontology for liquids. In Hobbs, J.R. and Moore, R.C., editors 1985a, *Formal Theories of the Commonsense World.* Ablex Publishing Corp.

Hayes, Patrick 1985b. The second naive physics manifesto. In Hobbs, J.R. and Moore, R.C., editors 1985b, *Formal Theories of the Commonsense World.* Ablex Publishing Corp.

Jeong, Jinhee and Hussain, Fazle 1995. On the identification of vortex. *Journal of Fluid Mechanics* 285.

Joskowicz, L and Sacks, E.P. 1991. Computational kinematics. *Artificial Intelligence* 51.

Junker, U and Braunschweig, B 1995. History-based interpretation of finite element simulations of seismic wave fields. In *Proceedings IJCAI-95.*

Lorensen, W.E and Cline, H.E. 1987. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics* 21(4).

Moin, P. and Kim, J. 1985. The structure of the vorticity field in turbulent channel flow. *Journal of Fluid Mechanics* 155.

Nishida, Toyoaki; Mizutani, Kenji; Kubota, Atsushi; and Doshita, Shuji 1991. Automated phase portrait analysis by integrating qualitative and quantitative analysis. In *Proceedings AAAI-91.*

Press, W.H.; Flannery, B.P.; and Vetterling, W.T. 1992. *Numerical Recipes in C.* Cambridge University Press.

Robinson, S.K. 1991. The kinematics of turbulent boundary layer structure. Tech. Mem. 103859, NASA.

Silver, D and Zabusky, N 1993. Quantitfying visualizations for reduced modeling in nonlinear sciences: extracting structures from the datasets. *Journal of Visual Communication and Image Representation* 4.

Spelke, E; Gutheil, Grant; and Walle, Gretchen-Van de 1995. The development of objet perception. In *Visual Cognition.* MIT Press.

Ullman, Shimon 1996. *High Level Vision.* MIT Press.

Yip, Kenneth and Zhao, Feng 1996. Spatial aggregation: Theory and application. *Journal of Artificial Intelligence Research.*

Yip, Kenneth 1995. Reasoning about fluid motion i: Finding structure. In *Proceedings IJCAI-95.*

Zhao, Feng 1994. Extracting and representing qualitative behvaiors of complex systems in phase space. *Artificial Intelligence* 69.