Distributed Visualization and Analysis of Fluid Dynamics Data

Hans-Christian HEGE*, Tino WEINKAUF*, Steffen PROHASKA*, and Andrei HUTANU*

* Zuse Institute Berlin, Scientific Visualization Group, Germany
{hege,weinkauf,prohaska,hutanu}@zib.de, http://www.zib.de/visual/

Fluid dynamics applications require a good understanding of the underlying physical phenomena. Therefore, effective procedures are necessary for analyzing and visualizing the various physical fields. Beside interactive and perceptually efficient techniques for visualizing flow fields directly, there is strong demand for methods that uncover hidden flow structures. Some recently developed feature based visual analysis methods are exemplarily presented. Fluid flow data typically are large and often are stored remotely or distributedly. The interactive visual analysis of such large data sets requires new software architectures – ideally utilizing emerging Grid standards. We discuss such architectures and report on specific realizations in the framework of the visualization system Amira.

Keywords: flow visualization, vector fields, feature extraction, topology, distributed systems, Grid technology

1 Introduction

The goal of visual data analysis is to extract and to graphically present essential characteristics of data sets – particularly data sets that are too complex or too large to be inspected by other means. Fluid dynamics (FD) applications, like e.g. optimization of technical designs, often require a detailed understanding of the underlying physical phenomena. Mental comprehension of these phenomena can be greatly supported by effective procedures for analyzing scalar, vector and tensor fields – in combination with techniques for creation of intuitively perceivable visualizations and means for interactive exploration.

Beside visualization techniques that depict fluid flows directly, there is strong demand for methods that reveal hidden flow structures as well as changes of such structures due to parameter variations. Here we will discuss a few techniques both for direct visualization and feature extraction, which have been recently developed at ZIB.(7)

In typical application scenarios the data sets to be analyzed are large and often remotely or distributedly stored. Interactive analysis of such data sets requires new data structures, new algorithms and new software architectures. It is an interesting question to what extend the emerging Grid technology standards and Grid services can be used for that. We discuss such distributed architectures from a general point of view and report on systems for interactive analysis of large, remote data as well as collaborative work, based on the visualization system Amira.(28)

2 Visualization Techniques for FD

Fluid flow phenomena are represented by stationary or time-dependent fields. Here we focus on the major data class, namely vector fields. In the last decade, a number of approaches for analyzing and visualizing vector fields have been developed. We distinguish between methods that directly depict vector fields and those that first extract features and then display these.

2.1 Direct Visualization

The probably most successful direct visualization technique for 2D vector fields is the so-called line integral convolution (LIC),(6) which first was developed for stationary fields. In this method a noise field is convolved along the field lines, resulting in textures that are highly correlated along the field lines and almost uncorrelated perpendicular to these. The resulting grey value texture often is colored to depict an additional scalar field, c.f. Fig. 1. The original method, which was too slow for practical use, has been accelerated

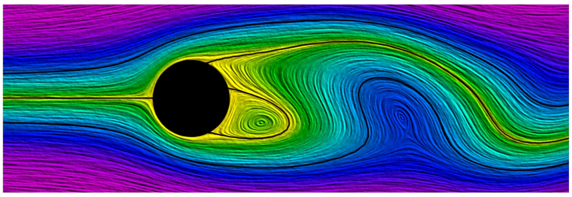


Figure 1: Visualization of a flow around a cylinder, using the line integral convolution (LIC) method.

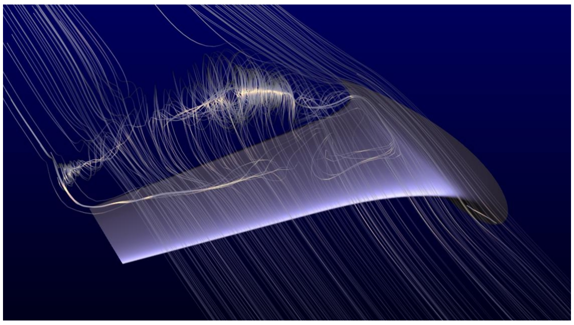


Figure 2: Visualization of a flow around a airfoil wing, using the 'illuminated streamline' method.

