Scientific Visualization - Some Novel Approaches to Learning

Ken Brodlie, Jason Wood and Helen Wright

School of Computer Studies
University of Leeds
Leeds, UK

{kwb,jason,helenw}@scs.leeds.ac.uk

Abstract

Scientific visualization is becoming an important part of the curriculum in a number of disciplines. It is a very practical subject, but the commercially available visualization software systems are not easy to learn. Thus we have explored the use of novel technology to help us in teaching students to use IRIS Explorer - this includes on-line tutorials, shared sessions involving teacher and student over a network, and the use of WWW. This work, which has been stimulated by teaching applications, has also motivated research into the wider area of collaborative visualization.

1. Introduction

Scientific visualization is now an important component of scientific and engineering computing. Indeed it is recognised as the only means of understanding the large datasets which now occur routinely - whether by measurement in the case of medical scanners or satellite photography, or by simulation in the case of chemical kinetics or computational fluid dynamics. Thus visualization as a discipline is starting to appear in computer science curricula, at both undergraduate and masters level. It is a very practical subject, and an important aspect of any visualization teaching must be practical sessions and coursework assignments. In this paper, we describe our experiences in teaching students how to use modern visualization systems, as preparation for them undertaking practical assignments. Some of the constraints we work under however mean that we cannot always be present at the same time and the same place as the student needs help - our efforts to use new technology to overcome this are the subject of the paper.

2. The Context - A Masters Programme in Vision, Visualization and Virtual Environments

At Leeds University, the School of Computer Studies offers a masters programme entitled ‘Vision, Visualization and Virtual Environments’. This includes a 12 week module on visualization covering the subject in some depth. The module includes the following topics:

- historical background and applications
- reference models for visualization
- visualization techniques
- rendering techniques
- visualization systems
- data management
- problem solving environments (where simulation and visualization are combined in a larger overall system)

This last topic is important because the module is also taken by students following the MSc programme in `Computational Fluid Dynamics’ (CFD), where the ability to create a single problem solving environment, in which the fluid dynamics computation and the visualization are integrated, is of fundamental importance.

A key part of the course is the practical work - indeed the assessment is split equally between coursework and written examination. Here we face a dilemma familiar to anyone teaching a practical aspect of computing. Should one use specially developed educational software as the basis - the advantages are simplicity, ease of use, and often the opportunity for the student to edit the source code to try out modifications... but there is the major disadvantage that the software may bear little resemblance to the visualization systems the students will meet when they leave university. Or should one base the practical work on a commercial visualization system - this may well be harder to use, and it is unlikely that a student can gain

access to the source code... but it will train the student rather better for visualization in the 'outside world'.
It is easy to make a convincing case for either alternative. Rightly or wrongly, we have followed the second approach, and our practical work is based on a state of the art, commercial visualization system: IRIS Explorer [2], originally developed by Silicon Graphics, but now developed and distributed by NAG Ltd.

Of course, having taken this decision, we now have to accept that a significant portion of the course must be devoted to teaching the students how to use a fairly complex piece of software. Indeed the arrangements for practical work are a challenge. The CFD students are based in the Chemical Engineering Department, not Computer Studies: the times when both sets of students are each free to attend practical classes is very limited. Moreover, the CFD students prefer to use workstations in Chemical Engineering (half a mile away!) - and indeed we prefer them to use their own equipment rather than encroach on our scarce computing resources!

Thus our teaching needs to allow for students learning - at different times, and in different places. Yet we as teachers have not the resources to cover a large number of different practical sessions. Our solution is described in the next section.

