Facilitating the Visual Analysis of Large-Scale Unsteady Computational Fluid Dynamics Simulations

Kelly Gaither[1] and David S. Ebert[2]

[1] The University of Texas at Austin
kelly@tacc.utexas.edu
[2] Purdue University
ebertd@purdue.edu

1 Introduction

Computational power has increased dramatically over the past decade and has allowed computational fluid dynamics (CFD) researchers to more accurately simulate many types of flow with more detailed computational meshes. This increase has supported the calculation of highly detailed unsteady simulations that model real world conditions. However, this new power has yielded terabytes of data, and CFD researchers now face the very difficult task of trying to find, extract, and analyze important flow features buried within these monstrous data sets. Although being able to accurately model these problems is a major accomplishment in the world of CFD, it comes with the caveat that the amount of information that must be sifted through has now grown to enormous proportions that overwhelm investigators.

Unlike the explosive growth in computational power, visualization tools for these very large data sets have experienced a more modest evolution, and are not yet sufficiently developed to significantly aid the feature detection, extraction and analysis process. Most visualization tools require the user to work at a very cumbersome, low level with the data. Users are required to dice their data sets into appropriate data parameter ranges to search for the features of interest, since detailed visualization of such large data sets is impractical. Additionally, current visualization tools do not have the necessary capabilities to allow high-level exploration and analysis of these enormous datasets on desktop PCs, thus limiting their widespread use by scientists and designers. Previously developed systems for the interactive visualization of large CFD simulations [1, 2, 4], either work at a low-level with the data [1] or require high-speed network connections between the CFD simulation and the desktop visualization system and only work with a fixed set of basic features [2, 4].

The typical approach for visualizing large-scale CFD simulations is to precompute offline the visual representations that might be of relevance during the analysis process. This stage brings with it the hazard that often occurs when subsampling data. If the sample rate is not chosen with care, relevant features may be missed and the analysis to the solution may provide incorrect or misleading results.

This process can also be very time consuming and tedious requiring multiple passes into large stores of data without automatic methods for guiding placement and location of visualization samples.

CFD researchers desperately need new techniques that simplify and automate the iterative search and extraction process of finding the vital information in their data set. Improved visualization tools for CFD researchers could have a drastic impact on the design and safety of vehicles such as aircraft, ships, submarines, spacecraft, and automobiles. This community needs a new system that allows the users to articulate appropriate features of interest, provides a compact representation of these features which preserves their intrinsic qualities, and allows users to interactively visualize the feature information on a desktop computer. Such a system would also have to overcome the additional challenges of loading a sufficient portion of the data set into the available memory on a modest desktop machine, transferring these data sets over a network connection between the archive and local machine, mapping the entire data set to a visual representation in a reasonable amount of time, and rendering the results at interactive rates.

Therefore, we believe that a new visualization pipeline is needed for large-scale CFD visualization. As described in the following sections, our new system allows researchers to work at an intuitive, perceptually meaningful level with their data at their desktop. This new pipeline incorporates a flexible domain knowledge interface and a compact procedural encapsulation of large unstructured datasets to allow interactive exploration and analysis of CFD data.

2 Relevant Issues

Improving the current process for visualizing large-scale unsteady CFD data sets requires an understanding of the problematic issues that have dominated the current techniques.

2.1 Large Scale Data

The issue that gets the most attention is the size of the data sets. Often, a single time step can be larger than the memory capabilities of modern hardware. This is further complicated when the time dimension is being explored as well. It would be difficult to determine an average data size since this is entirely dependent on the resolution of the discretized volume in which the geometry is modeled and the solution computed. It would be fair to say that the size of the data set and the resolution with which it needs to be visualized increases with the complexity of the underlying physical problem being modeled.

2.2 Computational Grid Structures

The discretization, commonly called a computational grid can be generated in a variety of manners. The simplest of these cases is a voxelized grid that contains three-dimensional elements, or voxels, that are all the same shape and size. There is no need for additional adjacency information, as this is implied in the way the grid is stored. Another method for generating a computational grid is curvilinear, a grid generated in the dimensions I, J, and K holding six sided cells that have a non-linear shape function applied to the faces. This grid structure also has implied adjacency information, but the interpolation functions are more complex inside the cells and on the faces. A third type of grid structure that is commonly found in the computational simulation domain is the unstructured grid, typically a collection of tetrahedra, prisms, pyramids, and hexahedra that match at the faces. Because this structure is not stored with any adjacency information, this structure requires the explicit calculation of neighbor information when necessary.

