Interchanging Interactive 3D Graphics for Astronomy[1]

C. J. Fluke[A,B], D. G. Barnes[A], and N. T. Jones[A]

[A] Centre for Astrophysics and Supercomputing, Swinburne University of Technology,
  PO Box 218, Hawthorn VIC 3122, Australia
[B] Corresponding author. Email: cfluke@swin.edu.au

Abstract: We demonstrate how interactive, three-dimensional (3D) scientific visualizations can be efficiently interchanged between a variety of mediums. Through the use of an appropriate interchange format, and a unified interaction interface, we minimize the effort to produce visualizations appropriate for undertaking knowledge discovery at the astronomer's desktop, as part of conference presentations, in digital publications or as Web content. We use examples from cosmological visualization to address some of the issues of interchange and to describe our approach to adapting s2PLOT desktop visualizations to the Web[2].

1 Introduction

There are four main ways that astronomers (and scientists in other disciplines) use visual representations of datasets:

1. for knowledge discovery;
2. for formal publication of research results;
3. for academic presentations; and
4. for (formal) education and (informal) public outreach.

One of the challenges of developing effective visualizations appropriate for all of these targets is the increasingly multi-dimensional nature of astronomy data. Consider the wealth of information offered by four-dimensional radio data cubes of neutral hydrogen clouds (RA, Dec., velocity and intensity); seven-dimensional cosmological simulation outputs (three-dimensional positions and velocities, particle mass); or the 77-dimensional Hipparcos Main Catalogue (Perryman et al. 1997).

A wide range of software packages, including custom codes and commercial solutions, are available for the steps of processing, analyzing and visualizing multi-dimensional astronomy data. However, despite the wealth of tools and techniques, most of astronomy is still performed, presented, taught and learnt in only two dimensions. Aside from the obvious explanation that producing and displaying two-dimensional (2D) images from scientific datasets is relatively straightforward (see item 14 in Globus & Raible 1994), another factor is relevant: 2D plots and figures produced for one medium (e.g. paper) are easily transferable to another medium (e.g. screen). That is, they are easy to *interchange*. A Cartesian scatter plot can be sketched on a whiteboard, drawn accurately on-screen for analysis, be saved in suitable format(s) for presentation in a lecture or conference talk, and for publication in a printed journal. The audience 'uses' the 2D plot in the same way, regardless of the medium via which it is delivered.

If we extend the visual dimensionality from two to three, no such simple interchange mechanism exists. The visualization tool used by the researcher on the desktop is typically not the same one that is used to make a conference presentation, and the creation of educational versions of datasets is usually achieved by handing the data to an animator to interpret using a commercial animation package. At each stage, data is converted for use from one tool to another, each step taking it further away from the original desktop experience, and requiring significant additional effort for the astronomer. Furthermore, there is no common user interface; a fully interactive desktop application is most often reduced to a series of still images. What hope, then, for more complex, multi-dimensional datasets?

We have previously reported on the s2PLOT interactive graphics library for creating three-dimensional (3D) visualizations for knowledge discovery (Barnes et al. 2006), and demonstrated the novel capability to produce interactive 3D figures in the Adobe Portable Document Format (PDF) to enhance academic publications (Barnes & Fluke 2008) and education (Fluke & Barnes 2008). We now describe solutions for interchanging 3D figures amongst the remaining output targets, as shown in Figure 1. Specifically:

- Desktop visualization is accomplished with an application (C, C++, FORTRAN, PYTHON; MYS2PROGRAM in Figure 1) that uses the s2PLOT library. With a programming interface similar to the popular PGPLOT 2D
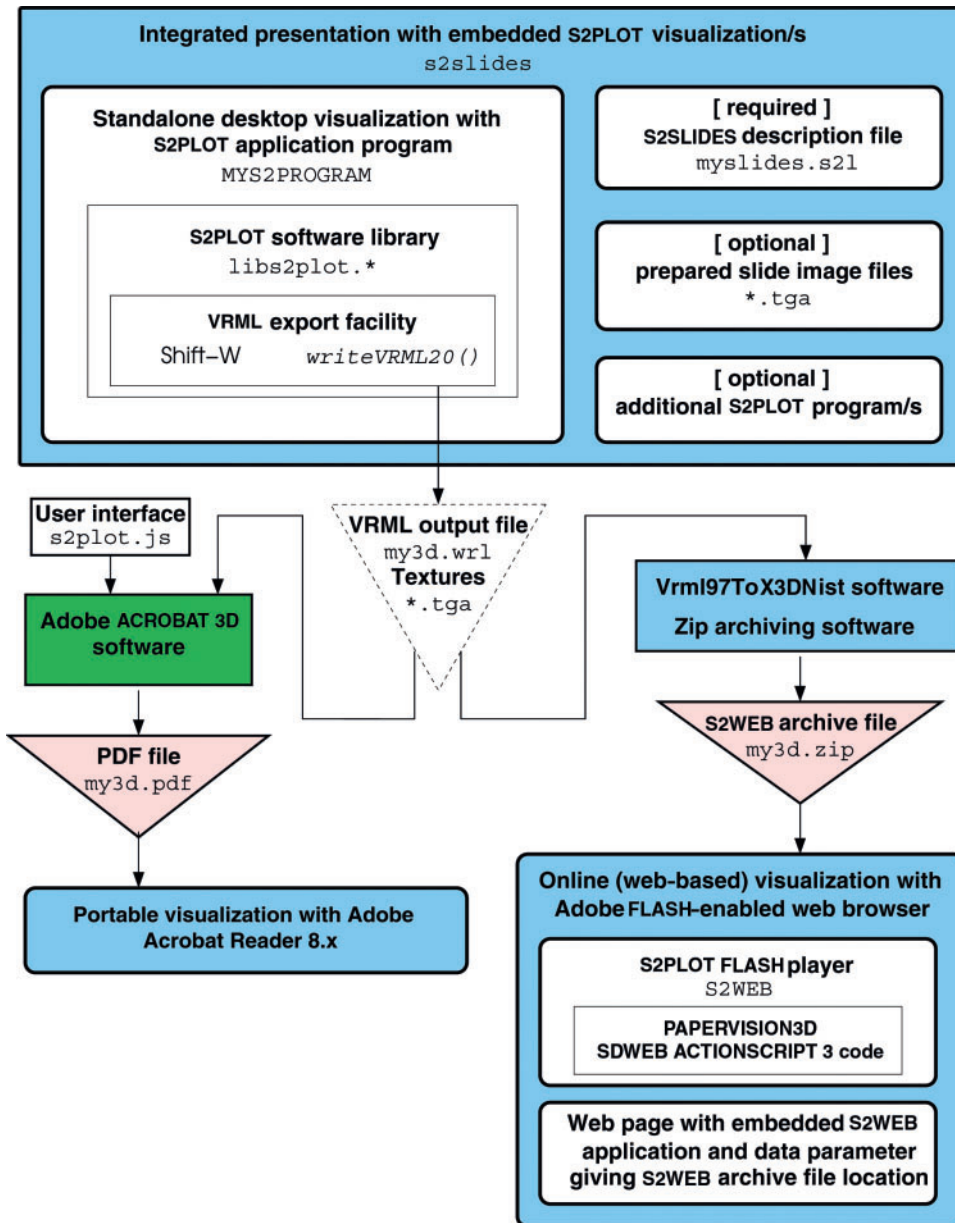
Figure 1 An integrated system for creating, interchanging and using interactive 3D figures in astronomy, based on the S2PLOT library. The large upper box encapsulates desktop visualization and S2SLIDES presentation modes, while the lower part of the figure shows the components concerned with production of 3D PDF and web-based visualizations. Shaded triangles indicate primary output products, and the dashed triangle indicates the (transient) interchange step: the VRML output file and textures do not need to be retained once the primary output product/s are created. Adobe ACROBAT 3D is a commercial product.

library,[3] S2PLOT provides a straightforward way for astronomers to produce 3D plots on-screen through simple function calls within their own custom codes.

- 3D figures are embedded in PDF files by exporting the S2PLOT program geometry to VRML (Virtual Reality Modeling Language; keypress Shift-W or function call writeVRML20() in standard S2PLOT programs), then importing the VRML to Adobe ACROBAT 9 PRO EXTENDED, Adobe ACROBAT 3D or the Adobe 3D TOOLKIT (commercial software). The standard S2PLOT interaction mode (as well as support for

volume rendering, billboards and so on) is implemented in the provided s2plot.js script, which can be added to the 3D PDF model. The exported VRML file is an interchange file only; it need not be kept after the PDF file is created.

- POWERPOINT-style presentation is enabled by the S2SLIDES application included with S2PLOT: it provides for navigation through a sequence of pre-rendered slides (*.tga file/s in Figure 1; e.g. bitmap images exported from Microsoft POWERPOINT), and can overlay standalone interactive S2PLOT programs on slides according to simple instructions in an input text file (myslides.s2l file in Figure 1). Note that S2SLIDES does *not* require

---

[3] http://www.astro.caltec.edu/tjp/pgplot.

