UKHEC

Scientific Application of OpenGL Vizserver within High Performance Computing

Briefing Paper Joanna Leng and Nigel W. John
Manchester Research Centre for Computational Science

October 2002

Synopsis

The aim of this report is to provide the reader with an introduction to remote visualization through the use of OpenGL Vizserver, an overview of its structure and examples of how it has been employed to benefit users of High Performance Computing.

1 Introduction

OpenGL Vizserver [4] offers many benefits to users of High Performance Visualization (HPV) and is a technology solution that can be deployed to help tackle the type of "Grand Challenge" problems relevant to HPC users. Vizserver is a remote rendering software system that allows users to take advantage of the hardware rendering of a HPV system remotely at their desk with no alteration to their application.

Vizserver is a SGI product and was developed to allow users to take advantage of the graphics rendering hardware available on SGI Onyx systems, Infinite Reality pipes, without leaving their desks. The application must produce images via OpenGL. Vizserver diverts all OpenGL rendering from the local machine to the host where the graphics hardware resides. The images rendered on the host are compressed and transmitted to the local machine over a high-speed data pipe. It has a simple structure because it is designed to take advantage of the remote display capabilities of the X windows system – see figure 1. RGB datasets can run at speeds up to 20Hz.

The architecture of Vizserver differs from other systems because it does not do any rendering, it just controls where the rendering occurs and the transport mechanism needed to carry the results to the client. The integration of Vizserver with the X windows system means no modifications are needed for an application to work with Vizserver. Users can take immediate advantage of Vizserver with existing visualization applications. It is also possible to integrate Vizserver into an application at a lower level (using the supplied) API. This type of usage is not covered in this case study report.

There are two types of Vizserver sessions: collaborative sessions and single-user sessions. Single-user sessions involve one client user (usually remote) using a free pipe on the server host. Collaborative sessions allow two distant users, one of them being a local client on the server host running an X server, to display and interact with the same applications in real time.

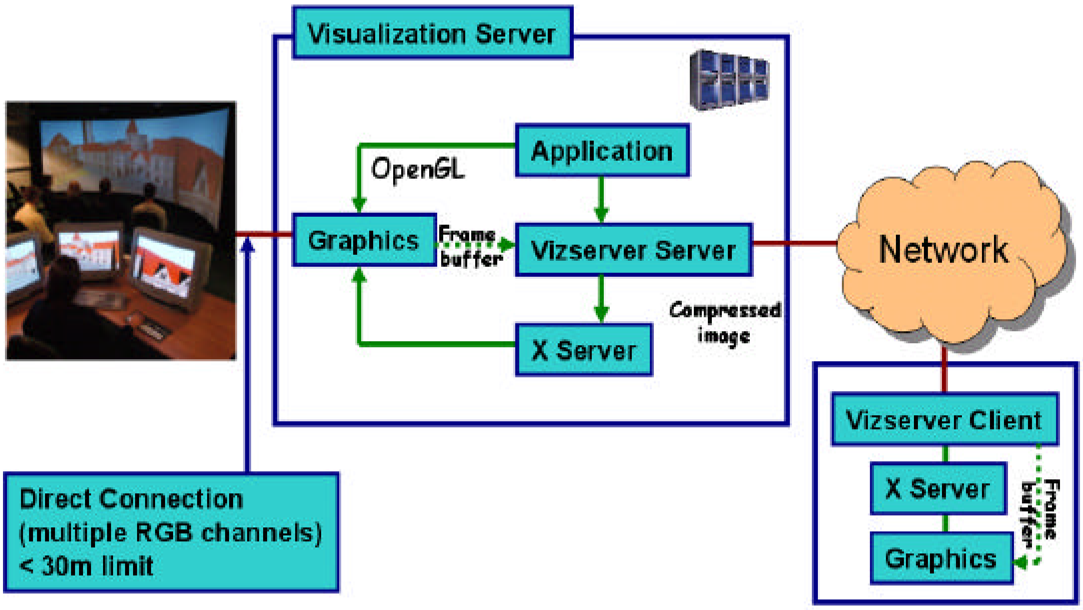

Figure 1: OpenGL Vizserver Overview

The client software is designed to run remotely on a number of platforms:

- IRIX 6.5.11+
- Solaris 2.6+
- Linux Red Hat 6.2+
- Windows NT 4.0, Windows 2000, Windows XP

1.1 Visualization Supercomputers and the Benefits of Vizserver

Visualization Supercomputers [3] are computers that have high-performance graphics, multiple processors, large memory, large amounts of disc space and quick access to disc. Although many visualization applications can be run today using desktop PCs there is a limit. The limit is normally caused by the size of the data. Large data fill all the available disc space, use all the memory and are slow to load into an application. Strategies can be developed to increase the limit like processing the data and storing it out of core instead of in memory but this comes at the expense of interactivity. It is the “Grand Challenge” problems that tend to have large data and so require Visualization Supercomputers.

