Methods for Interactive Visualization of Large Flow Data Sets

Wim de Leeuw and Robert van Liere

Center for Mathematics and Computer Science CWI,
P.O. Box 94097, 1090 GB Amsterdam, Netherlands

ABSTRACT

Modern computational fluid dynamics simulations are capable of the detailed simulation of fluid flow. The output data sets of these simulations are very large and information rich. The importance of data visualization is clearly recognized for the presentation of these data sets. For gaining new insight in the nature of flow, interactive visualization methods are essential. The goal of our work is to develop a environment which allows fluid dynamics experts to analyze very large flow fields. This paper motivates the need for an interactive visualization environment. The design of the environment is centered around two fundamental beliefs: First, the environment should integrate modeling and visualization. Second, interactive visualization is essential so that exploration is stimulated. We discuss two interactive visualization methods and their application to a flow field resulting from a direct numerical simulation. We believe that the working methods discussed in this paper will be typical of future visualization environments.

Keywords: interactive scientific visualization, flow visualization, high performance computing, direct numerical simulation

1. INTRODUCTION

The importance of data visualization is clearly recognized in scientific computing. Display of simulation results, exploration of large data sets, and interactive steering of computation all require some component of visualization. However, the demands imposed by modern large scale computational fluid dynamics simulations severely test the limits of today's visualization systems.[1] Computational fluid dynamics(CFD) simulations are capable of accurate simulation of fluid flow. The resulting data sets of such simulations are very large and information rich. This trend will continue as solutions to problems at higher Reynolds numbers, and therefore higher resolutions, are desired. Solving these problems will be possible as computers become more powerful and numerical solvers improve. Effective visualization methods are needed for the analysis of these large data sets.

For example, consider figure 5. The data is a slice of a 3D data set from a direct numerical simulation of turbulent flow. The image shows flow around a square cylinder; one can clearly see vortex shedding behind the cylinder. Fluid dynamics experts would like to obtain a detailed understanding of the transition from laminar to turbulent flow. Traditional visualization methods are not well suited to analyze these phenomena from such large and detailed data sets. More advanced visualization and data management methods are required.

The goal of our work is to develop a visualization environment which allows fluid dynamics experts to analyze very large flow fields. The design of the environment is centered around two fundamental beliefs: First, the environment should integrate modeling and visualization. Ultimately, we envision that visualization will be an integral part in the process of modeling complex flow phenomena. Second, for exploration of these data sets, interactive visualization is essential. Interaction is necessary in cases in which it is not a priori known which phenomena are of interest. The implementation of the environment touches upon many aspects of high performance computing, including advanced support for data presentation and navigation techniques, efficient data management, high bandwidth parallel I/O, scalable and distributed visualization techniques, and the incorporation of novel display technology.

This paper motivates and illustrates how our visualization environment has been applied to the analysis of a flow field resulting from direct numerical simulation. We believe that the working methods discussed in this paper will be typical of future visualization environments. In the next section we discuss why we believe 3D flow visualization is a challenging HPC problem. These challenges motivate the governing ideas of the environment. In section 3 we present two interactive visualization techniques that are suited to analyze large flow data sets. These techniques illustrate how the governing ideas can be put to practice. In section 4 we elaborate how our environment is used to explore a turbulent flow field. Finally, we draw some conclusions.

Further author information: E-mail: {wimc|robertl}@cwi.nl

2. FLOW VISUALIZATION AS A CHALLENGING HPC PROBLEM

An important consideration for a flow visualization environment is the collection of visualization methods and the effectiveness of these methods. Equally important are issues related to user interfacing and, in particular for large data sets, system architecture, data formats and data handling. In this section, we briefly describe various user requirements for such an environment, how these translate into advanced visualization techniques, and the role of high performance computing and networking.

