

Multiple Window Visualization on the Web using VRML and the EAI

Jonathan C. Roberts[†], Rob Knight, Mark Gibbins and Nimesh Patel

*Computing Laboratory, University of Kent at Canterbury,
Canterbury, Kent, UK, CT2 7NF*

E-mail: [†] *j.c.roberts@ukc.ac.uk*

Visualization exploration is often aided through the use of multiple views. Such techniques enable the user to observe the visualization through different forms (multiform) and to navigate the visualization via different methods. The web provides a convenient publication medium for visualizations; allowing remote access for observation and investigation of the realizations. Thus, it seems wise to investigate Multiple View visualization on the web. Indeed, we have successfully used VRML, Java and the EAI to develop such web-based multiple view visualizations. We believe that multiple window techniques are useful to present visualization on the web, and may be useful for other virtual worlds.

Keywords: visualization, VRML, EAI, multiple views, multiform

1. Introduction/Motivation

In the real world there are many uses and advantages for using multiples. For example, a child may learn their 'multiplication tables' by repeatedly saying out-loud the figures to learn and understand their form. Additionally, it is by changing the order of the numbers and the way a question is phrased that the child may better understand such numerical manipulation and calculations. Such order adjustment could be classified as a different view on the same information. Consider the London Underground Map (LondonUnderground, 2000) - and indeed other similar network maps-, this map is shown in an abstract form where the exact positions of the stations are adapted to provide a clearer way to present the connection of the stations. The connecting rails are shown without the twists and turns that would have been shown using from a geographical map of the network. This underground map is clearer for one task: that of evaluating station connectivity, but makes it difficult to calculate the geographic distance between stations - something that requires a geographical map.

In the above written examples we can see that the different views enable the user to understand the information better, because, the different forms reinforce the

information (through duplication) and each presentation method allows the user to understand and examine the information from a particular standpoint or to investigate the data by using a specialist representation technique that displays a certain aspect of the underlying information.

We believe that there is much to be gained in using multiple views and different forms (multiform) for virtual exploration. Each view may be confined to achieve a specific operation. For example, an individual window may be used to represent the 3D visualization, allowing 3D manipulation and investigation; other views for 2D realizations, while additional windows may be used for menu-based commands. Such a separation of windows and tasks allows each window to have simple and specific user interaction methods.

Our aim is to have many windows holding different views of the same information that are all inter-linked and presented in individual windows. For example, one window may show a realistic view of a virtual world, another would represent the location of the user on a virtual map, the worlds would be coupled so that each view coincidentally updates as the person moves; additionally, as the user navigates and investigates the world related textual information could, for example, be automatically displayed in another window.

We believe it is possible to include such techniques with the publication of a 3D visualization on the web. This would allow the user to explore and investigate the data and better understand the information.

2. Background

In this research we are integrating two areas: Multiple views and web visualization. The data is visualized using the IRIS Explorer visualization package and the results are exported as Virtual Reality Modelling Language (VRML) models. Hand coded behaviours are added to the VRML using JavaScript and the Script nodes. These multiple VRML models and other Java visualizations are connected together using the External Authoring Interface (EAI) via Java Applets and embedded in an HTML document and presented in a web-browser. In the following section we briefly review some relevant techniques from these areas.

2.1 Multiple views

In a previous paper, we discussed how it is possible to publish 'alternative representations of the same information' in VRML on the web (Roberts, 1997). These different visualizations give archaeologists an opportunity to examine and compare alternative graphical-views; such multiple views are useful as the realizations are often generated from sparse data, experts knowledge and some supposition! In this research, each new representation *replaces* the previous. Indeed we stated that 'alternate views can be easily compared, because the navigation position of the browser is kept constant when the instances are loaded'. Multiple views are useful in other areas of visualization, not just archaeology (Roberts, July 1998) and even within archaeology it

may be useful to separate the information into multiple windows, allowing a side-by-side comparison and navigation of the alternate views.

The content of the multiple views may be similar or diverse but is usually derived from the same original information. The multiple realizations can be formed from the application of different visualization algorithms, or by changing the mapping algorithm on the data (as depicted in Figure 1) or may be the same visual form but rendered from different positions and orientations. There are many reasons for using multiple windows, we briefly list some of these reasons below. For more information about multiple views see "Multiple-View and Multiform Visualization" by Roberts, (2000).