Commercially, SGI dominate this area with their Onyx2 graphics servers[1]. These are large, expensive machines that need to be sited at specialist centres. They need a precise environment and the administrator needs specialist knowledge.

It is not sensible or even desirable to place a machine in the office of each user, yet it is desirable to have the capabilities of such machines in many offices. Traditionally the use of such machines has been limited because the user has had to book a room with a specialist graphics display and use the Visualization Supercomputer away from their desk where they keep any number of key documents/programs which many are useful to reference.

Vizserver allows the high quality visualization of large data remotely. The researcher can examine the results from their desk with a minimum of disruption.

[1] Fuel and Octane workstations can also be used as servers, but performance is significantly slower than a similarly configured Onyx system.

2 Basic Architecture

Vizserver has been designed to efficiently grab the contents of the frame buffer after rendering has been completed and transport it to the client. Frame buffer retrieval, compression, and network transport are structured into a pipeline architecture, therefore taking advantage of the shared memory and the multiprocessors available on the Onyx systems.

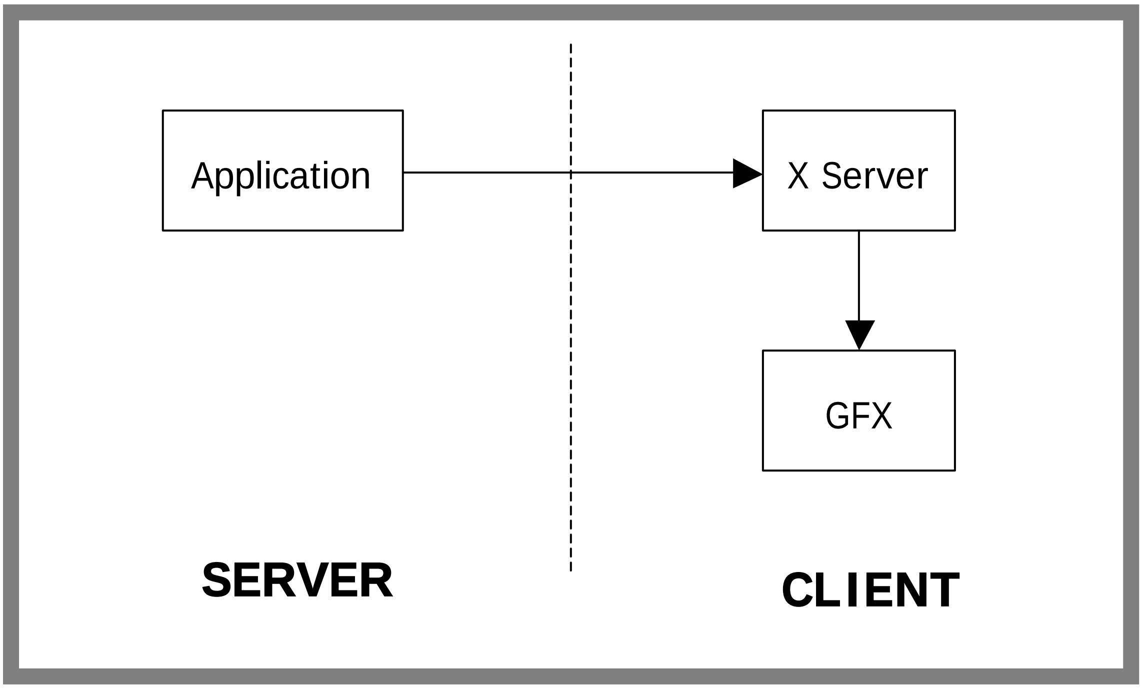

Figure 2: Under normal circumstances a remote application interfaces with the X server like this

The two main components of the Vizserver architecture are:

- Redirection of the OpenGL rendering to the host machine's graphics pipes.
- The transport of the rendered results to the client.

