# Visualization in Computational Fluid Dynamics: A Case Study

Robert Haimes and Dave Darmofal
Computational Fluid Dynamics Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology, Cambridge, MA 02139

## Abstract

*VISUAL3 is a highly-interactive environment for the visualization of 3D volumetric scientific data. The volume may be broken up in a structured or unstructured manner. Also, the problem may be static or unsteady in time. Because the data is volumetric (not just surfaces), and all the information can be changing, traditional CAD techniques are not appropriate. Therefore VISUAL3 was developed using immediate mode rendering methods.*

*A unique aspect of VISUAL3 is the dimensional windowing approach coupled with cursor mapping which allows efficient pointing in 3D space.*

*VISUAL3 is composed of a large number of visualization tools that can be generally classified into identification, scanning and probing techniques.*

## 1 Introduction

The data produced by Computational Fluid Dynamics (CFD) codes creates many problems for visualization systems. The data is volumetric; in other words, the information of interest is throughout a volume and not just on a surface. This volume may be broken up in many different ways. The simplest form of discretization is to cut the volume into a structured set of hexahedral cells. On the other extreme the volume may be broken into tetrahedra in an irregular manner producing an "unstructured" grid. CFD applications can also produce very large data-sets. For example, it is not uncommon to perform a full airplane calculation with over a million cells. Also, researchers are now looking at 3D CFD calculations that are time varying! Traditional visualization techniques based on CAD output only provide surface information and cannot give the CFD investigator much help in quickly understanding the results of these codes.

Within the last couple of years graphics workstations have advanced to the point where the high-end workstations, due to their speed in both "number-crunching" and 3D drawing capabilities, are appropriate for interactive volumetric scientific visualization. VISUAL3[1][2] was written to take advantage of these workstations in tackling the problems of investigating the results from CFD calculations including those that are unsteady in time.

VISUAL3 is not specifically a CFD viewer. Any volumetric problem that has been discretized is appropriate, infact VISUAL3 deals with generic scalar and vector fields.

## 2 Motivation for Creating VISUAL3

The work on VISUAL3 began because there was no hardware/software combination on the market that would satisify the post-processing needs of the CFD Lab at MIT. The baseline requirements for VISUAL3 were the ability to handle both structured and unstructured grids, the display of unsteady data, the ease of use by students, and above all high interactivity.

During Computer-Aided Education software development for undergraduates at M.I.T.'s Project Athena it became clear the interactivity was of paramount importance in conveying information to the user[3]. Having to wait for results of a user action breaks the train of thought and can be frustrating. In order for a visualization package to be a fruitful tool it must be able to answer questions with a minimum delay in time.

At M.I.T., we have the luxury of not having to sell our software in order to survive. Therefore, VISUAL3 was targetted at the then current high-end hardware (e.g. Stardent's GS series) to satisify the Lab's needs at the expense of market share. With how quickly the workstation capabilities are improving VISUAL3 is being ported to mid-price range machines.

Assuming the target platform has both good compute power and rendering abilities has ultimately allowed research into new techniques while maintaining high performance and interactivity. For example, a new efficient method was developed for cutting through the volume (either iso-surface or planar cut) that does not require voxels or "marching cubes"[4] since neither of these methods is appropriate for unstructured meshes.

In general, the scientific visualization software available seems to be aimed more at generating pictures for presentation than at understanding of fluid dynamics. An important design goal for VISUAL3 was that it be a tool in the investigation and interpretation of computational results. When this goal is met, a side benefit is that with better viewing and understanding of the flow results, the visualization software can be used for flow solver debugging. For example, unlike packages that generate "pretty" pictures, VISUAL3 performs all shading in a facetted manner. This is part of a design philosophy to give the investigator only the data. Phong shading, or anything that may smooth out blemishes (techniques routinely used in CAD) are always avoided.