by two orders of magnitude,[14, 27] and variants for visualizing instationary 2D vector fields have been developed.[3, 19, 25] The advantage of LIC methods is that the flow is depicted in an intuitive, spatially highly resolved way, which does not depend on a (potentially biased) selection of streamlines. Though both, computation of 3D LIC textures from 3D vector fields and visualization by direct volume rendering is straighforward, the results are not satisfying: occlusion severely hinders perception. Interaction techniques for clipping and emphasizing important subregions, as well as visualization techniques for enhancing depth and orientation perception, cannot provide a full remedy. In 3D, therefore, sparse and preferably transparent objects, representing structures of the field, are better suited. The method of 'illuminated streamlines'[29, 35] represents such a technique: individual streamlines are computed and are displayed using the Phong shading model. This provides to the human visual system useful information about the local orientation and curvature of curve segments, c.f. Fig. 2. This display technique with its favorable perceptual properties can be applied also to display other kinds of lines that represent 3D flow structures. Extension of such methods to time-dependent flows is a current research topic. For further information regarding direct visualization techniques for flow visualization we refer to a review paper.[18]

2.2 Feature Extraction

Feature based approaches[21] play an important role for the visualization of flow fields, because

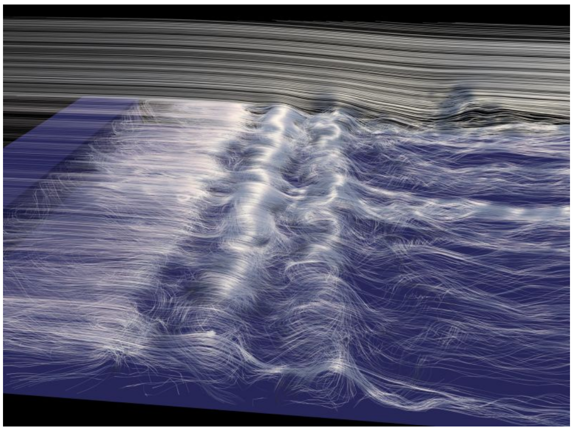


Figure 3: Flow around a backward-facing step. Turbulent regions are emphasized by seeding more streamlines in regions of high mean curvature. Data courtesy of Kaltenbach and Janke (TU Berlin).

they focus on the most important information and reduce the visual complexity. The decision which features are to be visualized always depends on the specific application. In this paper we concentrate on two representatives of feature based visualization: curvature based seeding of streamlines and topological methods.

2.2.1 Curvature Based Streamline Seeding

Curvature measures can be derived from geometric entities of 3D vector fields, i.e. streamlines or normal surfaces. Whereas normal surfaces are understood to be perpendicular to the vector field.

These curvature measures carry topological information as they tend to infinity only near critical points.[33] The mean curvature shows this behavior for all types of first-order critical points. These values can be computed without extracting the geometric entities, but just by knowing the vector field and its derivatives.

Based on these scalar quantities a probability distribution can be generated to seed more streamlines in regions of high curvature and less in areas of low curvature. This emphasizes turbulent parts of the flow and blends out laminar regions, see Fig. 3.

2.2.2 Topological Methods

The main idea of vector field topology is to segment the flow into areas of different flow behavior by extracting critical points and separatrices. They allow to describe even complex flow behaviors by only a limited number of graphical primitives.

Consider a 3D vector field

$$\mathbf{v}(x, y, z) = \begin{pmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{pmatrix}. \qquad (1)$$