3. IRIS Explorer - Collaborative Learning

3.1 IRIS Explorer

First, we need to give a brief outline of IRIS Explorer itself. It is one of a number of similar visualization systems - AVS [5], IBM Data Explorer [3] and Khoros [7] are three others. They are based on the observation of Haber and McNabb [1] that the visualization of a set of scientific data can be broken down into a number of sub-processes. These will include:

- **input** of the data
- manipulation of the data - for example, to extract a subset of interest - known as a **filter** process
- transforming the data to some geometric representation - for example, creating contour lines, and perhaps assigning notional colours to represent different contour heights - known as a **mapping** process
- rendering the geometry as an image - this is the computer graphics part and is known as a **render** process

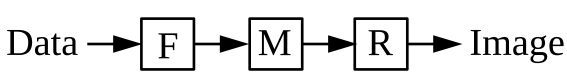

Figure 1: Haber-McNabb pipeline

These sub-processes form a pipeline along which data passes, en route from the source data to the final image - see Figure 1. The visualization systems provide libraries of these sub-processes, or **modules** as they are called. Visual programming is used to launch modules into an editing canvas, and to connect them in appropriate pipelines. An attraction is that the pipelines can be quickly assembled, and then modified - the modules being of fairly coarse granularity so that typically only a few are needed in any visual program. Figure 2 shows an example of an IRIS Explorer **map**, the term used to describe a particular pipeline. Notice that each module has its own user interface: the widgets on the module control parameters - for example, the number of contour levels to be shown.

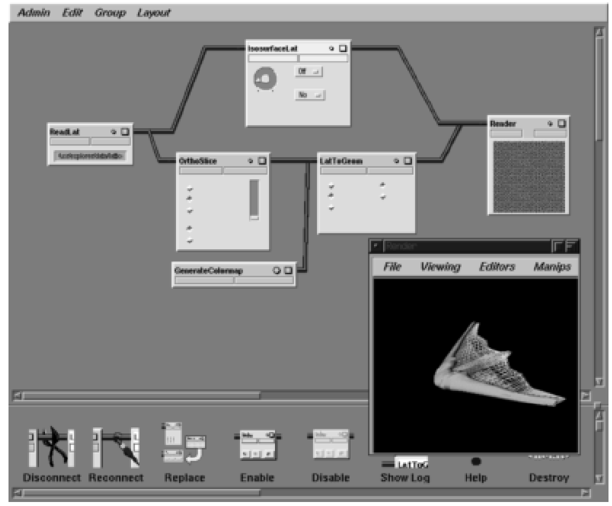

Figure 2: Screen shot of IRIS Explorer

The systems are non-trivial to use: for example, Lee et al [4] report that two hours were needed for two novice users to create a simple visualization network in IBM Data Explorer. Students need help in learning how to launch modules, how to connect them, and how to manipulate the widgets on the module interfaces. There are some two hundred modules in IRIS Explorer, so students need to understand the range of facilities available to them. This all takes time, but we need to get the students up and running as early as possible in the course.

3.2 On-line tutorial

An obvious first step is to provide the students with an on-line tutorial - see Figure 3. This allows the students to work at their own time, in their own place, at their own speed. The tutorial comes in three parts - each covering material we hope a student can assimilate in one week. By the end of the third week, the student should be an 'expert', and ready to tackle an assessed piece of coursework.

This works well, for most students, for most of the time. The CFD students can work on the Chemical Engineering workstations; the computing students can work in the Computer Studies labs. But what happens when a student gets stuck? The tutorial does lead the student through the different operations, but does not provide **all** the answers - otherwise the student is never challenged. So we need to provide support to students while they are learning from the tutorial.

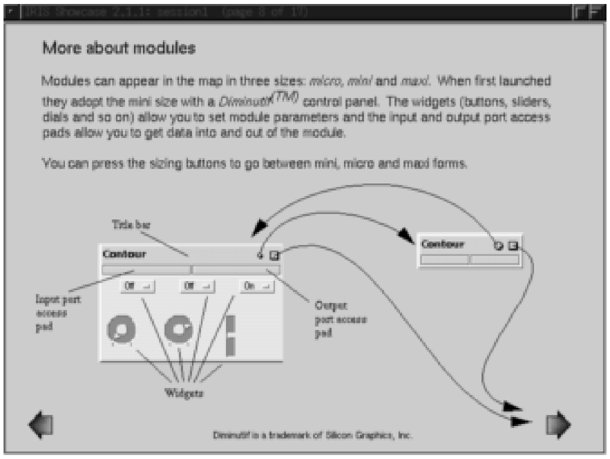

Figure 3: Screenshot of the tutorial

3.3 Collaborative Learning of IRIS Explorer

While on-line teaching materials can provide obvious benefits, there remains a need for human involvement in the learning process. Here we would like the student to be able to seek help, just as in any practical class. But if the practical class is at distributed locations, we have to accept that this help will be given over a network connection, rather than face-to-face.