An important design goal for VISUAL3 was extendability [2]. A programming interface was designed to allow the use of structured and unstructured gridding as well as unsteady data. The programmer may also define as many scalar and vector fields as necessary. This has the drawback of requiring some programming but provides a great deal of flexibility in handling any type of volumetric data.

## 3 Immediate Mode

The only 3D viewing software standard currently available is PHIGS. Using PHIGS for VISUAL3 development would have had the obvious advantage of portability but PHIGS has some significant drawbacks. Many current PHIGS implementations support only one window, but VISUAL3's design requires a multi-window approach. PHIGS is designed for CAD applications where the complete object is pre-defined and any movements or changes to the object are done by local transforms. This allows one part of the object to move in respect to another. In VISUAL3, using scanning tools, surfaces are always being created and destroyed. In an unsteady application all the data changes from one time to the next. In fact, changing the scalar color mapping in PHIGS requires that the whole object be rebuilt since color values are a part of the PHIGS data structure. Furthermore, most aspects of CFD visualization do not rely upon an object-oriented rendering which is one of the advantages of a PHIGS style standard. As a result of these problems, PHIGS was rejected for the implementation of VISUAL3.

VISUAL3 was developed using Stardent's XFDI extension to the X window system. XFDI provides immediate mode 3D rendering support enhancements to X by an X/XFDI server. Therefore most of VISUAL3 is an X application with a few non-standard internal calls to perform Gouraud shading for surfaces and 3D support for drawing lines and triangles with hidden surface removal. This is very much the same model as the current PEX standard.

It is unfortunate that the only standard Application Programming Interface (API) into PEX is currently PHIGS leaving no way to get to the server's immediate mode capabilities. VISUAL3 has been ported to a DECstation 5000 using DEC's PEXlib (a library of bindings that build PEX packets to send to the server analogous to Xlib for straight X servers) and a Stardent V800 using Stardent's PEXIM (a library supplying immediated mode 'hooks' to PHIGS). It is most important that some acceptable alternative API that supports immediate mode applications be agreed upon so that packages that perform scientific visualization can have the same kind of portability as other software!

## 4 Multiple Windows

### 4.1 A Dimensional Approach

An extremely important aspect of CFD visualization is the ability to simultaneously view data in multiple ways. Thus, a multi-windowing system is an essential part of VISUAL3's design. In particular, the ability to look at data in one-, two-, and three-dimensions simultaneously –dimensional windowing– is extremely important. Many visualization software packages offer the ability to view an object from different positions simultaneously. We have found that rotation and translation of an object can fulfill the same purposes and reduce screen clutter. VISUAL3's window set-up (see Figure 1 for layout) is designed with a dimensional approach in mind. The three main plotting windows are:

- 3D Window

  The 3D window displays data on three-dimensional surfaces, either from the bounding domain or from cuts and iso-surfaces. The user has complete control over the rendering attributes of each individual surface. The 3D window also displays three-dimensional lines such as tufts and streamlines. The view of the 3D object can be rotated, translated and enlarged using the dialbox.

- 2D Window

  The 2D window is used to display data on a mapped domain or cut surfaces (for which there exists a mapping to an $(x', y')$ coordinate system). The user can independently adjust the view in this window. The 2D window is used to initiate particle paths, perform many of the probes and point in 3-space.

- 1D Window

  The 1D window is used to display one dimensional data which is generated by various probing functions in the 2D window or from mapping streamlines.

The reduction of dimensions from each window can significantly aid in the understanding of a flow field. Often, patterns are much easier to recognize on two-dimensional planar cuts through a domain than in three-dimensions. Furthermore, information on a two-dimensional surface may be reduced to one dimension. For example, the user may interactively position a line in the two-dimensional window producing a graph in the one-dimensional window of the current scalar field along the line verses the line length.