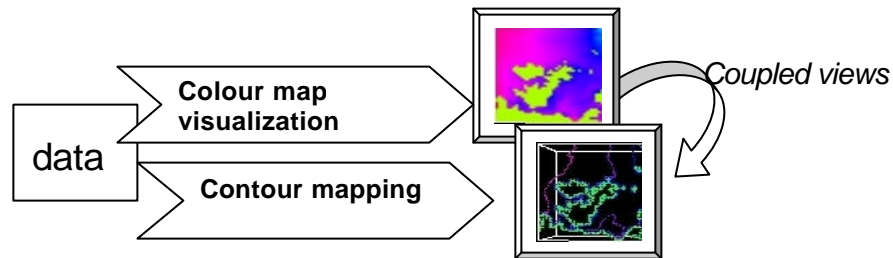


Figure 1: Different views may be generated by changing visualization parameters, or by using different visualization mappings. Once the views have been generated they may be investigated and manipulated together; then the views are said to be coupled.

Correct dissemination. Different visualization algorithms may be used to display the same data in multiple forms. This allows the user to see the information from different perspectives and may give the user a better understanding of the underlying data.

Comparison. The output of different algorithms, visualizing the same data, may be compared side-by-side, by difference images or data-level comparison (Shen, 1998).

Navigation. The multiple windows, if yoked or coupled together, may aid the user as they navigate the worlds. Yoking and coupling enables the same - or similar - operations to occur in every yoked window; for example, a rotation in one display would automatically cause the other yoked view to rotate similarly. Other coupled navigation methods exist including brushing, where, elements may be jointly selected by touching items in one view and having them highlighted in corresponding views (Ward, 1994).

Collaboration. Similar techniques are used for systems that allow remote collaboration; a difference from the systems that provide multiple views is that the displays are remote rather than being on the same screen. People have extended popular visualization systems to accommodate the collaborative tasks, for example, Klinker (1993) used AVS and Wood (1997) used IRIS Explorer in their collaborative visualization.

2.2 *Web visualization*

Over the past few years, with the general growth of the World-Wide-Web, we have seen more visualizations being published over the web and different publication methods being used. These methods, as presented by the seminal paper by Wood et al (1996) can be divided into four ways of publication: (1) static 2D images pre-defined by the publisher, (2) dynamic 3D models where the user can navigate through the 3D world, (3) dynamic 3D models served across the internet from a database of results and visualized in a form chosen by the client, and (4) the publisher creates and publishes a visualization framework that includes the realization and methods to investigate the information.

Formats other than VRML may be used to publish such visualizations over the web. For example, the VizWiz (Michaels, 1997) program uses Java to display users data that is uploaded to the server. Mikael Jern (2000) presented the OpenViz™ technology allowing business information to be displayed on the web. OpenViz is a component based framework that supports COM/ActiveX and JavaBeans component architectures. Indeed, with OpenViz the information may be displayed in multiple formats on the web.

2.3 *VRML, Java and EAI*

VRML models, as Walton (1997) said, provide a convenient way to display different visualizations. Indeed, they provide a good way to publish and embed the graphics within a web-browser. In such models the user may navigate and view the visualizations. VRML allows complex behaviours to be developed, that adapt and change the virtual world. These behaviours may occur external to VRML by a Java applet attached through the EAI; this allows the external applet to send and receive information to control the virtual world.

The Java Applet obtains a *handle* on the VRML browser. Events may be sent and received from the applet to the world, additionally, Java *callback* methods may be registered to run specific code when an event occurs in the VRML model. For example, El-Khalili and Brodlie (1997) utilised the EAI to communicate between two separate virtual worlds.

In any VRML/Java/EAI project a decision needs to be made as to how much of the behaviour is controlled by the external Applet and how much by the VRML code itself (Marrin, 1997). We use the EAI and a Java Applet to manage the events to jointly manipulate multiple VRML windows. A Java Applet receives events from the VRML windows and sends the events to the coupled windows. The VRML code includes behaviours to control the users navigation and exploration of the visualization. The Applet manages which VRML window and which VRML behaviour receives what events, it also manages the frequency of the events to control how often the related models are updated.

3. Examples

We present three examples of Multiple window visualization using VRML and the EAI.

3.1 VOVA

VOVA is a project looking at the 'Visualization Of Visualization Algorithms'. An example is an algorithmic demonstrator for the visualization of the Marching Cubes surface generation Algorithm (Lorensen, 1987). This algorithm generates an isosurface over a 3D dataset by evaluating a local set of data values for their contribution to the surface.

