gViz: Visualization and Computational Steering on the Grid

Ken Brodlie and Jason Wood

University of Leeds

David Duce and Musbah Sagar

Oxford Brookes University

Abstract

Visualization and computational steering are fundamental aspects of Grid computing. Visualization is important in the analysis and interpretation of the results generated by large scale simulation; computational steering enables effective use to be made of interactive access to high performance computing. This paper describes work in the gViz e-Science project: a reference model based on the dataflow paradigm to underpin the work; an XML application to describe visualization processing in a general way; and extensions of existing visualization systems to support visualization and computational steering in Grid computing environments. The work is illustrated through two demonstrators: one an environmental application, the other an application in computational biology which aims to understand heart abnormalities.

1. Motivation

Grid computing is extending the horizons of computational science, allowing aggregated computing resources to be harnessed in the solution of major problems. Visualization plays a crucial role: we shall fail to gain proper value from Grid computing without an effective means of interpreting the results from the large applications that can now be run. The challenge for the visualization community is to be able to provide the scientist with visualization systems which integrate seamlessly with Grid computing, and which provide for visualization of either offline data – from large data repositories - or online data – in computational steering applications. Moreover we need to express visualization applications in a way that is independent of any proprietary system: scientists need to be able to share visualizations, and record the process of generating a visualization so that it can be reproduced by another.

In this paper we describe work in the UK e-Science project, gViz, which has sought to evolve current visualization approaches to Grid environments. We have developed a new reference model for Grid-based visualization, and elaborated different aspects of the model in various mini-projects.

2. Reference Model

Our new reference model builds on the traditional dataflow model of Haber and McNabb (1990), which has underpinned many of the popular visualization software systems in use today, including environments such as IRIS Explorer and toolkits such as vtk. Wood, Wright and Brodlie (1997) extended this model to encompass collaborative visualization. The traditional model is a network of *conceptual* processes, turning numerical data into image data, and is expressed independently of software and hardware resource descriptions. Our new model first binds a software architecture to the model, and then brokers an allocation of physical resources from a Grid environment in order to obtain a final configuration. The software binding gives us a *logical* description, the hardware binding a *physical* description. We make no special distinction between visualization and computational steering, seeing the latter as just one type of visualization.

The model is elaborated in a paper at the All Hands 2003 conference (Duce and Sagar, 2003), and further discussed in the Eurographics State of the Art Report presented by several members of the gViz team (Brodlie et al, 2004).

3. Expressing Dataflow Networks as Diagrams and Languages

Tools are needed to allow users to express dataflow visualizations. We are very familiar with these at the level of network descriptions - both diagrammatic representation through the visual editors in AVS and IRIS Explorer, and also language representation through a vtk program in C. We have extended this concept to descriptions at the conceptual, logical and physical levels. We have defined a new XML application, called skML, which describes visualizations independently of the physical resources that might eventually be used to realise them. The language supports collaborative visualization, and allows a document at the conceptual level to be transformed automatically to corresponding documents at the logical level for different software systems (with proof of concept transformations for IRIS Explorer and Open DX). To complement the language, we have defined an editor (called the SVG Map Editor) that allows corresponding dataflow networks to be created. The user interface to the map editor is modelled on the map editor in IRIS Explorer; AVS is similar. The SVG Map Editor is written using web technologies, Scalable Vector Graphics (SVG) and JavaScript, so that it runs in a web browser.

Figure 1 shows the SVG map editor being used to create a network that reads in volume data, extracts an isosurface and displays the resulting geometry. Figure 2 shows the corresponding IRIS Explorer map that is generated automatically from the XML description underpinning the visual editor.