Modern CSCW technology will play a key role in this - indeed, surely learning is a very natural example of co-operative working: in a good learning environment, the teacher and student work closely together. Thus we can employ desk-top video conferencing and shared whiteboard to allow the teacher and student to develop a solution to the student's problem. In our particular environment, both Chemical Engineering and Computer Studies have Silicon Graphics workstations - hence the SGI InPerson toolkit can be brought into play to provide the CSCW tools needed.

But what about the visualization system we are trying to teach - IRIS Explorer? How does that fit into a CSCW model?

Immediately an obstacle is hit. As with all other commercial visualization systems, IRIS Explorer has a single user interface. Individual modules can be distributed on different processors - indeed this is an important feature of the system allowing computationally intensive modules to be run on a supercomputer, and rendering modules on a graphics workstation, say - but the user interface cannot be distributed. Thus student and teacher need to cluster around a single workstation in order to work together - see Figure 4 - not what we want!

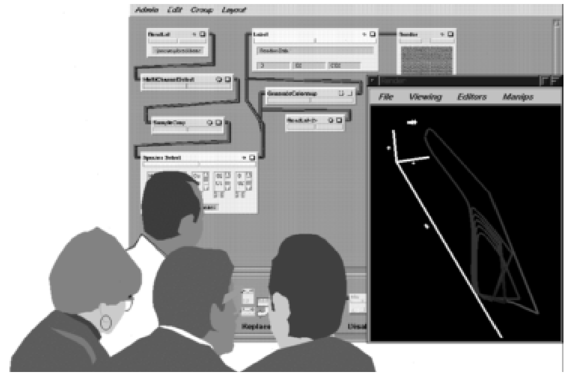

Figure 4: Clustering around single user interface

To support collaborative learning, we need a means of allowing teacher and student to run IRIS Explorer sessions independently, but be able to exchange data and pictures. Thus the teacher should be able to send a dataset to the student for them to visualize; and the student can send an image back to the teacher for checking. Indeed the teacher might want the student to send back data at any point in the pipeline for checking - perhaps after it has been filtered in some way, or after a geometric model has been created. IRIS Explorer has a sophisticated colour map tool, which assigns colours as RGB triples to given data values - the teacher may wish to send the student a colour map to use initially, before the student begins experimenting himself. We move from the 'same place' interface model of Figure 4, to a distributed collaborative environment shown in Figure 5.

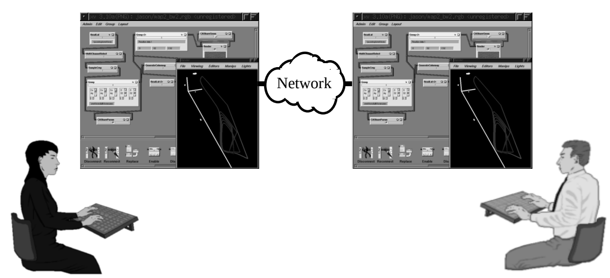

Figure 5: Distributing control of the user interface

To achieve this, we have developed a new set of IRIS Explorer modules which exchange data between different sessions - the data can be in the form of images, geometry, colour maps, source data, indeed any IRIS Explorer datatype. In fact, it is also possible to share parameter values. This allows the teacher and student to have dual control over the widgets on a module interface: thus the teacher can select for example the number of contour levels that a student will display in his pipeline. We have effectively extended the single pipeline model of the conventional visualization systems (as shown earlier in Figure 1), to the more powerful, multi-pipeline model shown in Figure 6. Data can be exchanged between pipelines as shown, and the modules are under the potential control of either person - teacher or student.