Our visualization displays two VRML windows with one window (A) depicting a visualization of the position of the current voxel being evaluated, and another VRML window (B) showing the end result (a simplified wire-frame version of the isosurface). There are four visible Java Applets used: (1) a control applet with an interface allowing the user to select the dataset, threshold and speed of animation; (2) an applet that highlights the current major case being used and displays a count of how many times each case has been used; (3) a code Applet that highlights a fragment of pseudo code, showing the progression of the algorithm through the code; and (4) a Status applet that describes the current process (see Figure 2). A fifth applet - not visually seen - is used to manage the whole application.

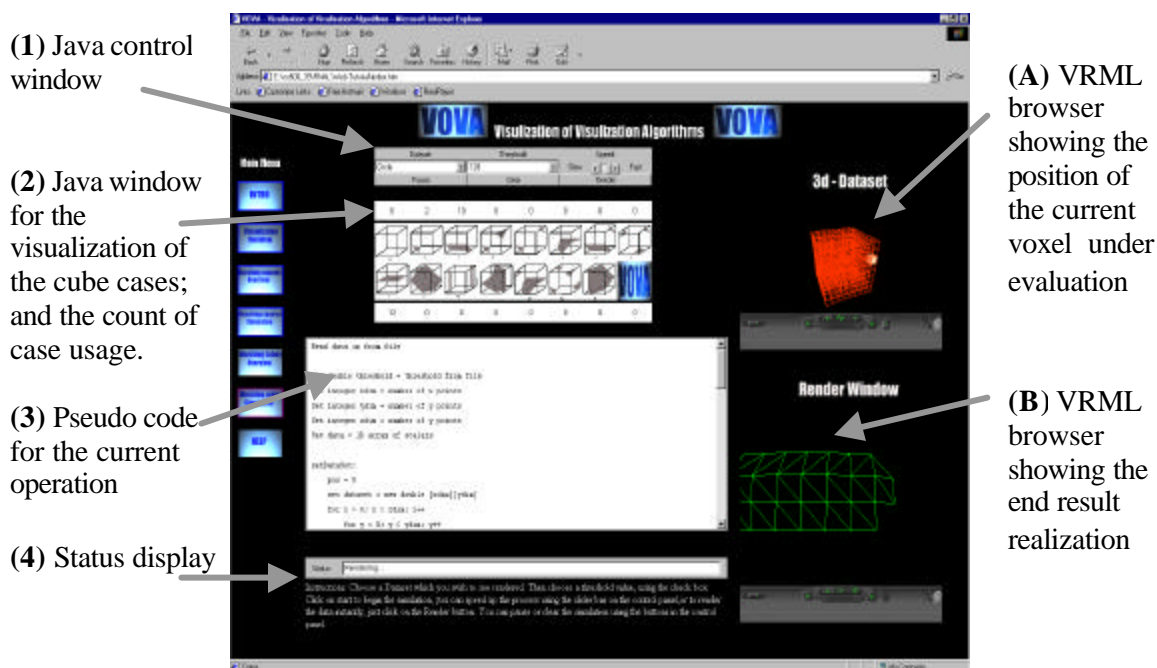


Figure 2 An algorithmic demonstrator for the visualization of the Marching Cubes Algorithm (VOVA).

We have used inter-applet communication to manage the five applets in this application. This allows us to obtain a handle on each of the applets running in the current browser-context and then treat the instance as any other class. The position of the current voxel being evaluated is translated appropriately as the march occurs through the 3D dataset (A), and geometry for the end visualization is progressively added to the result window (B) as the animation executes.

3.2 Multiple view visualization on the web

This example demonstrates linked Java and VRML visualizations, Figure 3. The visualizations show two cutting planes through the data that dynamically change as the user moves the large arrow. Each of the pointers in each window is linked together, and a change in the pointer position from any window dynamically updates the position in the others. This enables the user to navigate round the virtual world, observe the seabed and investigate the data.

This linked operation is managed by a single Java Applet. This is accomplished by