A first order critical point $\mathbf{x}_0$ (i.e., $\mathbf{v}(\mathbf{x}_0) = \mathbf{0}$) can be classified by an eigenvalue/eigenvector analysis of the Jacobian matrix $\mathbf{J}_\mathbf{v}(\mathbf{x}) = \nabla\mathbf{v}(\mathbf{x})$, iff $\det(\mathbf{J}_\mathbf{v}(\mathbf{x}_0)) \neq 0$. Let $\lambda_1, \lambda_2, \lambda_3$ be the eigenvalues of $\mathbf{J}_\mathbf{v}(\mathbf{x}_0)$ ordered according to their real parts, i.e. $Re(\lambda_1) \leq Re(\lambda_2) \leq Re(\lambda_3)$. Furthermore, let $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ be the corresponding eigenvectors, and let $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$ be the corresponding eigenvectors of the transposed Jacobian $(\mathbf{J}_\mathbf{v}(\mathbf{x}_0))^T$. (Note that $\mathbf{J}$ and $\mathbf{J}^T$ have the same eigenvalues but not necessarily the same eigenvectors.) The sign of the real part of an eigenvalue λ_i denotes—together with the corresponding eigenvector $\mathbf{e}_i$—the flow direction: Positive values represent *outflow* and negative values *inflow* behavior. This leads to the following classification of first order critical points:

Sources:	$0 < Re(\lambda_1) \leq Re(\lambda_2) \leq Re(\lambda_3)$
Repelling saddles:	$Re(\lambda_1) < 0 < Re(\lambda_2) \leq Re(\lambda_3)$
Attracting saddles:	$Re(\lambda_1) \leq Re(\lambda_2) < 0 < Re(\lambda_3)$
Sinks:	$Re(\lambda_1) \leq Re(\lambda_2) \leq Re(\lambda_3) < 0$

Thus, sources and sinks consist of complete outflow/inflow, while saddles have a mixture of both. A repelling saddle has one direction of inflow behavior (called *inflow direction*) and a plane in which a 2D outflow behavior occurs (called *outflow plane*). Similar to this, an attracting saddle consists of an *outflow direction* and an *inflow plane*. Each of the 4 classes above can be further divided into two stable subclasses by deciding whether or not imaginary parts in two of the eigenvalues are present ($\lambda_1, \lambda_2, \lambda_3$ are not ordered):

Foci:	$Im(\lambda_1) = 0$ and $Im(\lambda_2) = -Im(\lambda_3) \neq 0$
Nodes:	$Im(\lambda_1) = Im(\lambda_2) = Im(\lambda_3) = 0$

An iconic representation is an appropriate visualization for critical points, since vector fields usually contain a finite number of them. Several icons have been proposed in the literature.[11,12,15,30] We follow the design approach of[34] and color the icons depending on the flow behavior: Attracting parts (inflow) are colored blue, while repelling parts (outflow) are colored red (Figs. 4 and 5).

Separatrices are streamlines or stream surfaces which separate regions of different flow behavior. Each saddle point creates two separatrices: Considering a repelling saddle $\mathbf{x}_R$, it creates one separation curve (which is a streamline starting in $\mathbf{x}_R$ in the inflow direction by backward integration) and a separation surface (which is a stream surface starting in the outflow plane by forward integration), c.f. Fig. 6. A similar statement holds for attracting saddles. Other kinds of separatrices are possible as well: They can emanate from

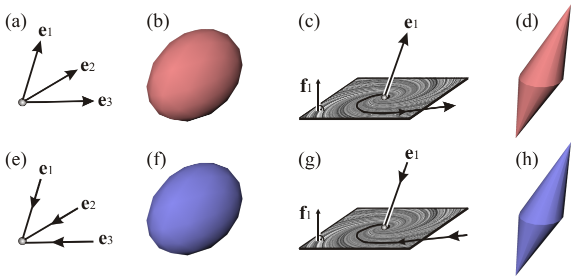


Figure 4: Sources and sinks; (a) repelling node and (b) its icon; (c) repelling focus and (d) its icon; (e) attracting node and (f) its icon; (g) attracting focus and (h) its icon.

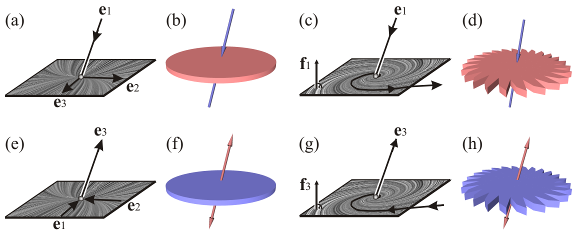


Figure 5: Repelling and attracting saddles; (a) repelling node saddle and (b) its icon; (c) repelling focus saddle and (d) its icon; (e) attracting node saddle and (f) its icon; (g) attracting focus saddle and (h) its icon.