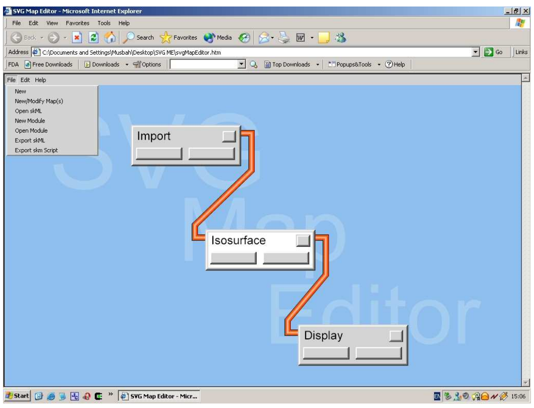

Figure 1 – SVG Map Editor

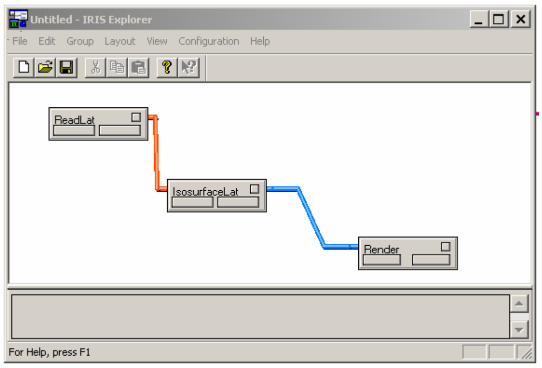

Figure 2 – Corresponding Map in IRIS Explorer

An IRIS Explorer module has been written to implement the translation from skML to IRIS Explorer map; the reserve transformation, saving IRIS Explorer map as skML has also been implemented.

Our approach to binding skML maps to physical resources uses more web technology, RDF (Resource Description Framework), to attach annotations to modules and links in the map. Annotations can take the form of resource constraints (a requirement for a particular kind of processor, say) or actual resources (this module is to run on a specified Grid resource). This conceptual approach has not been implemented in its full generality, but we have implemented a facility in the skML editor to assign modules to particular hosts and have linked this to the IRIS Explorer functionality for launching modules on specified hosts.

If a dataflow model for visualization is accepted at the conceptual level, we then have to consider what modules exist at the conceptual level and how these map into modules at the logical level. In our proof-of-concept work we have taken a pragmatic approach to this question, choosing a small set of modules to read data, compute an isosurface and render the resulting geometry and mapping these to corresponding modules in IRIS Explorer and Open DX. We use RDF annotations to indicate (in a limited sense) the meaning of each module. This points to the need for an ontology for visualization. The gViz project organised a workshop at NeSC in April 2004 to begin to explore what an ontology for visualization might be. A first sketch for the high level structure of an ontology resulted and further activity is being planned.

The skML work and the ontology work is important in the provenance of visualizations.

We can associate with a visualization an XML document describing how it was created – in such a way that it is not tied to any particular visualization system available in 2004 but has a lasting description.

4. Implementing Visualization in a Grid Environment

We have studied the implementation of visualization systems in a Grid environment. The traditional scenario is for an entire visualization session to be executed on the user's desktop machine, using a single piece of software. With Grids, we need to contemplate a distributed environment with heterogeneity in terms of both hardware and software.

A particularly important case occurs in computational steering, where we envisage a simulation component running autonomously on a remote resource, with some front-end visualization system running locally to view results and alter parameters. We have developed the gViz library to support the connection of these pieces of software. The library is in two parts: one part provides an API which the e-scientist can use to instrument their simulation code; the other part provides an API which allows matching capability to be integrated into the front-end visualization system. Our development environment for this is IRIS Explorer, but other systems could be used in its place (indeed we have also used SCIRun from the SCI Institute at Utah, and also vtk with qt).

The architecture is shown in Figure 3. Steering parameters are passed from the front-end system (here IRIS Explorer) through the gViz software to the underlying simulation. Results are passed back from the simulation through gViz software to the front-end system.

The library has been designed to work in diverse computing environments, including large clusters where normally only interaction with the head node is possible.