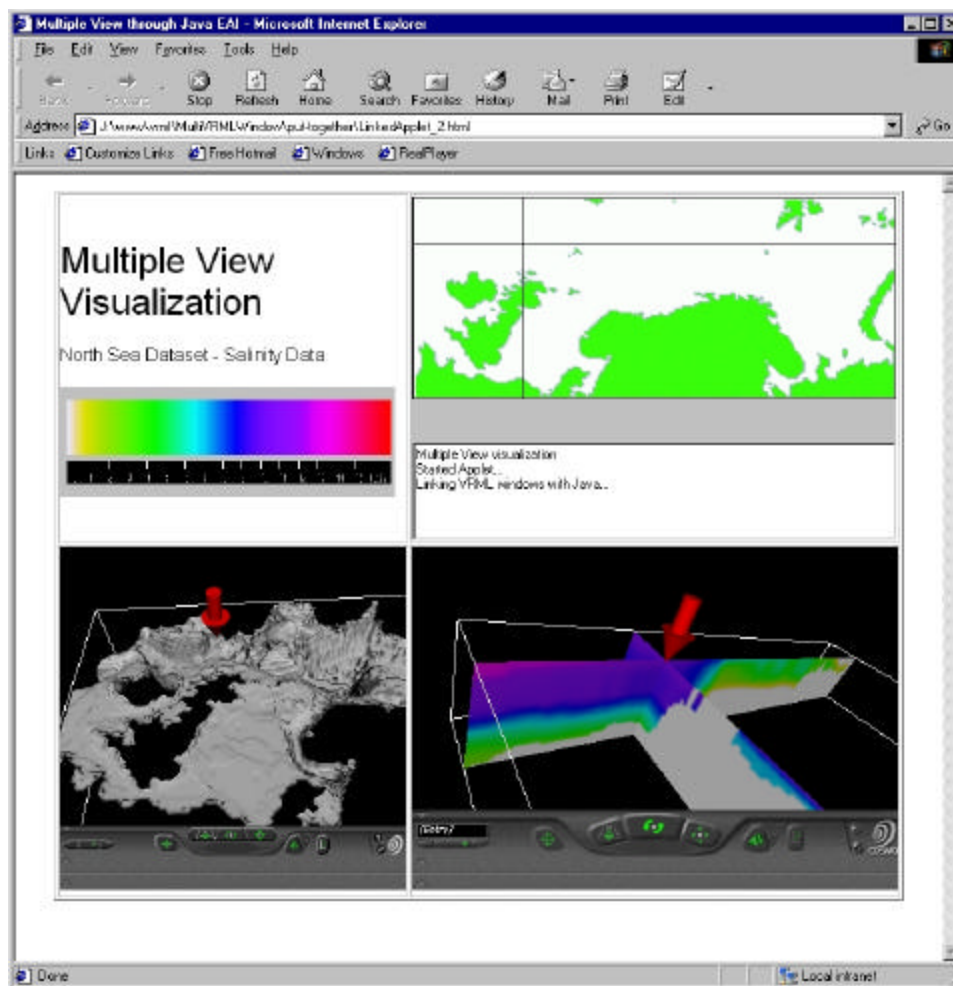


Figure 3 Multiple view visualization of Salinity data, using two VRML windows and a Java Applet, communicating through the EAI.

attaching a MouseListener to the Java Applet to recover the mouse position from the 2D map visualization, that in turn outputs positional events to the other VRML models. The reverse operation also occurs: with the position on the 2D map visualization being updated when the position of the marker in the VRML window is changed, which is achieved through registering callbacks to the pointer behaviour in the VRML worlds, the called methods then update the Java and other VRML window, respectively.

Each VRML model is self contained, and may be loaded on their own, as they include the behaviours to control the visualization exploration as separate worlds. For example, the VRML model that displays the ocean planes, from a given x,y coordinate, includes the behaviour to constrain the navigating arrow and load the appropriate slices.

3.3 Many Multiple Linked Worlds

This final example, Figure 4, shows that it is possible to design a web page with multiple VRML windows, that are coupled together and controlled by a single Java applet.