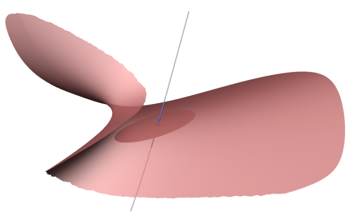
boundary switch curves,[34] attachment and detachment lines, or they are closed separatrices without a specific emanating structure.

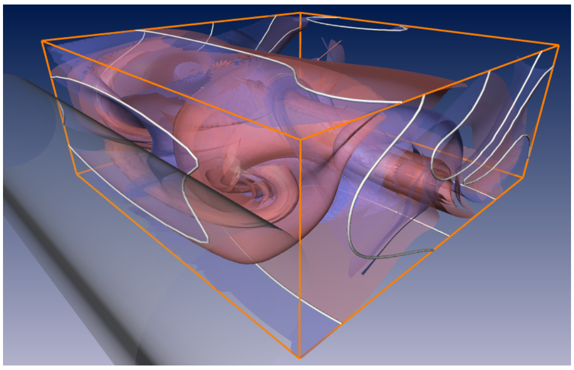
Visualizing a rather complex topological skeleton involves showing a number of separation surfaces. This does not lead to visually pleasing results, because these surfaces hide most parts of the skeleton (Fig. 7a).

A solution of this problem is the appliance of saddle connectors.[30] A saddle connector is the intersection curve of two separation surfaces, where one is emanating from a repelling and the other from an attracting saddle. This intersection curve is a streamline connecting both saddles, i.e. it starts at the repelling and ends at the attracting saddle. This concept was extended by applying the main idea to the separation surfaces emanating from boundary switch curves, yielding the concept of boundary switch connectors.[34]

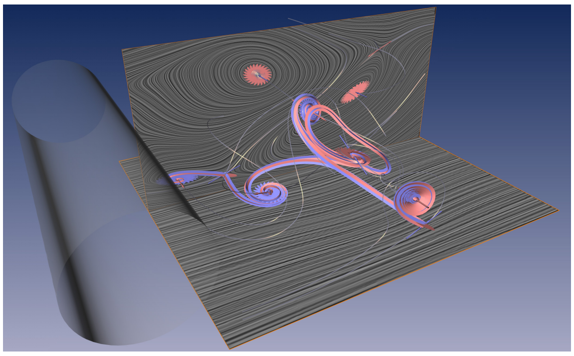
Ref.[30] uses double flow ribbons for visualizing saddle connectors (Fig. 7b). Although this approach incorporates the local behavior of the separation surfaces, the flow behavior of $\mathbf{v}$ can no longer be uniquely inferred from any point of

Figure 6: Separatrices originating from a repelling node saddle.





(a) Due to the shown separation surfaces, the topological skeleton of the vector field looks visually cluttered.



(b) Visual clutter has been reduced by the display of saddle connectors instead of separations surfaces. Additional LIC planes have been placed to show the correspondence between the skeleton and the flow.

Figure 7: Topological skeleton of a flow behind a circular cylinder. 13 critical points and 9 saddle connectors have been detected. Data courtesy of Gerd Mutschke (FZ Rossendorf) and Bernd R. Noack (TU Berlin).

the domain. Therefore, we enable the user to interactively demand the display of *single* separation surfaces by simply clicking on a saddle connector.(30) However, saddle connectors yield for the first time expressive visualizations of complex topological skeletons with a higher number of critical points and separatrices.

3 Visualization of Large/Distributed Data

Data, as produced by numerical simulations like fluid dynamics computations, or the data generated by image acquisition systems reaches ever growing sizes. Data sets of tens or hundreds of gigabytes are common today, tendency increasing. The direct application of the existing visualization algorithms is difficult and in some cases even impossible. New visualization algorithms and techniques need to be developed in order to manage the immense data sets. Simulation data is produced on large machines that can provide the amount of compute power needed. Acquisition systems are often directly connected to storage systems large enough to hold data gained from multiple sources. In most cases, simply transferring (complete) data sets to the visualization client, which is the standard procedure used to visualize small data sets, is not possible. Scientists use network connections to the machines that store the data in order to be able to visualize it. The task of *distributed visualization*(5) is to optimize the usage of server and client power and network bandwidth to provide interactivity and/or high quality images.