Microsoft POWERPOINT or any other commercial presentation software, although such software may be a convenient way to generate non-interactive slides for presentation with S2SLIDES.

- 3D figures are deployed on the Web by exporting to VRML, as for the PDF case, converting the VRML to X3D (a simple, syntax-only conversion accomplished with Vrml97ToX3DNist). This X3D is then zipped up with the required textures. The resultant archive (an 's2WEB archive file') can then be provided to the S2WEB program, which is an Adobe FLASH (hereafter 'FLASH') program that renders S2PLOT-originated X3D content. S2WEB allows the user to control the camera and rendering properties just as they do for desktop S2PLOT programs.

A key feature of our approach to interchange is the preservation of the user experience, particularly via the keyboard and mouse controls for interaction, data selection and control of the visual appearance (such as rendering mode).

This paper is organized as follows. In Section 2, we discuss our choice of Web technologies for integration with S2PLOT. We detail the practical application of these technologies that enable S2PLOT programs to produce 3D visualizations for multimedia deployment. We further explore the motivation for, and implementation of, a common user interface across the output targets for S2PLOT programs. In Section 3, we consider the specific case of cosmological data as motivation for the demands and requirements of interchanging 3D data and models, and present several applications enabled by our approach in Section 4. A brief discussion of our future plans follows in Section 5, and we make some concluding remarks in Section 6.

2 Interchange

Astronomers have been willing users of Web technologies, most significantly in the way that they communicate and collaborate across the world. Astronomers have also taken to fully online, digital publication of research papers and data archives. Of particular importance were the first online edition of the Astrophysical Journal published in September 1995 (Abt 2003), the abstract searching capabilities of the ADS Abstract Service (Murray et al. 1992; Kurtz et al. 2000), the widespread availablity of scanned articles (Eichhorn et al. 1994), and the establishment of the Centre de Donnees astronomiques de Strasbourg (CDS) online data archives (Genova et al. 1996), the precursor to the Virtual Observatory (Quinn et al. 2004). Amongst the physical sciences, astronomers are second only to the particle physics community as users and drivers of Internet technologies.

The present problem we are addressing is the provision of a pathway for publishing and sharing interactive 3D datasets on the Web. Accordingly, our approach is to develop and implement the necessary (informal) standards that yield a practical mechanism for the creation, interchange and use of interactive 3D figures.

2.1 Exporting S2PLOT Geometry

Simple and robust standardized 3D interchange formats have been available for quite some time, notably ISO/IEC 14772:1997 VIRTUAL REALITY MODELING LANGUAGE (VRML) and its successor ISO/IEC 19775:2004 EXTENSIBLE 3D (X3D). When implementing our 3D PDF solution (Barnes & Fluke 2008), it was necessary to choose an interchange format for exporting 3D geometry from S2PLOT programs. We selected VRML because it is one of the few open, non-proprietary file formats that Adobe ACROBAT 3D can import. Programs using versions 2.0 and upwards of the S2PLOT library can now write the S2PLOT geometry data to a VRML file that can be imported into ACROBAT 3D for the creation of 3D annotations in PDF documents.

It is worth making the distinction between our use of VRML and earlier uses of VRML for astronomy visualization (Crutcher, Plante & Rajlich 1998; Plante et al. 1999; Beeson et al. 2004). In previous systems, VRML was the final product: it had to be viewed in a VRML browser such as Cortona or FreeWRL. Historically VRML browsers have not been particularly complete (in terms of implementation of the entire language), fully cross-platform or especially stable. Nor have their user interfaces been adaptable to suit users' expectations. Our use of VRML is as an *interchange* format: it is simply an intermediate format for expressing S2PLOT geometry for onward conversion and/or import into other software.

The scene geometry in VRML is structured as a model tree, a concept common to many 3D applications and scene description formats. The tree allows related components of the geometry to be grouped in a 'branch'. This tree is preserved upon VRML import to ACROBAT 3D, and Adobe READER provides both an application programming interface (API, via Adobe 3D JAVASCRIPT) and a simple user interface for switching parts of the model tree on and off. For our 3D PDF solution, we added a function to S2PLOT that allows the user to arrange the geometry into a tree when exporting VRML. Selective use of this function (pushVRMLname) makes it possible, for example, to export two or more different visualizations of the same data from one S2PLOT program into one VRML file (we demonstrate an example of this in Section 4.3). Importing this VRML file to ACROBAT 3D together with some very basic JAVASCRIPT code enables *reader* control of the visualization state via buttons or other user interface elements in the target PDF file.