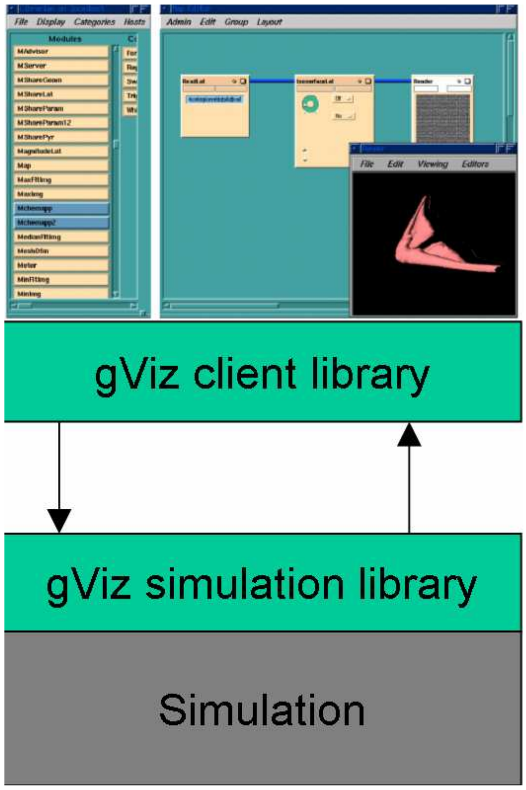

Figure 3 – Architecture of the gViz Steering Library

For homogeneous software in a Grid environment, simpler solutions can sometimes be achieved. In particular we have developed a Grid-enabled version of IRIS Explorer, in which modules are allowed to run on different resources, but the user interface remains on the desktop. Careful allocation of modules to resources can produce significant performance improvements. We can think in terms of designing the dataflow so as to maximize performance. These issues are being explored further in a new EPSRC project on Advanced Environments for Enabling Visual Supercomputing (e-Viz), in collaboration with the University of Wales at Swansea and Bangor, and the University of Manchester.

An advantage of using IRIS Explorer in this work has been the opportunity to exploit its collaborative facilities, allowing a geographically separated set of users to work together.

5. Demonstrators

We have created a number of demonstrators of computational steering and visualization in Grid environments, using both skML and the gViz library, and these have been shown at many national and international events.

We have demonstrated the use of the SVG Map Editor and skML for distributed collaborative visualization using IRIS Explorer and OpenDX as the visualization platforms. Figures 1 and 2 showed a map created in skML and the corresponding map in IRIS Explorer. Figure 4 shows the corresponding Open DX map. This was generated by transforming skML into the Open DX scripting language.

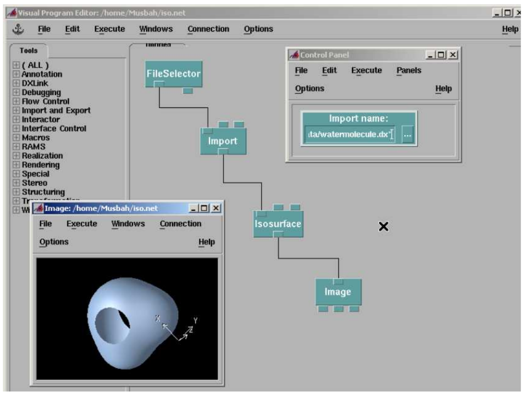

Figure 4 – Open DX map corresponding to Figure 1

We have also developed a collaborative visualization demonstration using skML. The demonstration was originally constructed by Jeremy Walton (NAG Ltd) for a State of the Art report on distributed and collaborative visualization given at the Eurographics 2003 conference (Brodlie et al, 2004). The demonstration involved two actors and involved a certain amount of manual editing of IRIS Explorer maps during the demonstration. The map files were re-expressed in skML as shown in Figure 5.

The skML map displayed in Figure 5 is in fact a collection of simpler maps (which we will refer to here as maplets), corresponding to particular steps in the demonstration. When this composite map is loaded into the skML module in IRIS Explorer, the user is presented with a list of the maplets. Loading a maplet causes IRIS Explorer to automatically make any necessary connections to existing modules and to create instances of any necessary utility modules (these latter are modules from the set of modules that support collaborative visualization in IRIS Explorer but are not explicitly included in any of the skML maplets). In a more general setting, the maplets could correspond to the roles that a user might take in a collaborative visualization session, thus providing a natural way for users to establish their participation in a session.

The skML map editor uses colour and transparency to distinguish different maplets within the composite skML map. This is useful when complex maps are being created and manipulated.

A demonstrator for the gViz computational steering library highlights an environmental disaster scenario, where a dangerous pollutant escapes from a chemical factory. The simulation models the dispersion, and the visualization allows the planners to predict where the pollutant is headed, so that evacuation plans may be put in place. This is a good illustration of computational steering in e-science, as we can experiment with different wind directions to run various ‘what-if’ scenarios. An illustration is shown in Figure 6, with IRIS Explorer as front-end. Notice the arrow widget by which the wind direction is manipulated.