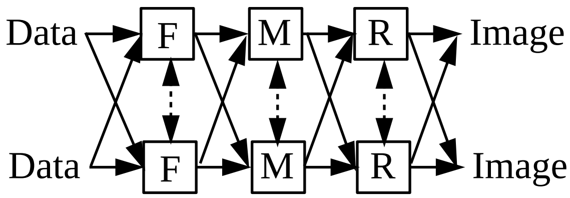

Figure 6: Extended Haber-McNabb model

In some instances, the student may be unsure of which module to select. Thus we also allow the teacher to launch a module in the student's map - indeed, if the student has no idea how to begin, the teacher can launch an entire map of connected modules on the student's machine.

In summary, teacher and trainer can share data, share control, share modules and share maps.

4. Practical Visualization Using World Wide Web

There is clearly a major investment of effort, on the part of both student and teacher, if one is to include the use of a system such as IRIS Explorer in a course on visualization. In a shorter course, this is a luxury one cannot afford. Indeed some would argue that simply to understand the ideas behind visualization techniques such as contouring, or volume rendering as used in medical imaging, the student should not have to learn the complexities of wiring modules together.

Thus one is led to a quite different approach. Here one wishes the student to see the picture produced by a particular algorithm, and allow them to experiment with different parameters of the technique, but to do this through a very simple and familiar interface. For example, for three dimensional scalar data, it is often useful to extract surfaces of equal value - called **isosurfaces**. These are the 3D analogue of contour lines for 2D scalar data. They could display, for example, equi-potential surfaces around a molecule. One would want the student to see an isosurface, to perhaps rotate it and zoom it, and then to vary the equi-potential level to see if this increased or decreased away from the centre of the molecule.

How to achieve this? The one user interface that every student today knows is the web browser interface, whether Mosaic or Netscape. Thus we have developed a visualization teaching facility according to the model shown in Figure 7. The student enters the details of the visualization on a form presented via the web browser: they enter the choice of technique (isosurface), parameters of the technique (isosurface level) and particular dataset (potential field data for some molecule). The web server runs a CGI script which passes the form details to a 'visualization web server', which in turn creates an appropriate IRIS Explorer map (using the scripting language of IRIS Explorer called Skm). Rather than generate an image, IRIS Explorer is programmed to create a geometry file representing the visualization; this geometry file (an Open Inventor **iv** file) is translated to the Virtual Reality Modelling Language (VRML), which is the WWW 3D graphics language. The VRML file is passed back to the web browser which invokes a suitable VRML reader (such as WebSpace) to display the visualization. Because live 3D geometry is passed back, rather than a passive image, the student has the freedom to interact with the visualization, rotating it as they wish in order to understand better the 3D nature of the equi-potential surface. They can explore it further by changing parameter values on the form, and seeing the resulting effect.

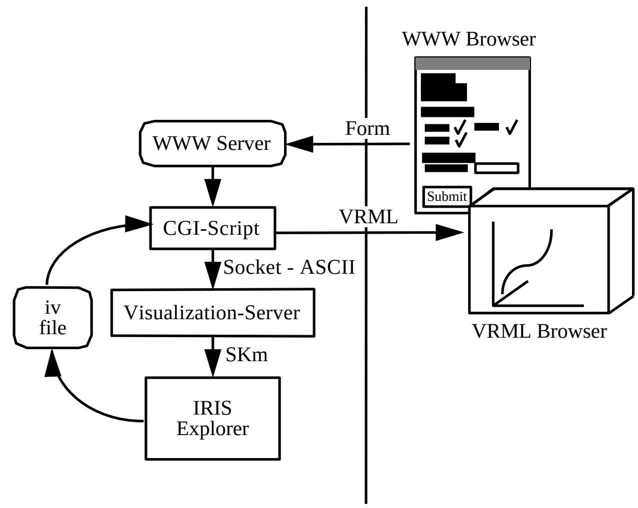

Figure 7: IRIS Explorer visualization web server

This, as we mentioned earlier, gives a facility that a teacher can use in any short course on visualization - or where visualization is included as a short part of a more traditional course on computer graphics. It can even be used in the early stages of a full course on visualization, before exposing students to the complexities of a visual programming system.