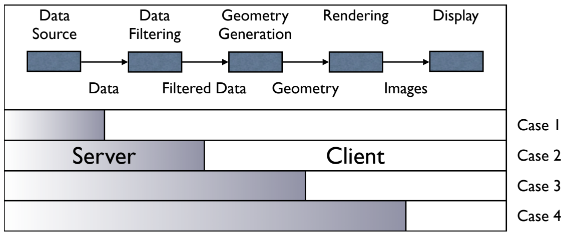


Figure 8: Dataflow in scientific visualization. In a client-server setting, calculations can be distributed in various way. Case 1: All calculations are performed on the client. Case 2: Data is preprocessed on the server. Case 3: Geometry transferred. Case 4: Images streamed from server to client.

3.1 Distributed Visualization Pipeline

Since the data is (at least initially) separated from the visualization client, we are dealing with a distributed system. The visualization pipeline as shown in Fig. 8 can be split in five stages: data access, data filtering (selection and modification of the data), generation of graphics data (geometries or features), rendering (transforming graphics data in images), and display. Depending on the hardware configuration of the targeted distributed system, we can choose different modes of distributing the visualization pipeline. The task is even more challenging when multiple computers are involved in a collaborative visualization session. In the following we will describe different distribution mechanisms, their integration in Amira and their usage in existing and prospective distributed and collaborative scenarios. Simple remote data access is not an option if data become large. Therefore, we start with remote data filtering.

3.2 Remote Data Filtering

In this case data is stored on the remote machine with a data server running which is able to perform simple data selection operations. Depending on the server configuration, parallel data filters,(10) efficient file formats,(13) or format independent data filters(24) may be used. A number of representative client-server or remote data access architectures are available including the Storage Resource Broker,(2) DataCutter,(4) Active Data

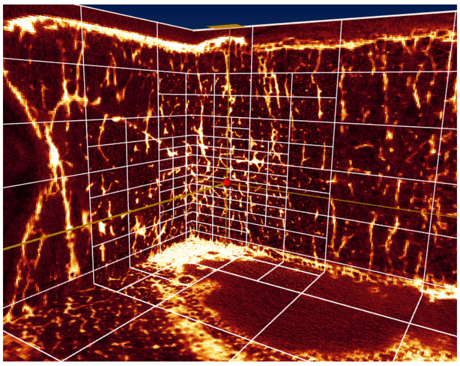


Figure 9: Orthogonal slicing using remote data filtering. An overview together with details around a focus point are displayed at the same time. Low resolution data is generated on the server as requested by the client.

Repository,[17] or OGSA.[9] In Amira, grid technologies based on GridFTP for remote data filtering were adopted.[13,22] GridFTP is the current Grid standard for remote data and file transfer.[1] It is an extension of the FTP protocol that includes grid security mechanisms. For remote data filtering is supported through customisable server-side processing instructions.

The advantages of this type of distribution are the simple hardware requirements for the server (no graphics hardware needed) and the flexibility in choosing the visualization mode for the client. Remote data filtering is extremely useful when visualizing image data (see Fig. 9), and in collaborative scenarios—they provide the perfect mechanism for making the data available to multiple visualization clients—especially when using the grid security mechanisms. A disadvantage is the relatively hardware and software requirements on the client side. In order to be useful, multi-resolution and progressive visualization algorithms have to be used together with the remote filtering mechanisms.[22]

3.3 Geometry Streaming

Another approach in distributing the visualization pipeline is to separate the filtering and graphic object generation stages from the last two stages (rendering and image display). The server calculates geometries or extracts features from the dataset which are transferred to the client which in turn handles the final rendering and the user interaction.[8,16] Depending on bandwidth, latency and computational power, a system might support different distributions of the visualization pipeline.[20]

In Amira, we are building a flexible system using the Simple Object Access Protocol (SOAP). SOAP[31] is a XML-based protocol that defines a standard to encode remote procedure calls. Using these, Amira can act both as a client and as a server, even simultaneously, which provides the full power remotely whithout implementing spezialized servers. All data readers, data filters and geometry generation algorithms are available locally and remotely. In a collaborative scenario this is useful for sharing data and geometry between multiple instances of Amira. Scientists running a collaborative visualization session are able to turn on data sharing, and other users participating in the session can then simply access data to analyze it using their visualization modules of choice, or transfer geometry for local rendering.