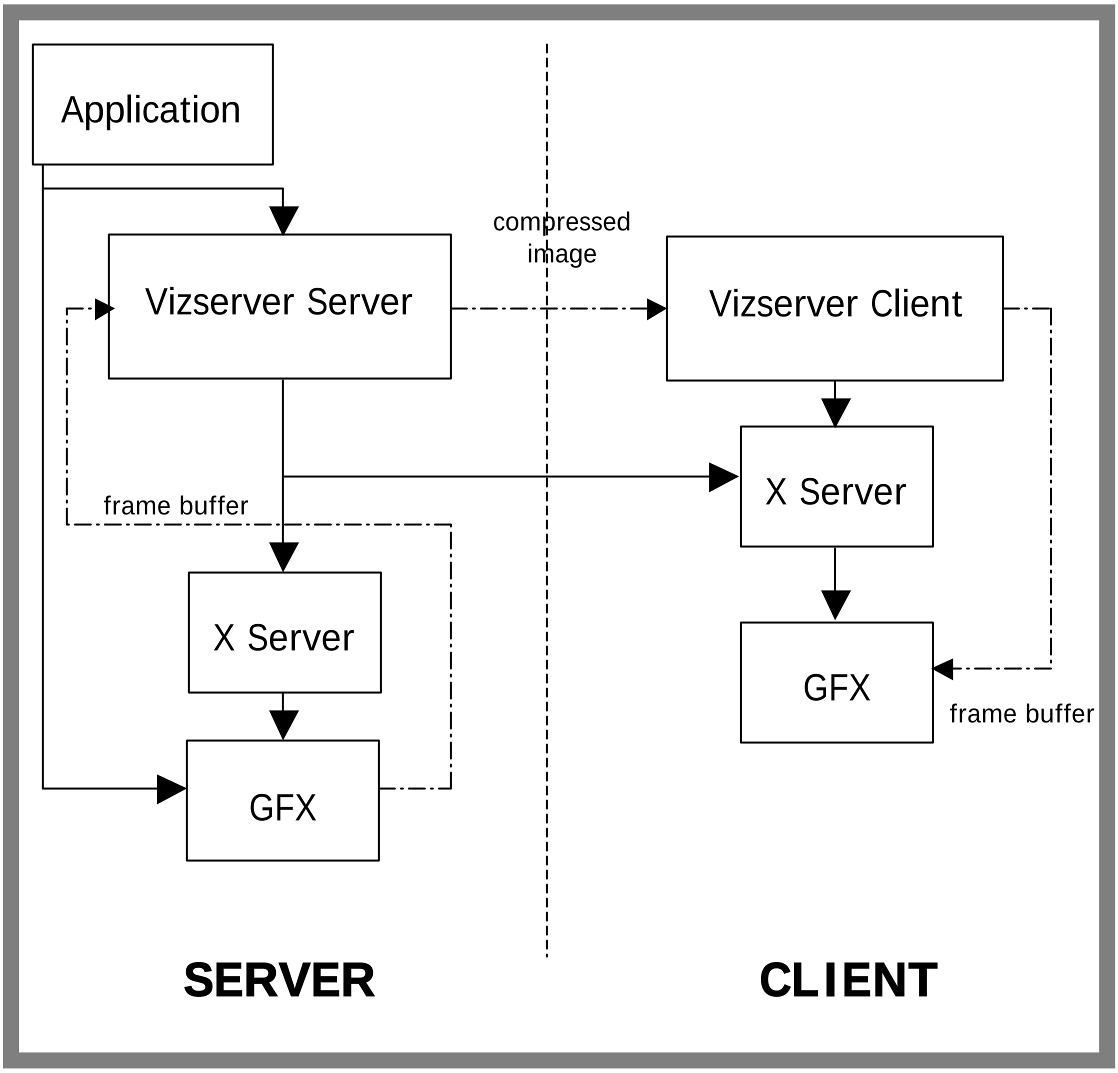

Figure 3: An application using Vizserver. Shows the two main components of Vizserver's architecture

The application interfaces with the X server through An OpenGL Vizserver layer, which redirects all OpenGL calls to the host machine’s graphics subsystem. When a frame is done, indicated by either glXSwapbuffers() or glFinish(), the final image is pulled out of the pipe through the OpenGL Vizserver layer, compressed, and sent to the client. The OpenGL Vizserver client then receives and decompresses the frame buffer images from the server and displays them in the correct window. The images are compressed using ColorCellCompression(CCC), which provides an 8:1 compression ration, and interpolated ColorCell(ICC) compression, which provides a 4:1 compression ratio.

It is important to note that each Vizserver session requires **exclusive access** to an InfiniteReality graphics pipe (two pipes are also an option). No other users can share this resource whilst the Vizserver session is being run.