Analysis of 3D Flow Fields As stated in the previous section, large scale CFD simulations produce very large and information rich data sets. Although aggregate properties using numerical/statistical methods are useful, e.g. for comparing output to physical experiments, more is required for the in depth understanding of the flow phenomena in the data. The role of visualization will become more important in the analysis of these data sets. However, in the context of very large data sets, various difficulties arise. Two issues are considered:

- flow phenomena

  While the data sets contain only values such as velocity and pressure, the flow expert is interested in flow phenomena on a higher level, such as flow separation, re-attachment, vortex formation, etc. Methods are sought to extract these phenomena from the data. The difficulty is that these phenomena themselves are not well understood. Extracting and displaying these phenomena from the underlying data at the required detail is still an unsolved problem.

- levels of scale

  Phenomena in flow fields are characterized by flow patterns of widely varying spatial and temporal scales. In state of the art simulations, pattern sizes may vary by three orders of magnitude. Visualization methods should be able to cope with these different levels of scale.

Advanced Visualization Techniques There is no natural visual representation for 3D vector fields. Many visualization techniques have been proposed,[2,3] such as arrow plots, streamlines or particle based methods. These techniques are effective for certain data sets, but may be less effective for others. In order to extract the required information from time dependent data sets, more sophisticated visualization techniques are needed.

The governing philosophy that has driven the development of all visualization techniques is: in the analysis process the user begins with global inspection of the data at low level and iteratively progresses towards higher level concepts. There are two techniques have addressed this philosophy:

- texture based visualization methods

  Texture based visualization methods map a vector field to a texture.[4,5] The primary advantage over other flow visualization techniques is that texture can give a continuous view of a field opposed to visualization at only discrete positions, as with arrow plots or streamlines. The visual effect of direction in a texture is achieved by line structures in the direction of the vector field. These lines are the result of coherence between neighboring pixels in the texture. Coherency in the texture will be higher in the direction of the vector field than in other directions.

- feature visualization

  Instead of direct visualization of raw data, feature visualization techniques extract meaningful structures from the data and depict these structures schematically. In this way high level abstract visual representations can be produced. Features can have a higher information content, enabling the user to disregard redundant data and can help reduce complexity.

  Examples of feature visualization include flow topology analysis,[6] vortex detection and tracking,[7] localization of shockwaves, etc.

Role of High Performance Computing and Networking State of the art CFD simulations are capable of generating many of gigabytes of data. High performance computing and networking plays an important role in providing near real time response when processing these data sets. Interactive response times are essential when exploration of data sets is required. Two HPC related issues are considered:

- data movement

  Distributed computing allows a separation of visualization resources from the compute and file storage resources. While distribution provides an intuitive mapping of machine resources, it is not without cost. Performance of these systems will in large part depend upon the systems' ability to transfer large amounts of data between visualization tools. These tools must be designed to handle data management efficiently and to take advantage of high bandwidth parallel I/O.

- computation of visualization techniques

  Sequential visualization techniques do not suffice for the presentation of very large data sets. Techniques should be designed to execute on parallel machines and must be scalable on the size of the data set.

  A different problem arises when a single data set is larger than the capacity of main memory. Unfortunately, most visualization techniques assume that complete data sets reside in core, resulting in potential inefficient mappings. Visualization techniques should be crafted to take out of core algorithms into account. Out of core algorithms have been reported in Cox and Ellsworth.[8]

3. METHODS FOR HIGH PERFORMANCE VISUALIZATION

In this section we briefly discuss the implementation of two visualization techniques. These serve as illustrations of techniques which can be used to explore large flows.

Interactive spot noise Spot noise[4] is a texture synthesis technique which can be used to present a global overview of a vector field. In spot noise small icons, called spots, deformed according to the underlying data are used to show a flow field. If many spots are used to represent the flow, the individual spots can no longer be discerned and texture is perceived instead. This idea is illustrated in figure 1. A spot noise texture is characterized by a scalar function f of position $\mathbf{x}$. It is defined as

$$f(\mathbf{x}) = \sum a_i h(\mathbf{x} - \mathbf{x}_i)$$

in which $h(\mathbf{x})$ is called the spot function. It is a function everywhere zero except for an area that is small compared to the texture size. a_i is a random scaling factor with a zero mean, $\mathbf{x}_i$ is a random position. In non-mathematical terms: spots of random intensity are drawn and blended together on random positions on a plane.