### 4.2 Cursor Mapping

Another important concept related to the dimensional windowing approach is cursor mapping. Cursor mapping is the simultaneous display of the current cursor position in the one-, two-, and three-dimensional views. For example, suppose a line plot is displayed in the 1D window. When pointing (with the cursor) in the 1D window, the user is actually pointing in 2-space by reversing the 2D to 1D mapping. Similarly, this point in 2-space corresponds to a point in 3-space. This is a simple and powerful concept. Being able to point at and locate unusual features in the volume (and accessing VISUAL3's tools) the user can begin to answer questions like; is this a numerical anomaly or some interesting feature of the flow?

This cursor mapping is also extremely useful when used with streamlines. The calculation of a streamline requires an initial 3-space location to begin numerical integration. With structured grids, this is often done by simply selecting a specific computational grid line or node. However, with unstructured grids, this technique is impossible. With cursor mapping, the user simply points to a location in the two-dimensional window which corresponds to a unique 3-space point from which a streamline may be spawned. Without a multi-window system pointing in 3D is, at best, difficult.

# 5 Visualization Tools

Besides multiple windows, we believe that having multiple visualization tools is also extremely important. Multiple tools allow the user to view the same data in many ways, often providing additional insight. In general, the ideas for these tools have been stimulated by the decades of work done in experimental flow visualization and user feedback. Consequently, VISUAL3 is constantly gaining new capabilities.

During the development of VISUAL3, a wide range of visualization tools were implemented. No tool has ever been removed from VISUAL3; however, a tool may not be included if it is not interactive. Thus, the user has a broad pallette of tools for general investigation. In some cases not all techniques are useful, the effectiveness of a tool depends on what the user is looking for and/or attempting to understand. This requires some expertise from the user in that he/she must understand how to use the tool as well as have some underlying understanding of the physical problem.

Visualization tools may be divided hierarchically according to their abilities[5]. The main tasks during visualization of a CFD data set are feature identification, scanning, and probing. Identification techniques help the user to locate flow features over the entire domain and can save a great deal of effort and time during visualization. Scanning techniques allow the user to interactively search the domain by varying a single parameter. For example, the cutting plane, a two-dimensional surface slice through the domain, is a scanning technique because its position may be changed along its normal. Thus, the plane may be scanned through the domain. Finally, probing techniques are very localized visualization tools. A point probe which returns the scalar value from the current cursor position is a probing technique. This breakdown of visualization methods is in some sense simply a reduction of dimensions. In other words, identification techniques are three-dimensional, scanning techniques are two-dimensional, and probing are one-dimensional. As with multiple dimensional windows, we again stress the importance of reduction of dimensions as an integral part to effective and efficient visualization. The remainder of this section briefly describes some of the various tools we have implemented in VISUAL3. A more complete description of the tools and algorithm issues may be found in [1][2][5].

## 5.1 Identification Tools

Probably the weakest link currently in visualization is the ability to quickly identify features. Although some work has been done, a great deal of improvement can still be made in useful feature identification algorithms. Currently, VISUAL3 supports vector clouds for feature identification purposes. Vector clouds display the local velocity field in areas which have been thresholded to display only the interesting portions of the flow. In the vortical flow over a delta wing, total pressure loss is high in the vortices; thus, thresholding away the low total pressure regions should result in an image showing the vortical regions of the flow. Figure 2 is an example of vector clouds using total pressure loss as the scalar threshold variable in a transonic delta wing flow[6]. The primary and secondary vortices can be clearly seen in the image.

We are also researching other possible feature extraction methods. For example, a discontinuity locater is being developed to locate the positions of shocks within a flow field. See Section 5.2.2 for more information on shock detection. Recently, an X-ray feature identification algorithm was also implemented[7]. Rays are sent through a computational domain and the ray intensity is decreased according to the scalar field values it travels through. However, the X-ray technique requires essentially a ray tracing algorithm which is usually slow and hampers interactivity. Also, like the medical use of X-rays images, the results are some times confusing.

## 5.2 Scanning Tools

Scanning techniques, although still relatively recent developments, are much more established than feature identification techniques. This set of methods allows the researcher to scan through the domain either through space or through scalar and vector field values. We discuss some of the more widely used scanning techniques below.