The biggest bottleneck for Vizserver is the speed of the network connection from the client to the server. This must be a minimum of 100 Mbits/second to ensure good interaction, and preferably a dedicated connection. However, it is possible to run Vizserver over slower connection speeds if high image compression rates are used. Other performance factors that should be considered are the application window size, server and client robustness, application menu usage, compression rate, spoil frame ON/OFF, etc...

2.1 OpenGL Vizserver Roadmap

At the time of writing, OpenGL Vizserver v3.0 is shipping. Future functionality is being planned by SGI to be included in forthcoming releases. These include:

- Better collaboration – support for more than two simultaneous clients.
- Stereoscopy support. Currently applications running Vizserver cannot utilise stereoscopic projection (active or passive).
- Support for overlay planes.

Note that the Windows clients do not now need X-Windows software. Previous releases of Vizserver required Hummingbird's eXceed software to be installed. This restriction has been removed with v3.0.

2.2 Alternative Solutions

Other hardware companies have seen the potential of Vizserver for increasing utilisation of systems, and are developing similar solutions. There is currently no other commercial company that has a product similar to Vizserver. Recently Hewlett Packard [1] announced their intention to develop a large visualization clusters that would have some functionality similar to that of Vizserver. This work is being done with the American National Laboratories ASCI VIEWS project. There is another research based solution produced by Stanford University, California called Chromium [2]. This again is a cluster based system developed in collaboration with and used in the ASCI VIEWS project. Chromium works with OpenGL and allows the distribution of the data/rendering. Data processing and primitives production is done on the large platform where the data sits. The results are pushed to remote rendering systems, scaled to the data and display sizes. These rendering systems could be could be desktop systems or large rendering engines, depending on the size/type of display. Another specialised solution for real time volume rendering is the TeraRecon VolumePro card, which fits into the PCI slot of a desktop PC. They also have a version of their software that allows the PC with the VolumePro card to be accessed remotely.

3 Applications

Manchester Visualization Centre (MVC) is using Vizserver in several areas, and is actively involved in the beta test programme for new releases of the software (we are the only beta site in the UK). Two different projects provide examples of how Vizserver can be deployed. The first is a medical application where a surgeon needs the benefit of HPV in the operating theatre [6]. The second is a computational steering application that has been run across three sites [5].

3.1 Volume Rendering In an Operating Theatre

Currently, up to 40% of pancreatic and liver tumours prove to be inoperable at the time of first surgery. This is a statistic that has concerned Dr. Rory McCloy, a specialist liver and pancreatic surgeon at the Manchester Royal Infirmary for sometime. He has been working in collaboration with Dr. Nigel John of the Manchester Visualization Centre to try and improve the outcome of surgery.

The standard 2D information given to surgeons in the operating room can be difficult to interpret. The patient is scanned prior to the operation. The surgeon is then given a large sheet of transparent film with a series of 2D images on it of the site around the tumour. The surgeon must then mentally reconstruct the 3D location and shape of the tumour. The surgeon needs to know the location, extents and to which organs the tumour is attached for the operation to be successful.

Op3D is the visualization application that has been developed to allow the surgeon to interactively explore a 3D reconstruction of the tumour in the operating theatre. It was from the start designed to work with Vizserver so that visualization supercomputing power could be accessed remotely. The operating theatre is a sterile environment with a restricted amount of space. The patient is in a sensitive state and their safety is paramount so the surgeon must have software that is simple to use and not distracting.

A major objective of Op3D is to make it easy for a surgeon to use and manipulate the visualization during surgery. A joystick is used as the control device rather than a keyboard or mouse. Prior to surgery the joystick is covered with a sterile plastic bag, allowing the surgeon to switch between performing surgery and manipulating the visualization without risking contamination.

Op3D is designed to do things in a predetermined order according to the surgeon's preference. Surgeons can create their own profiles that record the steps they will follow during surgery such as slicing a 3D volume, rotating and zooming 3D imagines or creating 3D cut planes to reveal the structures hidden below the surface. That way they can easily move from step to step at the click of a button without having to make selections from pull down menus. A surgeon can also practice with the data before entering the operating room.