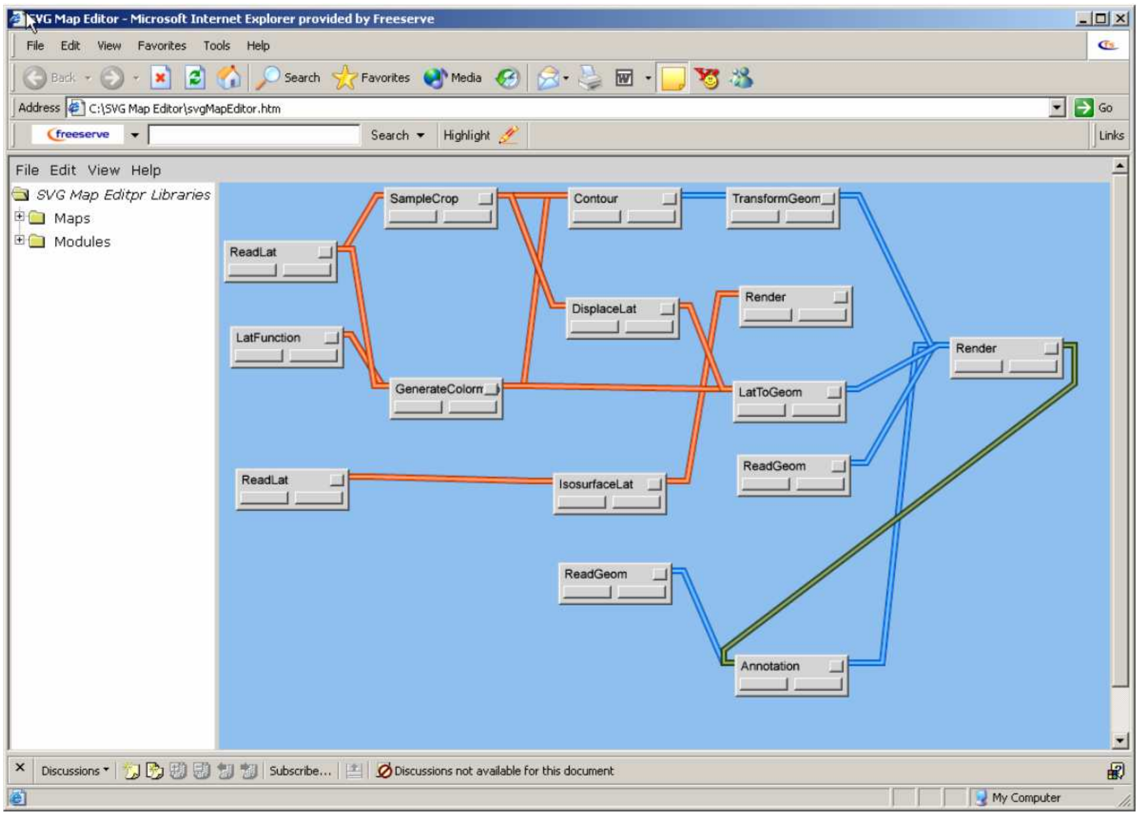

Figure 5 - skML map for collaborative visualization demonstration

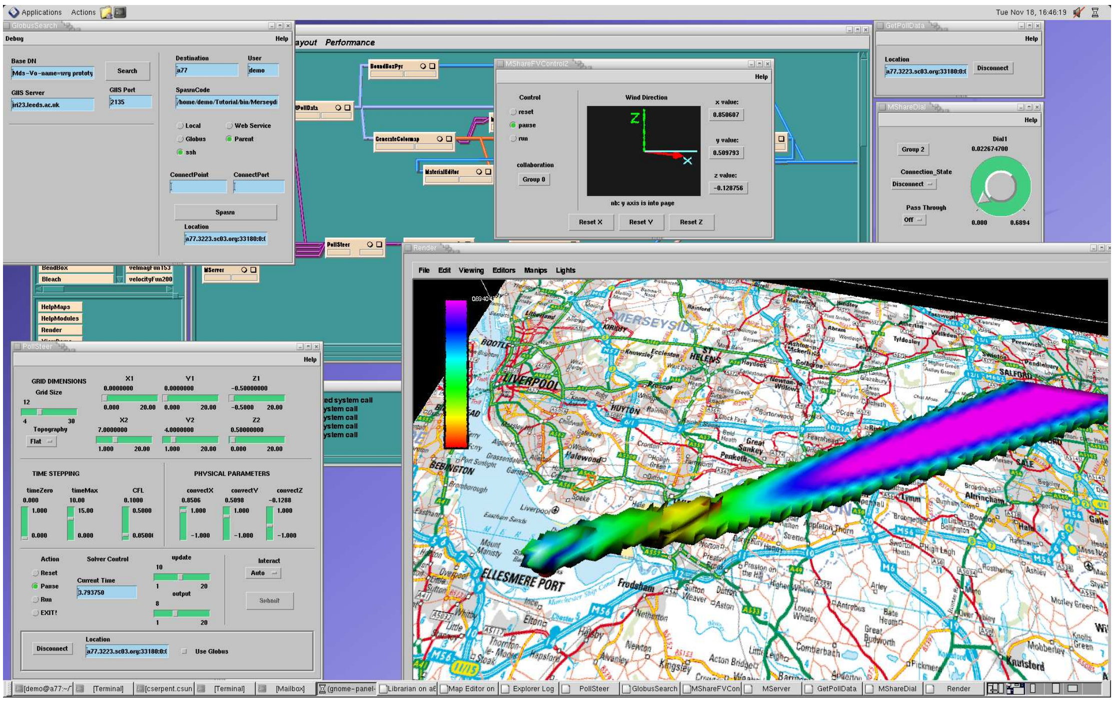

Figure 6 - Pollution Demonstrator using the gViz library with IRIS Explorer as front-end