2.3 Multivariate Data

The CFD domain has a base set of variables that are typically stored, from which all other parameters can be derived. Although the solutions are typically computed cell-centered, the solution parameters are often stored at the nodes of the computational grid. The solution values that are stored are the Q values: density, u component of velocity, v component of velocity, w component of velocity and energy. Additional variables such as pressure, temperature, and helicity can be derived from this base set of stored quantities.

2.4 Regions of Interest

Efficiently navigating and querying through these vast amounts of data requires some way of determining what is important and what is not. These regions of interest can be further refined into specific structures that the analyst would like to isolate and query. Current methods for detecting and tracking features are discussed in the next section.

2.5 Visualizing at Interactive Frame Rates

Interacting with the data is a key component to successful analysis. Prior to the introduction of a sophisticated Graphics Processing Unit (GPU V a single chip processor with integrated transform, lighting, triangle setup/clipping, and rendering engines capable of processing a minimum of 10 million polygons per second) [5], methods for displaying large-scale data at interactive frame rates required a sub-sampling of the data that was amenable for interactive display. With the maturation of highly parallel graphics hardware, we can now handle more complex computations that operate locally on the graphics chip. Additionally, combining compactly represented procedural encoding techniques to compress the data and the graphics processing unit to uncompress and visualize the results gives us much more flexibility in the sophistication of the displays.

3 Feature Detection and Feature Tracking

Current feature detection techniques consist primarily of analytical methods for identifying cells within the computational grid that meet some functional criteria and methods that track the formation and dissipation of cells that have been pre-defined to contain feature characteristics.

Analytical feature detection began as early as 1875 when Poincare made the connections between differential equations and the topology of vector fields. A historical survey can be found in [6]. One of the first applications of Poincare's theory of differential equations to three dimensional fluid flows was done by Lighthill [7]. In [8], this approach was extended to include a more general class of flows. Automatic techniques for identifying the topology in simulated flow fields was first introduced by Helman and Hesselink [10–12] and Shirayama and Kuwahara [9]. Visualizing this topology to understand the underlying physical flow structure was introduced by Globus, Levit and Lasinski [13]. They presented methods for calculating and classifying three dimensional flow structures such as sources, sinks, and saddles by computing the eigenvalues of the Jacobian matrix. Since then, analytical methods have been used to detect structures such as vortices [14, 15], shocks [3, 16], wakes [17], and recirculation regions [18]. The current techniques allow little if any flexibility for defining features through a higher level language, and these methods give little insight into the behavior of features over a period of time. Examples of these kinds of analytically described features can be seen in Figs. 1–4.

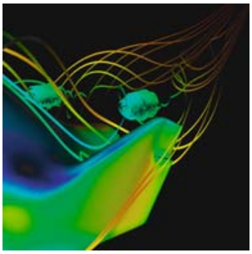

Figure 1: Vortices Over X38

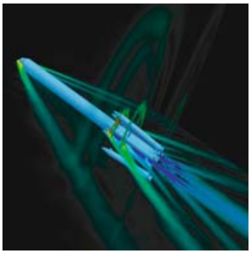

Figure 2: Shock Rings Around DeltaII

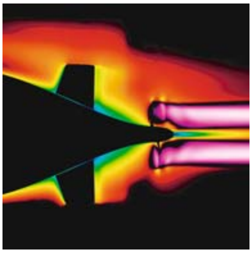

Figure 3: Wake from Propellers

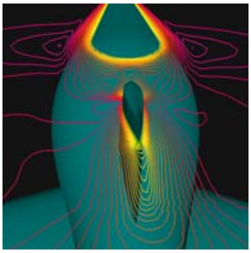

Figure 4: Recirculation Region

Figure 1 illustrates a vortex tube that forms from the air flow over an X38 crew recovery configuration computed at a 30 degree angle of attack. This figure displays the most common method for displaying vortices–particle traces. The traces in Fig. 1 are colored with pressure values.

Shown in Fig. 2 is an example of shock rings that form around a Delta II configuration during the transition from subsonic transport to hypersonic transport. The shock rings are shown by isolating density values contoured on cutting planes computed orthogonal to the body. These values are highlighted by blending out all other values with the use of transparency.