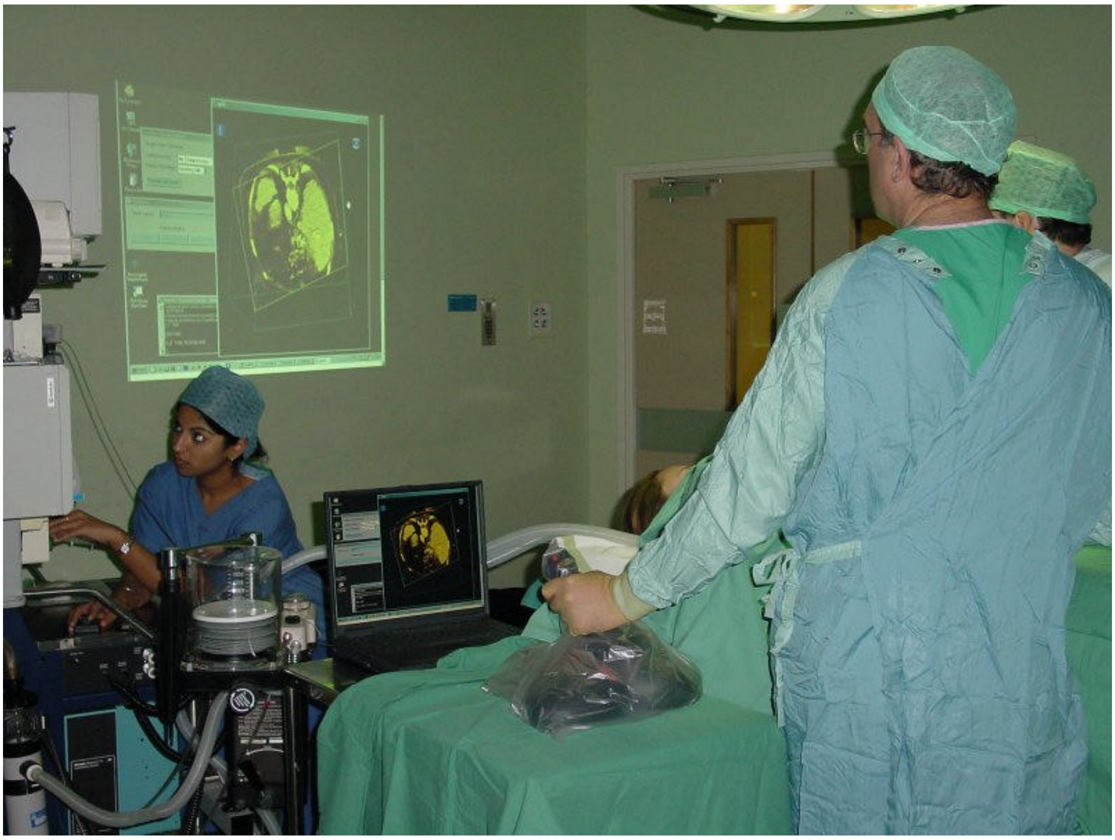

Figure 4: The Op3D System in use

Using Vizserver also gives the doctors the ability to work with Op3D in a collaborative mode. Two distant surgeons at different hospitals can through vizserver use Op3D through to discuss and compare opinions.

This application has been run in the clinical environment since April 2002, and the first two of these were filmed for the BBC and the Discovery channel. A demonstration of this application was also given for delegates to the Global Grid Forum given in Edinburgh in July 2002. This demo did not occur in an operating theatre but was held as a collaborative session between MVC and the University of Glasgow. Using SuperJanet, we were able to run the application at an acceptable rate over more than 200 miles!

Current work on this project is extending Op3D so that it can interface with Grid middleware. The Op3D-Grid project has been funded by the ESNW (E-Science North West), with SGI as the commercial partner.

3.1.1 Hardware Configuration

Op3D, the specialist visualization application, was run on a 32-processor (MIPS R14000) SGI Onyx 300 visualization system housed at the Manchester Visualization Centre (MVC), the University of Manchester. One of the six available InfiniteReality 3 pipes was made available for the session. The operating theatre was more than a kilometre away at Manchester Royal Infirmary. The only new equipment required in the operating room is a laptop computer (connected to the network) and a video projector.

The University of Manchester has a one Gigabyte network backbone. A 100baseT connection was installed from the backbone to the hospital operating room. This has allowed the application to be run at over 10 frames per second with no image compression.

The surgeon can interact with the visualization using a joystick which has been covered in a sterile plastic bag. The video projector enlarges the images and projects them onto the wall of the operating room for viewing.

3.1.2 Software

Op3D is a specialist visualization program developed at MVC. Op3D was developed using OpenGL Volumizer a volume rendering library from SGI which sits on top of OpenGL. Volume rendering is a technique especially suited to volumetric data of the type produced by medical scanners.