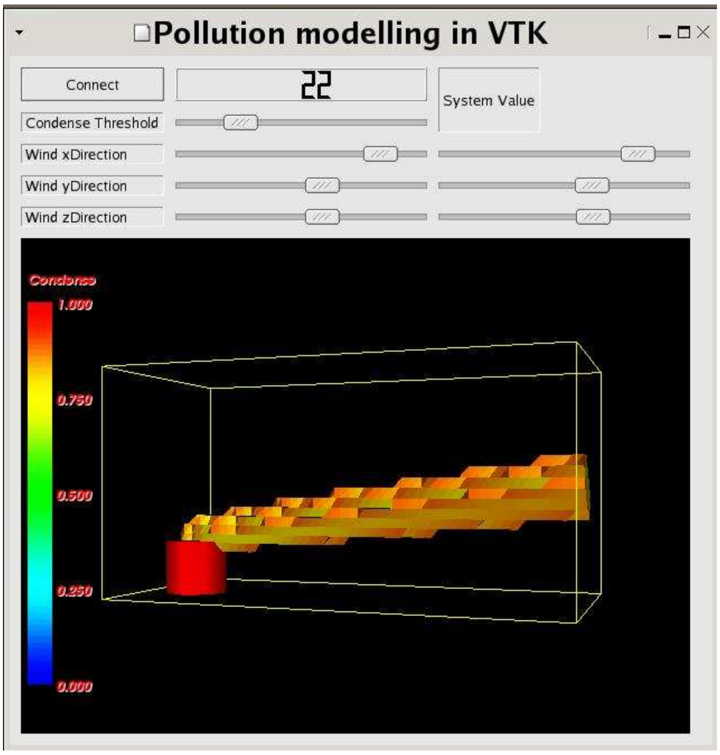

Figure 7 - Pollution Demonstrator using the gViz library with vtk as front-end

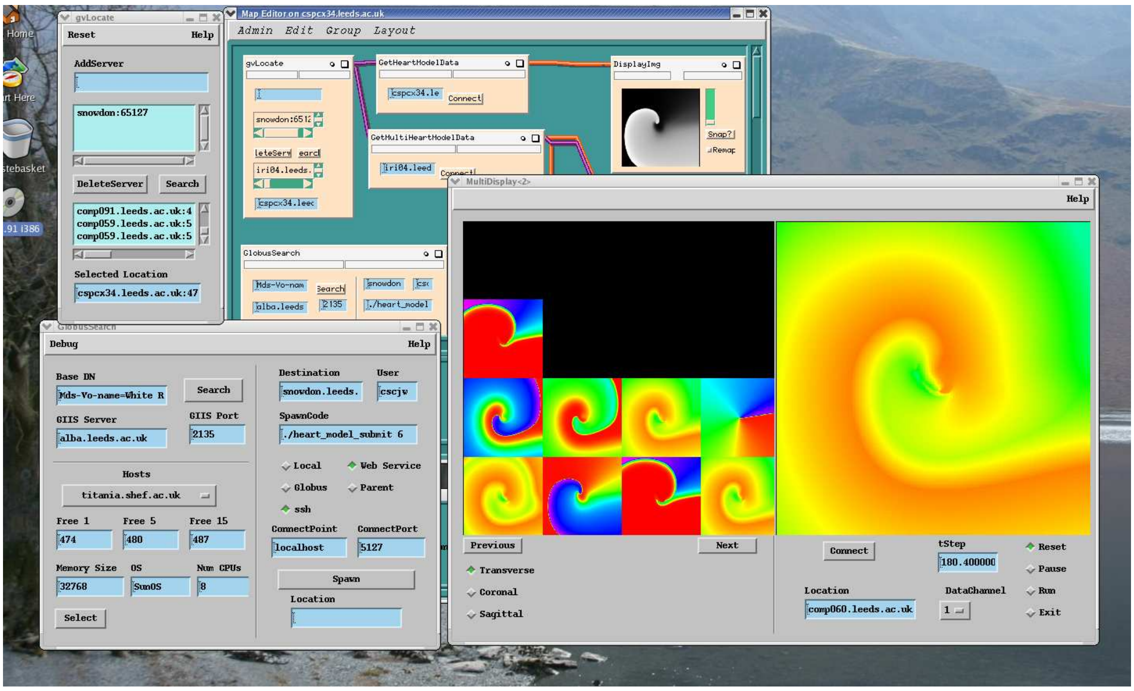

Figure 8 - Monitoring heart models

We have also used this demonstrator to illustrate collaborative visualization, with a meteorologist controlling the wind direction, a numerical modeller studying error values and an environmental scientist interpreting the results.

To highlight the independence of the gViz library from any particular visualization system, we have also built the demonstrator using the vtk toolkit, with qt providing the user interface. This is shown in Figure 7.

Another gViz demonstrator has application to computational biology, in collaboration with Professor Arun Holden (University of Leeds) and Dr Richard Clayton (University of Sheffield). These scientists are investigating the electrical behaviour of the heart, and in particular are simulating re-entrant arrhythmia, or irregular heartbeat. In this demonstrator, a number of simulations are set in operation, each running a heart model for a specific set of parameters. The simulations run on Snowdon, a 256-node cluster which forms part of the White Rose Grid. Each simulation runs on a different node of the cluster. Figure 8 shows a screenshot of the demonstrator in action. Notice that IRIS Explorer, acting as the gViz software front-end, provides a portal from which the different simulations can be monitored. The image shows the characteristic rotating spiral waves which are of particular interest to heart modellers.

Acknowledgements