Figure 3 displays the wake region that forms during a hydrodynamic simulation of a submarine during a stopping and turning maneuver. This wake region is generated from the turbulence shed off of the propeller and has been found to be the primary cause of noise and drag during live maneuvers.

A recirculation region generated from flow around the rudder and fins of a submarine is shown in Fig. 4. This region can result in an area of suction and cavitations causing significant drag on the body during maneuvers.

Understanding the added time dimension as it relates to the physical problem is a crucial part of analyzing the solution, both for verification and for integration into complex physical design. Work has been done to develop a full 3D volume feature extraction and tracking algorithm that tracks thresholded, connected regions [19, 20]. This work is responsible for an initial classification of events: Continuation (an object continues, it may shrink or grow), Creation (a new object appears), Dissipation (an object dies), Bifurcation (an object splits into two or more objects), and Amalgamation (two or more objects merge). These features are classified by tracking properties of connected regions such as mass, moment and volume changes. This method looks at the next time step as an indicator of what type of event is occurring. This method, however, suffers in practice because it has no method for handling the periodic Dissipation and Creation of events such as tip vortices that are created and die in periodic intervals. Additionally, there is no scheme for giving the complete information for features that behave non- deterministically.

An example of the types of composite features that are of interest is shown in Fig. 5. This figure (courtesy of David Marcum, Mississippi State University) shows a set of potential features that exist on a finned missile configuration. In viscous simulations, the boundary layer contains a significant amount of important information and is the primary indicator for transition from laminar to turbulent flow. The shock wave is found as a connected or nearly connected set of cells containing a discontinuity. The expansion fan gives an indication of an expansion shock that occurs because of the movement of the fin. The wake sheet is of critical importance when studying both noise and vibration as both can result in an unstable transition. Finally, vortices are of critical importance when studying noise and instability.

Although each of these features can be individually identified by analytical techniques, we are often interested in the relationships between features and the given geometries and between features of similar and differing types. To properly identify this behavior, we must turn to a combination of analytical techniques, statistical properties of the volumetric domain, and heuristics applied during the analysis. This allows us to work at a higher level, producing a hierarchy of decisions that together form the conclusions about features, such as existence, persistence, coherence, and correlation to other features.

Working at this higher level is critical to the ability to identify features that are neither well understood nor well defined through analytical descriptions. For example, lambda shocks (Fig. 6) are well understood in theoretical aerodynamics, but are not well described analytically. The general shape is known, but the orientation, strength and persistence of this type of feature requires a significant amount of heuristics to properly identify. The lambda shock in Fig. 6 was captured through a series of shaded cutting planes sliced orthogonal to the wing. This lambda shock is shown using density to contour the cutting planes. Thresholding was used to make uninteresting regions transparent.

Understanding the correlation and intermingling of features is necessary to properly study events such as mixing. Figure 7 illustrates one time step in a sequence of time steps that demonstrate one solute being mixed over time with another.

Turbulent flow presents unique challenges. The scale of features in a turbulent boundary layer is at a very fine scale, typically orders of magnitude more precise than laminar boundary layers. This precision is necessary to capture all of the small scale features that contribute to turbulent behavior. Additionally, we face the problem that the features that occur in a turbulent boundary layer are not all well defined. Given the proper scale, we can easily find vortical structures or eddies. It is significantly harder to analytically describe seemingly random behavior such as that shown in Fig. 8, showing turbulent channel flow. We can easily see that features of interest exist in the flow by looking at cutting planes contoured with velocity magnitude. However, these features are very difficult to describe analytically. Therefore, it is critical to add statistically based information and heuristics to properly identify the features of interest.

The issue becomes even more complex when trying to understand and analyze the features present in complex flow of complex geometries, such as those attempting to simulate real-world conditions. Figure 9 shows four views of a Delta II configuration with strap on boosters. These boosters are used to propel the Delta II into space, but drop off shortly after. It is the peeling off of these boosters that can cause very complex structures and instabilities to form that can affect the viability of the Delta II in flight. Understanding these structures is the key to being able to compensate and adjust for them. However, this is a process that is not well understood at all from the feature detection community. It is here that we must implement and deploy our hierarchy of features, giving us the ability to detect features that we suspect are present and those that we do not know are present. Additionally, understanding the interplay of these features allows us to understand the causal relationship of possible instabilities, thus reducing the possibility of vehicular failure.