3.4 Image Streaming

The last approach (case 4 in Fig. 8) is an image streaming approach like VNC[23] or Vizserver.[26] While Amira fully supports these technologies, we have also implemented a video streaming solution. Based on the standard RTP/RTSP protocols, this implementation gives our users the possibility to connect with a light-weight video client, like QuickTime or MPlayer, to a running instance of Amira and (passively) participate in the collaborative session. This could also provide the means to integrate Amira in a video conference system. The mechanisms can take advantage of a multicast infrastructure and are very efficient in regards to consumed bandwidth.

Together with the SOAP server that is already able to serve high-resolution snapshot requests, this will enable a complete range of image sharing mechanisms (live – low resolution, static – high resolution) between the collaborating users. Using Amira as a component in web-based applications[32] will become easier by using these technologies.

4 Conclusions

Visualization and analysis techniques for flow fields, developed in the last decade, provide powerful means for interactive analysis of complex fluid flow data. Current research is mainly concerned with methods for depiction of instationary 3D fields, improved and more sophisticated feature detection both for stationary and instationary fields, graphical representation of flow features – as well as methods for visualization of huge, remote and distributed data, while maintaining interactivity.

Acknowledgments

We thank Holger Theisel and Bernd R. Noack for fruitful collaboration. TK acknowledges financial support by the DFG Sfb ‘Control of Complex Turbulent Shear Flows’, SP by the European Space Agency Project AO-99-030, and AH by the European Commission Project 'GridLab', IST-2001-32133. All visualizations in this paper have been created using AMIRA – a system for advanced 3D visualization and volume modeling,[28] see `http://amira.zib.de/`.

References

(1) W. Allcock, J. Bester, and J. Bresnahan et. al. GridFTP: Protocol extensions to FTP for the Grid. *GWD-R (Recommendation)*, April 2003.

(2) C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC Storage Resource Broker. *Proceedings of CASCON'98, Toronto, Canada*, 1998.

(3) J. Becker and M. Rumpf. Visualization of time-dependent velocity fields by texture transport. In *Eurographics Workshop on Visualization in Scientific Computing*, pages 91–102, 1998.

(4) M.D. Beynon and R. Ferreira et. al. DataCutter: Middleware for filtering very large scientific datasets on archival storage systems. In *Proc. 2000 Mass Storage Systems Conf.*, pages 119–133. IEEE CS Press, 2000.

(5) K.W. Brodlie, D.A. Duce, J.R. Gallop, J.P.R.B. Walton, and J.D. Wood. Distributed and collaborative visualization. *Computer Graphics Forum*, 23(2):223–251, 2004.

(6) B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. *Computer Graphics*, 27:263–272, 1993.

(7) ZIB Dept. Scientific Visualization. Scientific Visualization – Current Research P/frerojects. www.zib.de/visual/projects, 2004.

(8) K. Engel, R. Westermann, and T. Ertl. Isosurface extraction techniques for web-based volume visualization. In *IEEE Visualization '99*, pages 139–146, 1999.

(9) I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke. Grid Services for Distributed System Integration. *Computer*, 35(6):37–46, 2002.

(10) L. A. Freitag and R. M. Loy. Adaptive, multiresolution visualization of large data sets using a distributed memory octree. In *Proc. SC99: High Performance Networking and Computing*, Portland, OR, November 1999. ACM Press and IEEE CS Press.

(11) A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *IEEE Visualization '91*, pages 33–40, 1991.

(12) H. Hauser and E. Gröller. Thorough insights by enhanced visualization of flow topology. In *9th Int. Symposium on Flow Visualization, CDROM*, 2000.