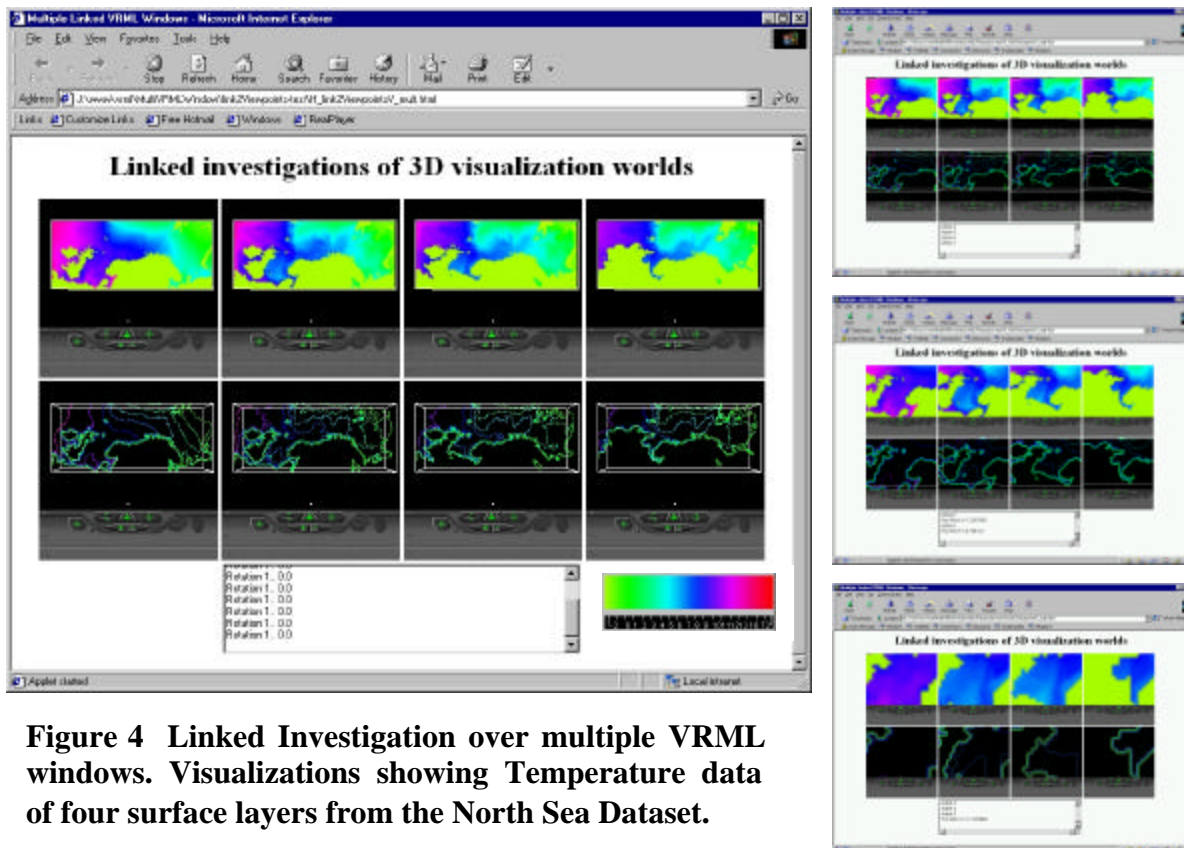


Figure 4 Linked Investigation over multiple VRML windows. Visualizations showing Temperature data of four surface layers from the North Sea Dataset.

As before, each of the VRML worlds are linked together and controlled from the Java Applet via the EAI. The current position of the viewer, in the 3D world, is detected by a ProximitySensor. The Java Applet attaches callbacks on these sensors, the callback method then sends updated viewpoint information to the bound Viewpoint node in the scene.

This is a traditional Model, View, Controller design, with the Java Applet holding a copy of the model. In practice we found that many events were being passed around from all these callbacks, indeed, we had an explosion of events! Thus, we wrote some filter code that reduced rate at which the coupled views were updated and made sure that no events went back to the original controller view. It may be better to detach the callback methods from all but the controlling view when the mouse enters the window; but this would generate a slight pause, before the user can manipulate a view, when the user moves from window to window.

4. Evaluation/Discussion

In these multiple-view publications we decided to bundle the data (in the case of VOVA) and the visualizations (in the other examples) with the published realizations; rather than dynamically rendering them from a visualization server (Wood 1996). Sometimes it is better to publish solely the visualizations as the publishers do not wish to give away the data (even if it can be encrypted in the publication). However, a slower startup time is achieved when bundling the visualizations as the visualization works when the data is completely retrieved. It would have been better to integrate some sort of selective loading, when the visualizations are actually required.

Immersion may be another problem with such multiplicity. 3D Visual exploration is often deliberately setup to immerse the user in a virtual world, either using specialist stereoscopic displays or a flat screen; they are designed in such a way as to convince the user to feel that they are *immersed* in the given world. Depending on the level of immersion required, it could be argued that immersion is inhibited when the user needs to look at multiple windows. Thus, it may be more effective to have one world containing many multiple visualised models, rather than separate windows, as demonstrated in one of the example models in (Roberts, September 1998).

Following from immersion, navigation is another key factor in both virtual world and visualization investigations. The multiple windows do aid the user by providing specific task oriented windows. For example, it is often easier to navigate in a straight line using a two dimensional interface that is constrained to only allow commands that operate along a plane, than it is to try to navigate in a like movement in an unconstrained 3D world.