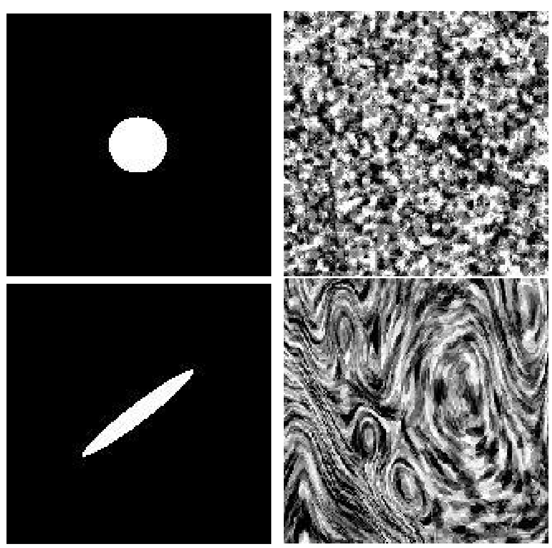

Figure 1. Spot(left) and generated texture(right). Circular spots(top) and deformed spots (bottom).

The downside of spot noise is that it is very computationally expensive. A large number of particle paths and particle positions must be calculated, spots must be transformed, scan converted, textured and blended. An interactive spot noise algorithm was presented by de Leeuw and van Liere.[9] The scalable algorithm partitions work evenly among processors and multiple graphics pipes. Interactive speeds can be obtained for data sets similar to the ones in section 4 by using a16 R10K CPU Silicon Graphics Onyx2 with 4 InfiniteReality graphics pipelines.

High texture generation speeds can be used for interactive adjustment of spot noise parameters to highlight certain aspects of the flow, or to zoom in on details of the flow. Also, animations of time dependent flow can be generated on the fly.

Interactive hierarchical flow topology Vector field topology was introduced by Helman and Hesselink.[6] It presents essential information by partitioning the flow field in regions using critical points which are connected with streamlines. Critical points are points in the flow where the velocity vector equals zero. These points, in which the medium does not move, can be classified based on the behavior of the flow close to it. For this classification the Eigen values of the velocity gradient tensor are used. The velocity gradient tensor – or Jacobian – is defined as:

$$\mathbf{J} = \nabla \vec{u} = \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \quad (1)$$

in which subscripts denote partial derivatives. Based on the two possibly complex Eigen values, five different cases are distinguished. For a *saddle point* the imaginary parts are zero and the real parts have opposite signs. *Repelling node*, imaginary parts are zero and the real parts are both positive. *Attracting node*, imaginary parts are zero and the real parts are both negative. *Repelling focus*, imaginary parts are non zero and the real parts are positive. *Attracting focus*, imaginary parts are non zero and the real parts are negative. When the real part of the Eigen values is zero the type of critical point is determined by higher order terms of the approximation of the flow in the neighborhood of the flow.

The five cases are diagrammed in figure 2.

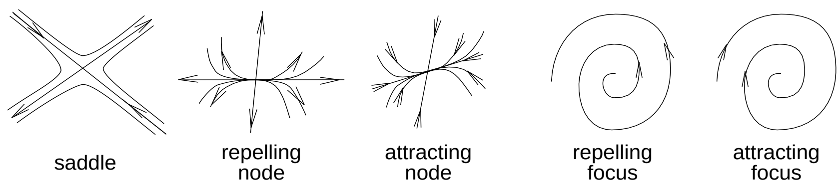

Figure 2. Five types of critical points.

In flow, the number of critical points can be very large. For the case described in section 4, 500 critical points in a single slice are not unusual. In addition, the number of critical points might vary quickly over time. Many of the critical points are caused by flow features at small time and space scales. This is not only computationally expensive but also clutters the display distracting the attention from the main structure of the flow. Furthermore, if animation is used, the frequent appearance and disappearance of critical points and associated stream lines gives a very unstable appearance.

A large number of critical points have only a local influence on the flow field. The global structure is determined by only a few critical points. In de Leeuw and van Liere[10] a method was described to reduce the number of critical points based on the flow area accociated with source and sink type of critical points. This method proved too slow for use interactively on large time dependent data sets. We want to limit the number of critical points which are displayed to those that are of interest at a certain level of scale at interactive speed. To achieve this we implemented three strategies. Each strategy is controlled by a parameter that assigns a weighting factor to all critical points. By interactively varying the parameter, the user can manipulate the number of critical points displayed.