Vizserver was used to pass images created on the supercomputer at MVC to the operating room. Control information was passed back from the operating room to the super computer allowing the surgeon interactivity.

3.2 Isosurfacing and Volume Rendering of a Steered Code

RealityGrid is a large e-science project based at several sites across the UK. There are two sites one in London and one in Edinburgh that are interested in understanding the phase separation of a mixture of two fluids by computational simulation. They already have codes that allow them to do this but without computational steering and visualization of the results it takes them a long time to appreciate the parameter space of their model. RealityGrid has developed software and integrated it with existing software to make computational steering and visualization possible across the sites/country. Vizserver is one of the components used by RealityGrid.

A demonstration of the latest version of this application was given at the UK e-Science All-Hands Meeting in Sheffield at the beginning of September 2002. A full description has been given below.

3.2.1 Hardware Configuration

The simulation itself was run on eight processors of an SGI Onyx at Queen Mary, University of London. The visualization software also used eight processors on a second SGI Onyx at the University of Manchester and the steering software itself was run on a laptop at the North West Regional e-Science Centre's booth in Sheffield.

The connection between the laptop and the Onyx in Manchester had a bandwidth of approximately 10 MBps.

3.2.2 Software

The demonstration used a number of software components:

- Parallel 3D Lattice Boltzmann simulation code written in F90 (developed by Professor P Coveney's group at Queen Mary, University of London);
- Visualization software performing (parallel) volume rendering and isosurfacing using the vtk library (C++ wrapped by Tk/Tcl);
- Steering and application-launching software - Java GUI to the Unicore client produced by Fecit (Fujitsu European Centre for Information Technology).

The Unicore client allowed the user to select the type of application they wished to run, the machine on which to run it and the number of processors to use (where applicable).

Vizserver was used to compress and transfer the images (OpenGL frame buffers) produced by the visualization software running in Manchester to the laptop's screen in Sheffield. This enabled the user in Sheffield to interact with the visualization (rotate, zoom-in etc.) as though the visualization were running locally.

3.2.3 Middleware

The demonstration utilised Fecit's Unicore server implementation as a Hosting Environment which, combined with a standard Web Services platform (www.themindelectric.com) provided an almost-complete OGSA (Open Grid Services Architecture)-compliant framework.

Within this framework, a Grid Service (for any supported steerable application) is dynamically created by a Grid-Service Factory. The resulting job may be started on any remote (Unicore) system and observed/controlled from the client. Output files produced by the job can be routed to a remote site or returned to the client.

For the demonstration, the steering software allowed the user to request the simulation to output a data file. Such output files were automatically routed to the second remote site where the visualization grid service was executing.

This framework demonstrated fault tolerance too. A serial isosurface visualization was closed and then a parallel volume rendering visualization was started, all without disrupting the simulation.

3.2.4 The Physics and the Benefit of Steering

The demonstration centred on the simulation of the phase separation of a mixture of two fluids. Using the steering software, the user was able to control the magnitude of the repulsive force between the two fluids so that they changed from totally immiscible layers of fluid to totally a miscible fluid. The GUI of the steering application was run on the laptop and provided read/write access to this parameter as well as reporting the values of other simulation variables.

The simulation followed the system as it evolved from a starting point of a binary emulsion. The visualization output enabled the user to follow the effects of their steering and thus identify parameter values of physical interest.

References

[1] Hewlett-Packard Company, *U.S. Department of Energy Selects HP for ASCI PathForward Scalable Rendering Program – Technology Targeted at Building Large Visualization Clusters,* October 28th 2002, http:// www.hp.com/hpinfo/newsroom/press/2002/021028a.html

[2] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahern, P. Kirchner, J. Klosowski, *Chromium: A Stream-Processing Framework for Interactive Rendering on Clusters,* SIGGRAPH 2002

[3] N. W. John, J. Leng, *Scientific Applications of Visualization, Virtual Reality and High Performance Visualization Computers*, technology watch report for UKHEC March 2001

[4] C. Ohazama, *White Paper OpenGL Vizserver,* SGI 1999

[5] The RealityGrid project, *RealityGrid: a Tool for Investigating Condensed Matter and Materials presentation,* UK e-science All Hands Conference, 2nd-4th September 2002

[6] L. Verswayveld, *SGI technologies and grid visualisation advanced membrane protein research at Henri Poincare University*, VMW (Virtual Medical Worlds), December 2002, http://www.hoise.com/vmw/02/articles/vmw/LV-VM-12-02.html