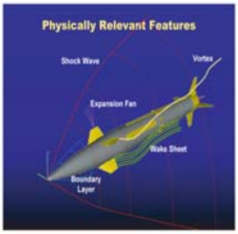

Figure 5: Finned Missile Features

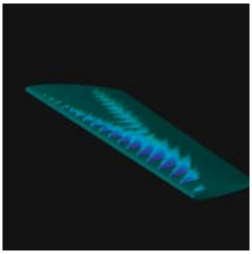
Figure 6: Lambda Shock

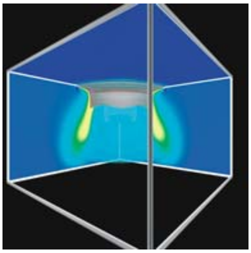
Figure 7: Mixing Solutes

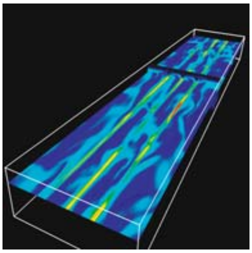
Figure 8: Turbulent Channel Flow

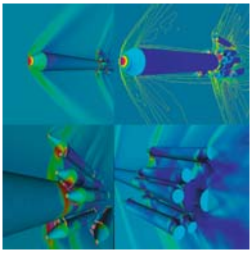
Figure 9: DeltaII Features

4 A New Visualization Pipeline for Large-Scale CFD Simulations

It is clear that the existing methods for visualizing these large-scale data sets need improvement. To this end, we are solving these CFD Visualization problems by developing techniques for creating a procedural abstraction for very large data sets, developing effective and efficient methods for mapping from the procedural to visual representation, and applying these techniques to the problem of visualizing large CFD simulations. Our new methods provide interactive visualization of very large data sets on a desktop computer, and will scale gracefully across a range of computer power and bandwidth situations. These new methods can be summarized in a new pipeline for visualizing large-scale CFD data sets. This new pipeline can be adapted easily to any sequences of large data sets whether measured or simulated. This pipeline operates in the following manner:

4.1 Detect Candidate Features

General characteristics of a given data set can be found by applying a set of statistically based methods. This gives the opportunity to find properties such as connected cells, including 1 nearest neighbor, 2 nearest neighbor, etc, persistence, energy, entropy, and area. These are useful in determining areas of potential features or to detect candidate cells that may contain features that can be found by a combination of analytical techniques and domain dependent heuristics.

4.2 Further Refine Features

These candidate cells can be further refined by applying a combination of analytical techniques with procedural descriptions of domain-dependent heuristics. For example, we consider every feature to be a reaction to an action. If the problem that is being solved is the physics surrounding a geometric body, then a feature cannot exist without having some causal relationship to either the geometry or to other features.

4.3 Procedurally Encode Features

To facilitate the visualization of these features on a desktop PC, we represent these features using a procedural representation of implicit models based on radial basis functions.

4.4 Apply Multiresolution Techniques

We can then adapt the procedural representation to the appropriate level of detail using multi-resolution techniques.

4.5 Incorporate Knowledge Gained into Metadata

The feature information is encapsulated as domain specific knowledge in the metadata. This allows us to explore these extremely large data sets both at the feature level and, more importantly, at the higher level of relationships among features, including feature to feature relationships and feature to geometry relationships.

4.6 Visualize Directly From the Procedural Representation Using the GPU When Possible

We then visualize the data directly from the procedural representation, using and extending numerous familiar CFD visualization techniques (e.g. cutting planes, isosurfaces, volume splatting, direct volume rendering, particle clouds, streams, rakes, line-integral convolutions, and glyphs).

4.7 Verify Accuracy

Accuracy is a key component to the viability of this pipeline. Although it is not currently standard practice to measure the accuracy of the CFD simulations, it is necessary to ensure that the procedural is accurate to some pre- define threshold. For our work, we have defined an acceptable tolerance to be 5communicating with researchers in the CFD domain. We are facilitating this by careful tracking of approximation error throughout the entire process, including scanning, modeling, reconstruction and visualization.