5. Conclusion

Multiple views are useful for investigation of visualizations over the web. The simultaneous views enable the virtual world to be controlled and investigated by different user-interface methods through the many windows. We believe that such separate multi-window virtual control is useful for visualization and for the exploration of other virtual worlds. Our examples show that multiple windows and the coupling of views is feasible in a desktop virtual environment through the use of the EAI and inter applet communication.

6. Acknowledgements

I acknowledge Jarle Berntsen, University of Bergen, Norway, and the Marine Institute, Norway for the North Sea Dataset. I thank Magali Raynaud for proof reading the document.

7. References

N.H. El-Khalili and K.W. Brodliie, "Distributed VR Training System for Endovascular Procedures", edited by Richard Bowden, in Proceedings of Fourth UK VRSIG Conference, pages 110 -119. 1997.

Mikael Jern, "Collaborative Visual Data Navigation on the Web". Proceedings Information Visualization 2000. Pages 268-273. IEEE Computer Society, 2000.

Gudrun J. Klinker. An environment for telecollaborative data exploration. In Proceedings Visualization '93, pages 110-117. IEEE Computer Society Press, 1993.

London Underground Maps, 2000, <URL: http://www.londonunderground.co.uk/mau_maps.shtml>

William E. Lorensen and Harvey E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", ACM Computer Graphics, Vol 21, No.4, 1987. Pages 163-169.

Chris Marrin, Jim Kent, Dave Immel, Murat Aktihanoglu. "Using VRML Prototypes to Encapsulate Functionality for an External Java Applet". Electronic Publication. 1997.
< http://www.marrin.com/vrml/papers/InternalExternal/Chris_Marrin_1.html>

Cherilyn K. Michaels and Michael J. Bailey, "VizWiz: A Java Applet for Interactive 3D Scientific Visualization on the Web. Proceedings IEEE Visualization '97. Pages 261-268. IEEE Computer Society Press. 1997 < <http://www.sdsc.edu/vizwiz/> >.

Jonathan C. Roberts and Nick Ryan. "Alternative Archaeological Representations within Virtual Worlds". In Richard Bowden, editor, Proceedings of the 4th UK Virtual Reality Specialist Interest Group Conference - Brunel University, pages 179-188, Uxbridge, Middlesex, November 1997.

Jonathan C. Roberts "On Encouraging Multiple Views for Visualization". In Ebad Banissi, Farzad Khosrowshahi, and Muhammad Sarfraz, editors, IV'98 - Proceedings International Conference on Information Visualization, pages 8-14. IEEE Computer Society, July 1998.

Jonathan C. Roberts "Publishing Time Dependent Oceanographic Visualizations using VRML", In Simon Grange editor, Proceedings of the 5th UK Virtual Reality Specialist Interest Group Conference, September 1998.

Jonathan C. Roberts "Multiple-View and Multiform Visualization". In Robert Erbacher, Alex Pang, Craig Widdenbrink and Jonathan Roberts, editors, Visual data Exploration and Analysis VII. Proceedings of SPIE, volume 3960, pages 176-185. IS&T and SPIE, January 2000.

Qin Shen, Alex Pang, Sam Uselton "Data Level Comparison of Wind Tunnel and Computational Fluid Dynamics Data", Proceedings IEEE Visualization '98. Pages 415-418. IEEE Computer Society Press, 1998.

OpenViz. Advanced Visual Systems, 2000. < <http://www.openviz.com> >

Jeremy Walton and David Knight, "Rock`n`Roll: Using VRML 2.0 for Visualization", Eighth Eurographics Workshop on Visualization in Scientific Computing, 1997.

Matthew O. Ward. "XmdvTool: Integrating multiple methods for visualizing multivariate data". In proceedings Visualization '94. IEEE Computer Society Press. Pages 326-333. 1994.

J. Wood and K. Brodliie and H. Wright, "Visualisation over the World Wide Web and its Application to Environmental Data", Proceedings IEEE Visualization '96. Pages 81-86, 1996.

Jason Wood, Helen Wright and Ken Brodliie, "Collaborative Visualization", Proceedings of IEEE Visualization 1997 Conference, edited by R. Yagel and H.Hagen, pp 253-260, ACM Press.

Jonathan C. Roberts, Rob Knight, Mark Gibbins and Nimesh Patel. "Multiple Window Visualization on the Web using VRML and the EAI". Proceedings of the Seventh UK VR-SIG Conference, Robin Hollands (Editor). September 2000, Pages 149-157.