- Pairing: an often occurring small disturbance of the flow is a vortex. The topological structure of a two dimensional vortex is shown in figure 3. It is a non-saddle (source or sink) combined with a saddle point. The size of the disturbance is estimated by the distance between these two critical points. After all critical points are located and classified the pairwise distance between all non saddles and saddles can be determined. The lower limit of this distance is used as a parameter for selecting critical points. All pairs with a distance below the threshold are not shown in the visualization.
- Subsampling data: Larger structures in the flow will not be limited to a few cells in the data. By looking for critical points in data which is subsampled to a certain extend only flow features at a larger scale will be found. The controlling parameter is the subsampling factor.

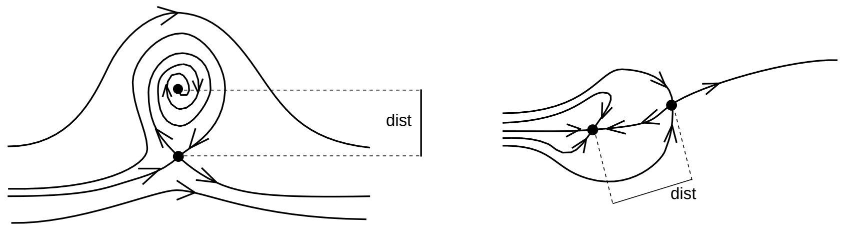

Figure 3. Two types of critical point pairs. Left a focus and saddle point. Right a attracting node and a sadlle.

- Persistence: Critical points pop up at a certain time, move through the flow, and disappear. By tracking the critical points over time the life time can be determined. This strategy can be implemented through a look ahead buffer which stores a number of timesteps. The controlling parameter is the minimum lifetime of a critical point.

Data management A general purpose architecture has been developed which emphasizes on flexible data management and movement.[11,12] The architecture is schematically diagrammed in figure 4.*

It is centered around a distributed *data manager* that acts as a blackboard for communicating values. Visualization processes can connect to the data manager and exchange data with it.

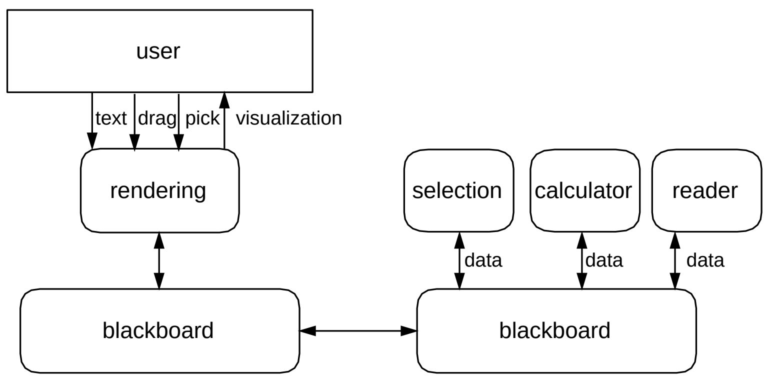

Figure 4. Distributed blackboard architecture.

The purpose of the data manager is twofold. First, it manages a database of variables. Processes can create, open, close, read, and write variables. Second, the data manager acts as an event notification manager. Processes can subscribe to state changes in the data manager. When such a state change occurs the process will receive a notification from the data manager. For example, if a process subscribes to mutation events on a particular variable, the data manager will send a notification to the process whenever the value of the variable is mutated.

A sample configuration is given in figure 4. This configuration consists of two data managers connected by a network and 4 visualization processes: the renderer, a data slicer, a calculator, and a data reader. The user interacts with the visualization by picking/ dragging geometric objects or entering text. Processes exchange data by reading from and writing to variables to the data managers. Data managers maintain the coherency of variables. Many other processes can attach to the data manager.