5 Conclusion

Our system containing the new visualization pipeline will allow CFD researchers to work more effectively by interactively exploring their data to pinpoint the features of interest. This will facilitate and allow them to adaptively refine their solution grids within those areas, and iteratively compute solutions on these adapted grids to improve their simulation and solve the underlying flow problem more quickly. Moreover, the results of this project will provide solutions not only for CFD researchers, but also for a wide variety of visualization challenges and applications. Our main goal is to develop techniques that allow visualization exploration, feature detection, extraction and analysis at a higher, more effective level through the use of procedural data abstraction and representation.

Acknowledgements

This work is supported by the National Science Foundation under grant NSF ACI-0121288. The authors would like to thank and acknowledge those responsible for computing the simulations that produced the data in the Figs. 1–9. The data for Figs. 1–6 and Fig. 9 was generated at the Simulation and Design Center, Mississippi State University. The data for Fig. 7 was generated at the Computational Fluid Dynamics Laboratory at The University of Texas at Austin. The data for Fig. 8 was generated by Dr. David Goldstein's laboratory at The University of Texas at Austin.

References

1. Freitag, L., Loy, R.: Using Desktop Graphics Workstations for Interactive Remote Exploration of Large Data Sets. Proceedings Using Desktop Graphics Workstations for Interactive Remote Exploration of Large Data Sets, (2000)
2. Haimes, R.: Pv3: A Distributed System for Large-Scale Unsteady CFD Visualization, AIAA 94-0321, (1994)
3. Lovely, D., Haimes, R.: Shock Detection from Computational Fluid Dynamics Results, AIAA 99-3285, (1999)

4. Haimes, R., Kenwright, D.: On the Velocity Gradient Tensor and Fluid Feature Extraction, AIAA 99-3289, (1999)
5. NVIDIA: Graphics Processing Unit (GPU) at http://www.nvidia.com/object/gpu.html (1999)
6. Abraham, R. Shaw, C.: Dynamics: The Geometry of Behavior, parts 1-4, Ariel Press, Santa Cruz, CA, (1984)
7. Lighthill, M.: Attachment and Separation in Three Dimensional Flow, Laminar Boundary Layers II, ed. L. Rosenhead, pp. 72-82, Oxford University Press (1963)
8. Perry, A. Fairly, B.: Critical Points in Flow Patterns, Advances in Geophysics, 18(B), pp. 299- 315, (1974)
9. Shirayama, S. Kuwahara, K.: Flow Past a Sphere: Topological Transitions of the Vorticity Field, AIAA-90-3105-CP (1990)
10. Hesselink, L., Helman, J.: Evaluation of Flow Topology from Numerical Data, AIAA-87-1181 (1987)
11. Helman, J., Hesselink, L.: Representation and Display of Vector Field Topology in Fluid Flow Data Sets, IEEE Computer, pp. 27-36 (1989)
12. Helman, J., Hesselink, L.: Surface Representation of Two- and Three-Dimensional Fluid Flow Topology, Proceedings of IEEE Visualization 1990 (1990)
13. Globus, A., Levit, C., Lasinski, T.: A Tool for Visualizing the Topology of Three-Dimensional Vector Fields, Proceedings of IEEE Visualization 1991 (1991)
14. Banks, D., Singer, B.: Extracting Vortices From an Unsteady Flow, Proceedings of IEEE Visualization 1994 (1994)
15. Sujudi, D., Haimes, R.: Identification of Swirling Flow in 3-D Vector Fields, AIAA 95-1715, (1995)
16. Sobieczky, H., Hannemann, M.: Computational Shock and Mach Waves Visualization Aiding the Development of Aerodynamic Design Techniques, Proceedings of the 21st International Symposium on Shock Waves, (1997)
17. Haimes, R.: Automated Feature Extraction from Transient CFD Simulations, Proceedings of the 7th Annual Conference of the CFD Society of Canada, Keynote Address. Halifax, NS, May (1999)
18. Haimes, R.: Using Residence Time for the Extraction of Recirculation Regions, AIAA 99-3291 (1999)
19. Silver, D. Wang, X.: Volume Tracking, IEEE Transactions on Visualization and Computer Graphics, 3(2), June (1997)
20. Silver, D., Wang, X.: Tracking Scalar Features in Unstructured Datasets, Proceedings of IEEE Visualization 1998 (1998)