### 5.2.1 Cutting Planes

Cutting planes are planar surfaces which slice through the computational domain. The scalar field values at the nodes of the cut cells are interpolated onto the planar surface and then rendered in two-dimensions. The plane may then be scanned in its normal direction allowing the user to investigate the entire domain if necessary. The plane normal direction, location, and size may be interactively set by inputs to the dialbox. A very efficient algorithm for unstructured grid cutting planes has been successfully implemented by Giles and Haimes[1]. We again use the transonic delta wing data set to show the results of a cutting plane in all windows (Figure 1). The line probe is also active in the 1D window.

### 5.2.2 Iso-Surfaces

Iso-surfaces are three-dimensional surfaces of a constant scalar value. In reality, cutting planes are simply the geometric subset of iso-surfaces where the constant value parameter is a geometric coordinate. Iso-surfaces are extremely useful for two purposes. First, the scanning of iso-surfaces is an effective method for visualizing three-dimensional scalar gradient fields since each consecutive surface represents a small change in a scalar from the previous surface. Thus, in areas where gradients are locally high, consecutive iso-surfaces will essentially maintain the same shape. Similarly, in areas where the gradients are low, consecutive iso-surfaces will displace considerably. Second, iso-surfaces may be useful as a feature identification technique. If a flow feature may be identified by a unique scalar value, then an iso-surface of that value will show the desired feature. A shock finding algorithm has been implemented by constructing an iso-surface where the Mach number normal to the shock surface is 1.0[5]. Figure 3 shows the shock surfaces in a hypersonic delta wing calculation by Lee[8].

#### 5.2.3 Tufts

Tufts are small vectors whose direction and size represent the vector field, which is typically the velocity vector. A regular grid of points on the cutting plane defines where the tufts are displayed. The projection of the tufts appear in the two-dimesnional window. The three-dimensional view shows the tufts in their true position. A slight variation places arrows only at the intersection of the cutting plane and a computational cell edge. This option limits errors arising from interpolation.

Since tufts are defined on cutting planes which may be scanned, they have been classified as a scanning technique. The tufts in Figure 4 are seen in both the two-dimensional and three-dimensional windows. The tufts highlight the vortical nature of the flow around the wing.

### 5.3 Probing Tools

Probing methods, the final type of visualization technique, provide the most localized information. These tools are primarily used in the final step of investigating a flow feature to gather quantitative information such as the pressure distribution on a surface, or velocity profiles in boundary layers. Typically, these probes mimic the action of common experimental techniques and, therefore, are often used in comparing calculations to experiments. We illustrate just a few of these probes below.

#### 5.3.1 Pathlines

A pathline is a line through the flow field which is everywhere tangent to the local vector field. If the vector field is the velocity field, then the pathline is termed a streamline. In a steady flow, a streamline is the path a particle follows when released in the flow. The results from three streamlines may be seen in Figure 5.

Spawning off a pathline is a simple procedure. VISUAL3's ability to position in 3-space allows a 'point and shoot' procedure with an absolute minimum of user interaction. Variants allow a line, circle and grid of pathlines as well as an interactive streamline (a 'streamer') initiated from the 2D window. The streamer follows the cursor when located in the two-dimension view and provides fine probing of a vector field. In fact, the scanning technique 'tufts' in concert with the 'streamer' allows quick and effective understanding of vector fields.

#### 5.3.2 Tubes

Tubes are pathlines with circular cross-sectional areas based upon the local crossflow divergence. Thus, the thickness of streamtube cross-sections represents the local expansion of the flow field. Besides providing information about the local expansion of the flow, tubes are much easier to understand than regular pathlines because of the additional thickness. The intertwining of tubes can be visualized; with pathlines, the relative positions of several lines is hard to determine from a three-dimensional view. The streamtubes in Figure 5 are at the same position as the streamlines from above.