5. Conclusions and Future Work

We have seen how new technology - CSCW and WWW - can be employed to help students learn the practical aspects of scientific visualization. The work described here has been trialled in our visualization module in the MSc course. The on-line tutorial is now an essential part of the course, and allows us to handle distributed practical sessions without any increase in staffing. The collaborative learning using the extensions to IRIS Explorer has been used successfully with selected students. In its one-on-one mode, it is likely to be of benefit to students with special problems, but this individual style of teaching will not scale up to a large class. However the technology will also support a mode in which the lecturer acts as master, and drives a number of student workstations as slaves. These slaves can be detached at any time and asked to tackle an exercise independently, then joined up later with the lecturer's workstation to check against a working solution.

The IRIS Explorer web server is being trialled in our module on virtual working environments, where students learn about technology for remote delivery of learning materials.

The on-line tutorial and the visualization web server can be used at any time, at any place, by the students; the CSCW approach to collaboration between student and teacher obviously requires the same time frame, but is independent of the location - the only requirement is that student and teacher be connected by network.

It is sometimes argued that research represents the leading edge in university activity, while teaching follows some way behind, picking up research ideas and including them within curricula only after a period of maturation. But here there are signs that teaching is motivating research.

First, there is growing interest in the research community in the area of collaborative visualization systems, as described in section 3: for example, the European-funded PAGEIN [8] project has looked at collaborative visualization in the aerospace industry. Our own research has been given an extra spur by this teaching application, leading us to propose a significant extension of the Haber-McNabb visualization reference model - an extension encapsulated in the diagram of Figure 6, and described fully in [6]. The benefits of collaborative visualization are likely to be significant: software vendors can use this technology in the support of, and training in, visualization systems; research teams of scientists and engineers can use the techniques to co-operate on the analysis of simulation and experimental data - without the requirement to physically meet together.

Similarly, the visualization web server opens up a wealth of opportunities for further development. One can envisage now the ability to post-process any data (of suitable form) anywhere on the World Wide Web, to present it to the reader as a picture rather than an indigestible list of numbers - stock market data, air pollution data, meteorological data and medical data are only a few of the possibilities.

References

[1] G. Abram and L. Treinish. An extended data-flow architecture for data analysis and visualization. *Computer Graphics*, 29(2):17-21, 1995.

[2] D. Foulser. IRIS Explorer: A Framework for Investigation. *Computer Graphics*, 29(2):13-16, 1995.

[3] R.B. Haber and D.A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In B. Shriver, G.M. Nielson and L.J. Rosenblum, editors, *Visualization in Scientific Computing*, pages 74-93, IEEE, 1990.

[4] K. Lee, J. Ni, T. Halverson, E. van Wyk and J.R. Brown. Is an MVE the right environment for your visualization application. *Computer Graphics*, 29 (2):49-51, 1995.

[5] H.D. Lord. Improving the application development process with modular visualization environments. *Computer Graphics*, 29(2):10-12, 1995.

[6] J.D. Wood, H. Wright and K.W. Brodlie. CSCV - computer supported collaborative visualization. In *Proceedings of BCS Displays Group International Conference on Visualization and Modelling*, 1995.

[7] M. Young, D. Argiro and S. Kubica. Cantata: Visual programming environment for the Khoros system. *Computer Graphics*, 29 (2):22-24, 1995.

[8] CSCW in AVS and IRIS Explorer. ONERA: Office National d'Etudes et de Recherches Aerospatiales. *http://visu-www.onera.fr/ONERA/onera.html*. 1995

Acknowledgements

Our thanks to EPSRC who have funded the COVISA project at Leeds. We are also grateful to NAG Ltd, developers of IRIS Explorer, for their assistance. Finally, we thank students of the MSc class for acting as willing testers of our work.