There are various advantages how this architecture handles data management related tasks. First, the data manager handles data storage efficiently to local processes by shared memory. Second, data movement is also efficient. Processes read and write data from the data manager in parallel. Also, the reader process can perform parallel I/O to disk. Third, data is moved between data managers only upon demand, so that only the minimum amount of data is transported over the network.

*We assume that data sets are stored on disk and are analyzed as a post-processing step. Interactive visualization, in the sense of computational steering, is still beyond the scope of current compute technology.

4. DIRECT NUMERICAL SIMULATION OF TURBULENT FLOW

Problem and Data Set Verstappen and Veldman,[13] discuss methods for direct numerical simulation (DNS) of turbulent flow. DNS is the most accurate, but also the most expensive, way of computing turbulent flow. In this particular problem a DNS of a turbulent flow around a square cylinder at Re = 22,000 (at zero angle of attack) has been performed. Of particular interest is the detailed visualization of vortex formation and the transition from laminar to turbulent flow. Flow experts would like to use visualization as a tool to test hypotheses about flow phenomena and – after a detailed inspection of the animation – as a means to pose new hypotheses.

The computation and the size of the resulting data base is impressive: computation lasted three weeks on a 12 CPU Cray C90. The resolution of the rectilinear grid was 316x540x64; the grid was finest nearby the cylinder. The total number of time steps computed was 100,000. It was impossible to store all data on disk; thus a selection of the data was made. 7500 timesteps of a XY-slice were taken, resulting in about 50 Gigabytes of data. Future plans are to process a 3D selection of the data.

Results The goal of the visualization is to make interactive and detailed animations of this data set. In contrast to prerecorded video sequences, the interactive animations allow users to iteratively select visualization mappings and then play through a part of the data set.