(13) H.-C. Hege, A. Hutanu, and R. Kähler et. al. Progressive retrieval and hierarchical visualization of large remote data. In *Proc. 2003 Workshop on Adaptive Grid Middleware*, pages 60–72, 2003.

(14) H.-C. Hege and D. Stalling. Fast LIC with piecewise polynominal filter kernels. In *Mathematical Visualization - Algorithms, Applications and Numerics*, pages 295–312. Springer, 1998.

(15) J. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11:36–46, May 1991.

(16) N. Jensen, S. Olbrich, H. Pralle, and S. Raasch. An efficient system for collaboration in tele-immersive environments. In *Fourth EG Workshop on Parallel Graphics and Visualization*, pages 123–131, 2002.

(17) T. Kurc and Ü. Çatalyürek et. al. Visualization of large data sets with the active data repository. *IEEE Comput. Graph. Appl.*, 21(4):24–33, 2001.

(18) R. S. Laramee, H. Hauser, H. Doleisch, and B. Vrolijk et. al. The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum*, 23(2):203–221, 2004.

(19) Z. P. Liu and R. J. Moorhead. AUFLIC: An accelerated algorithm for unsteady flow line integral convolution. In *Eurographics / IEEE TCVG Symposium on Visualization '02*, pages 43–52, 2002.

(20) E. J. Luke and C. D. Hansen. Semotus visum: a flexible remote visualization framework. In *IEEE Visualization '02*, pages 61–68, 2002.

(21) F. H. Post, B. Vrolijk, and H. Hauser et.al. Feature extraction and visualisation of flow fields. In *Eurographics 2002, State of the Art Reports*, pages 69–100, 2002.

(22) S. Prohaska, A. Hutanu, R. Kähler, and H.-C. Hege. Interactive exploration of large remote micro-CT scans. In *IEEE Visualization*, pages 345–352, 2004.

(23) T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1):33–38, January/February 1998.

(24) T. Schütt, A. Merzky, A. Hutanu, and F. Schintke. Remote partial file access using compact pattern descriptions. In *4th International Symposium on Cluster Computing and the Grid (CCgrid)*, 2004.

(25) H. W. Shen and D. Kao. UFLIC - A line integral convolution algorithm for visualizing unsteady flows. In R. Yagel and H. Hagen, editors, *Proc. IEEE Visualization '97*, pages 317–323, 1997.

(26) Silicon Graphics, Inc., 1600 Amphitheatre Pkwy, Mountain View, CA 94043, United States. *OpenGL Vizserver 3.1 White Paper - Application-Transparent Remote Interactive Visualization and Collaboration*, April 2003.

(27) D. Stalling and H. Hege. Fast and resolution independent line integral convolution. *Proceedings Siggraph '95*, pages 249–256, 1995. Los Angeles.

(28) D. Stalling, M. Westerhoff, and H.-C. Hege. Amira – a highly interactive system for visual data analysis. In C.R. Johnson and C.D. Hansen, editors, *Visualization Handbook*. Academic Press, 2004.

(29) D. Stalling, M. Zöckler, and H.-C. Hege. Fast display of illuminated field lines. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118–128, 1997.

(30) H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields. In *IEEE Visualization 2003*, pages 225–232, 2003.

(31) W3C XML Protocol Working Group. http://www.w3.org/TR/soap/.

(32) Y. Wang, D. A. Yuen, Z. A. Garbow, and G. Erlebacher. Web-based service of a visualization package "Amira" for the geosciences. *Visual Geosciences*, 2003. Accepted.

(33) T. Weinkauf and H. Theisel. Curvature measures of 3D vector fields and their applications. In *Journal of WSCG*, volume 10:2, pages 507–514, 2002.

(34) T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Boundary switch connectors for topological visualization of complex 3D vector fields. In *Data Visualization 2004. VisSym'04*, pages 183–192, 2004.

(35) M. Zöckler, D. Stalling, and H. C. Hege. Interactive visualization of 3D-vector fields using illuminated stream lines. In *Proc. IEEE Visualization '96*, pages 107–113, 1996.