#### 5.3.3 Point Probe

The point probe has two major functions: the local scalar value is displayed in a text window, and the local vector is projected on the two-dimensional plane. As a result of the cursor mapping function discussed earlier, this also displays the local vector in the three-dimensional window. This vector is a single 'tuft' eminating from the current cursor position.

#### 5.3.4 Boundary Layer Probe

The boundary layer probe plots the variation of a scalar along a direction normal to a boundary surface. The normal may be interactively positioned anywhere on the boundary surface by moving the cursor in the 2-D window. As the name indicates, this probe is effective for, investigating boundary layer effects.

#### 5.3.5 Pathline Probe

The pathline probe displays the variation of a scalar along a pathline. The cursor mapping ability allows the pathline probe to pinpoint the exact location of unusual pathwise behaviors. An additional feature is the ability to use the pathline probe to position a cutting plane. When the probe is on, the cutting plane may be placed at any location on the current pathline with the plane normal being the local path-direction at the cursor location. Using this tool, a cutting plane can be easily placed at a shock or any other interesting feature along the pathline.

## 6 Conclusions

The lack of adequate scientific visualization software for volumetric data has led us to the in-house development of VISUAL3 – an unstructured, unsteady, three-dimensional visualization package. During the design, careful attention was paid to high interactivity and the understanding of fluid dynamics rather than the production of high-quality pictures. VISUAL3's portability is only limited by the fact that there is no current standard for an immediate mode API to PEX.

The use of multiple windows as a reduction of dimensions is very helpful in the comprehension of results. Furthermore, a unique cursor mapping technique allows efficient pointing to a location in 3D space. VISUAL3 also supports a large variety of visualization tools. In particular, visualization tools which automatically locate flow features will be enhanced. These identification techniques can save large amounts of time by detecting flow features without the need to painstakingly scan the entire domain.

## Acknowledgements

The authors wish to thank the VISUAL3 user base (for their fine sugestions and comments) and the two people without whom this work would not be possible: Professor Mike Giles and Professor Earll Murman. Their assistance and ideas have been invaluable to the project and this paper. Dave Darmofal would like to thank the DOD NDSEG Fellow program for their support of his research and education. This work was partially supported by AFOSR under grant 89-0395 monitored by Dr. Len Sakell.

## References

[1] M. Giles and R. Haimes. Advanced Interactive Visualization for CFD. *Computing Systems in Engineering*, 1(1):51–62, 1990.

[2] R. Haimes and M. Giles. VISUAL3: Interactive Unsteady Unstructured 3D Visualization. AIAA Paper 91-0794, 1991.

[3] E. M. Murman, A. R. LaVin, and S. C. Ellis. Enhancing Fluid Machanics Education with Workstation Based Software. AIAA Paper 88-0001, 1988.

[4] W. E. Lorensen and H. E. Cline. Marching Cubes in a High Resolution 3D Surface Construction Algorithm. *ACM Computer Graphics*, Vol. 21(4):pp. 163–169, 1987.

[5] D. Darmofal. Hierarchal Visualization of Three-Dimensional Vortical Flow Calculations. Master's thesis, M.I.T., March 1991.

[6] T. Becker. Hybrid 3-D Euler/Navier-Stokes Calculations of Transonic Vortex Flows over the NTF Delta Wing. Master's thesis, M.I.T., August 1989.

[7] D. Darmofal and R. Haimes. Visual Feature Identification for 3-D Data Sets. AIAA Paper 91-1583, 1991.

[8] K.-M. Lee. Numerical Simulation of Hypersonic Flow Over a Blunt Leading Edge Delta Wing. Master's thesis, M.I.T., October 1989.

Figure 1: Cutting plane example - full screen view

Figure 2: Total Pressure Loss Vector Clouds

Figure 3: Shock surfaces rendered by shock strength

Figure 4: Tufts example - full screen view

Figure 5: Comparison of streamlines and tubes