Partners in the gViz project are the Universities of Leeds, Oxford Brookes and Oxford; CLRC; NAG, IBM UK and Streamline Computing. We would like to thank our colleagues in the project (Mike Giles and David Gavaghan, Oxford University; Julian Gallop, RAL; Steve Hague and Jeremy Walton, NAG Ltd; Brian Collins and Alan Knox, IBM UK Ltd; and Mike Rudgyard, Streamline Computing). At Leeds, we would like to thank our colleagues Chris Goodyer, Ying Li, Mark Walkley and Sean Wang in the School of Computing. Particular thanks to Arun Holden (University of Leeds) and Richard Clayton (University of Sheffield) who motivated the computational biology application and provided the heart modelling code.

References

Brodlie, K.W., Duce, D.A., Gallop, J.R., Walton, J.P.R. and Wood, J.D., 2004. Distributed and Collaborative Visualization, presented as State of the Art Report at Eurographics 2003, and to be published in Computer Graphics Forum, Vol 23, No 2, pp 185-213.

Duce, D.A. and Sagar, M. 2003. gViz – A Reference Model and Application of XML for Visualization and the Grid, Proceedings of UK All Hands Meeting 2003, pp242-245. Indexed at: http://www.nesc.ac.uk/events/ahm2003/AHMCD/index_authors.htm

Haber, R.B. and McNabb, D.A., 1990 Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In *Visualization In Scientific Computing*, Shriver, B., Neilson, G.M., and Rosenblum, L.J., Eds, IEEE Computer Society Press, 74—93.

Wood, J.D., Wright, H., and Brodlie, K.W. 1997. Collaborative Visualization. In Proceedings of IEEE Visualization '97. Yagel, R., and Hagen, H., Eds, 253-259.