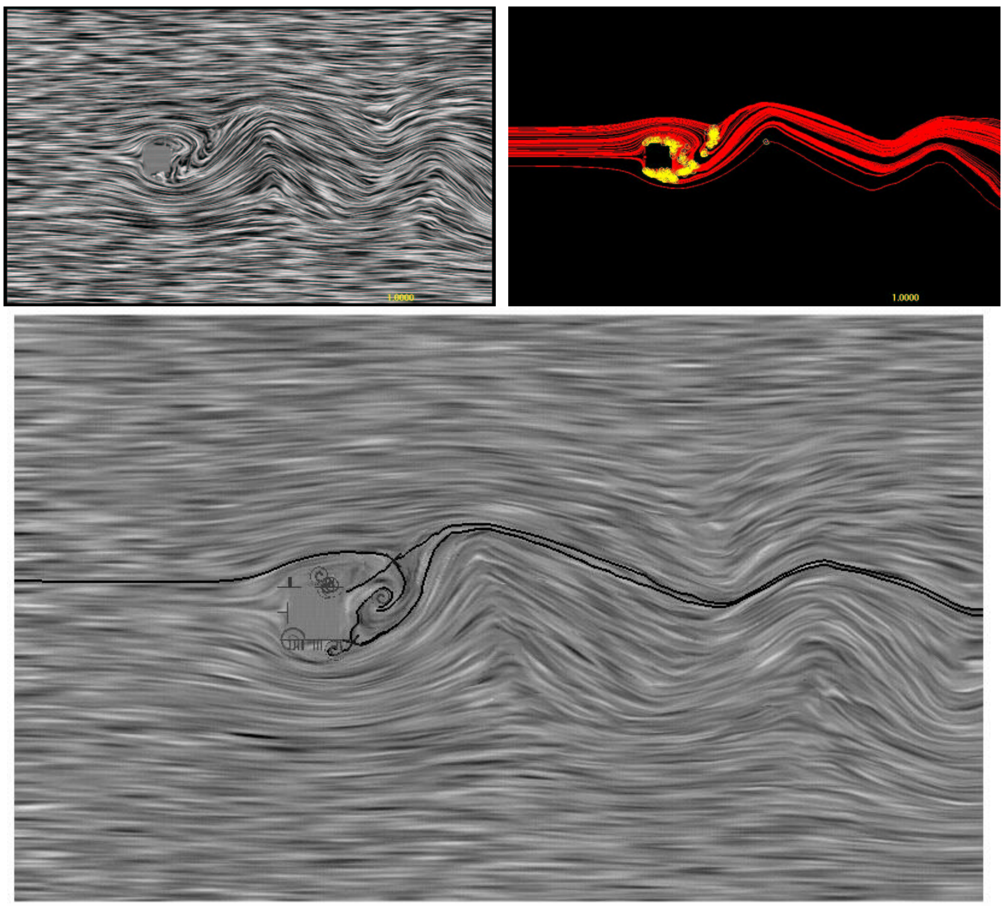

Figure 5. Three global views of flow around a square cylinder. Spot noise only (top left), all critical points with stream lines (top right), spot noise and a selection of critical points (bottom).

Figure 5 shows three views of the flow at a particular time. On the top left, spot noise was used. This provides a continuous view of the flow and contains all data. The top right image shows the topology of the flow. The complete set of critical points is shown. Small colored icons are used to denote the set of critical points: a yellow spiral icon denotes a focus, a blue cross denotes a saddle point, and cyan/magenta disks denote repelling/attracting nodes. Red streamlines are used to connect the icons.

This image shows 559 critical points and 916 streamlines. The bottom image combines spot noise with the flow topology. In addition, by adjusting parameters, only 40 critical points are shown. These points give a clear insight in the global structure of the flow without excessive cluttering the image.

Figure 6 shows two detailed views of the previous image. These images have been realized by interactively zooming. In the left image an area around the cylinder has been selected. By zooming further on an area below the cylinder gives the image on the right. Various spot noise and topology parameters are adjusted during zooming. When animation is used the formation and evolution of vortices around the cylinder can be studied.

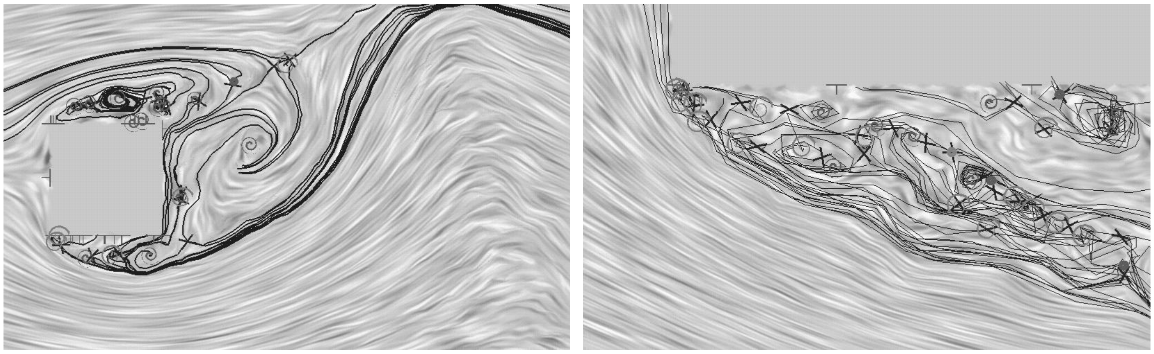

Figure 6. Two zoomed in views of the previous image. Area around the square cylinder with selected critical points (left), and an area below the cylinder with all critical points (right).

The environment is capable of generating these images at interactive speeds, allowing animations to be made on the fly. This includes reading the necessary data from disk, transporting the data to the visualization engine, mapping the data to geometry, and rendering the final image.

Evaluation This application clearly benefits from the added value of an interactive flow visualization environment. This data set contains an abundance of detailed information. Interaction is necessary because it is not a priori known which aspects of the flow are important.

The combination of spot noise and flow topology techniques provided additional insight that was difficult to obtain when only one technique was used.[14] By using traditional topology visualization methods on these data sets, excessive cluttering can not be avoided. Using the hierarchical approach, simplified views of the topology can be obtained without cluttering. In addition, due to interactive zooming, topological information at various levels of scale can be obtained.

During this work a number of additional observations were made:

- The simulation took approximately three weeks to compute on a Cray C90 at the Academic Computing Services Amsterdam, SARA. The transfer of data from compute server to our local file server took about four days. Data could be processed interactively once it was on the local machines. We see data transfer as major concern for effective data analysis of large scale simulations.
- The resulting visualizations are information rich. To display the results, a large 179cm ElectroHome Retro back projection enclosure was used. Although the quantitative content is the same as with small displays, visualizing data sets on large displays provide additional insight. We believe that this is because small details – which cover only a few pixels – are easier to see.

The resulting visualization was only 2D, but full 3D interactive animations are desired. However, problems remain to be addressed: the visualization methods discussed in this paper have to be extended to cope with 3D data. Also, storage of the full 3D data set on disk is currently not possible. Although extending the environment to 3D involves much more work, we believe that our motivations with respect to interactivity and integration will be be usable in 3D data.

In the near future DNS can be applied to flows with a Reynolds number in the order of 10^5. The increased size of the resulting data sets will put an even greater burden on the interactive visualization environment.

5. CONCLUSION

This paper motivates the need for an interactive visualization environment for the analysis of 3D flow. The design of the environment is centered around two fundamental beliefs: First, the environment should integrate modeling and visualization. Ultimately, we envision that visualization will be an integral part in the process of modeling complex flow phenomena. Second, interactive visualization is essential for gaining new insight in the nature of flow.

High performance computing plays an important role in flow visualization. First, since the simulated flow is becoming more detailed, advanced visualization methods are required to present the information from the underlying data. Second, in order to stream data to the visualization pipelines at constant rates, management of very large data sets require efficient usage of high bandwidth networks and parallel I/O.

The implementations of two visualization techniques – spot noise and hierarchical flow topology – have been discussed. Both techniques are well suited for presenting large data sets. Interactive usage of these techniques, particularly in the case of zooming, makes them useful for exploration. Although much work still needs to be done for a 3D environment, we believe that the approach discussed in this paper will be useful for the design of future visualization environments.

REFERENCES

1. P.R. Woodward. Interactive scientific visualization of fluid flow. *IEEE Computer*, 26(10):13–25, October 1993.
2. J.J. van Wijk, A.J.S. Hin, W.C. de Leeuw, and F.H. Post. Three ways to show 3D fluid flows. *IEEE Computer Graphics & Applications*, 14(5):33–39, september 1994.
3. L. Hesselink and T. Delmarcelle. Visualization of vector and tensor data sets. In L.J. Rosenblum et al., editors, *Scientific Visualization: Advances and Challenges*, pages 419–433. Academic press, 1994.
4. J.J. van Wijk. Spot noise – texture synthesis for data visualization. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):263–272, July 1991.
5. B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):263–272, August 1993.
6. J.L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991.
7. D. Silver, N. Zabusky, V. Fernandez, and M. Gao. Ellipsoidal quantification of evolving phenomena. In *Scientific Visualization of Physical Phenomena*, pages 573–588. Springer Verlag, 1991.
8. M.B. Cox and D. Ellsworth. Application-controlled demand paging for out-of-core visualization. In R. Yagel and H. Hagen, editors, *Proceedings Visualization '97*, pages 235–244. IEEE Computer Society Press, 1997.
9. W.C. de Leeuw and R. van Liere. Divide and conquer spot noise. In *Proceedings Super Computing '97 (http://scxy.tc.cornell.edu/sc97/program/ TECH/DELEEUW/INDEX.HTM)*, 1997.
10. W.C. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In D. Ebert, M. Gross, and B. Hamann, editors, *Proceedings Visualization '99*, pages 349–354. IEEE Computer Society Press, 1998.
11. J.J van Wijk and R. van Liere. An environment for computaional steering. In G.M. Nielson, H. Müller, and H. Hagen, editors, *Scientific Visualization: Overviews, Methodologies, and Techniques*, 1996.
12. R. van Liere, J.A. Harkes, and W.C. de Leeuw. A distributed blackboard architecture for interactive data visualization. In D. Ebert, H. Hagen, and H. Rushmeier, editors, *Proceedings Visualization '98*, pages 225–231. IEEE Computer Society Press, 1998.
13. R.W.C.P. Verstappen and A.E.P. Veldman. Direct numerical simulation at lower costs. *Journal of Mathematical Engineering*, June 1997.
14. A.E.P. Veldman. Private communication.