2.2 Web Deployment with FLASH

FLASH applications are ubiquitous on the Web, which, together with the breadth and depth of the ACTIONSCRIPT 3 API, makes FLASH the ideal system for the development of a Web-based 'player' of S2PLOT content. Unlike Java — the only other viable environment for Web-based 3D at this point in time — FLASH applications run in-browser, integrate seamlessly with static Web content, and the necessary plugin(s) are generally kept up-to-date.

Ideally for the academic community, FLASH applications can be developed with the freely-available Adobe FLEX 3 SDK.

Writing a complete 3D engine for FLASH would be a formidable task. Happily there already exists a maturing, open-source project implementing a 3D environment in FLASH: PAPERVISION3D[4]. PAPERVISION3D is a software 3D API for FLASH, distributed under the MIT license. It provides a high-level 3D engine capable of displaying a standard set of 3D primitives, mapping a variety of textures onto objects, including image and movie textures, and navigating and manipulating 3D space. PAPERVISION3D objects can be defined in ACTIONSCRIPT 3 code or can be read from external files that contain the geometry information.

PAPERVISION3D allows camera and environment redefinition such that we were able to port the code controlling the S2PLOT interface to ACTIONSCRIPT 3, giving a very closely matched interface to the one presented by native S2PLOT desktop applications. PAPERVISION3D even allows for object interaction, so that the S2PLOT *interactive* 'handles' can be made available within FLASH applications.

Our Web-based solution for 3D export from S2PLOT entails a simple, syntactic translation of the exported VRML to X3D. This translation retains the model tree structure, and consequently our FLASH S2WEB program can also provide explicit or implicit user-control of the model tree and visualization as has been accomplished for PDF.

Predefined strings within the model tree are used to denote S2PLOT 'handles' (which are selectable elements of scene geometry) and 'billboards' (textured rectangles which always face the camera), and to support volume rendering. Using predefined (or 'special') strings for pushVRMLname makes downstream ACTIONSCRIPT 3 or JAVASCRIPT code *universal*, e.g. one Adobe 3D JAVASCRIPT code for rotating billboards correctly will work for all S2PLOT VRML output containing billboards.

X3D was chosen as the most appropriate interchange format for the following reasons:

- For the set of features requiring support, X3D is equivalent to VRML 2.0, easily convertible from VRML, and S2PLOT was already capable of outputting VRML.
- XML-encoded X3D is very simple to read and parse in FLASH; ACTIONSCRIPT 3 has XML parsing as a native part of the language, and the XML Document Object Model (DOM) structure is mirrored by the ACTIONSCRIPT 3 objects created. This is particularly useful for parsing the hierarchical geometry model tree, and maps isomorphically to the PAPERVISION3D geometry tree.
- Although text-based X3D documents are larger than the equivalent geometry expressed in a binary format, FLASH can easily read zipped files. A zipped archive can also contain the necessary textures for a scene, so an entire 3D scene can be exchanged in a single file

(this is much like Google Earth's KMZ files, which are zipped archives of Collada files with textures).

Our final Web-deployment platform is therefore a FLASH PAPERVISION3D application, capable of reading XML-encoded X3D converted output from S2PLOT programs, and providing a very similar user interface to S2PLOT. This enables two types of applications: a FLASH X3D static geometry-viewer (like the native S2PLOT program S2VIEW), and custom S2PLOT FLASH applications. The viewer has the advantage that it can display any supported static geometry output by S2PLOT, without having to write any ACTIONSCRIPT 3 code (an unfamiliar language to most astronomers). Nevertheless, the S2PLOT layer we have built on top of PAPERVISION3D is simple enough that scenes *can* be animated just by editing a short callback, in the same manner in which geometry is made dynamic in the S2PLOT library.

3 Cosmological Data

In order to demonstrate the practicalities of interchanging multi-dimensional data, we consider the case of cosmological visualization: representation of the three-dimensional spatial structure of the Universe, including both observational and simulation datasets. Where such information exists, we extend this definition to include time-evolving datasets (e.g. evolution of structure formation or the hierarchical merging of galaxies), and derived data products such as catalogues and merger trees. A modern, fully digital cosmological visualization allows the user to rotate, zoom, pan and even interactively select from datasets.

3.1 Sources of Data

Observational cosmological datasets are usually obtained from galaxy redshift surveys whereby 2D sky positions (RA, Dec.) are combined with redshift (z) to give a 3D spatial location. The most common representation is the cone or wedge diagram, as shown in Figure 2 using galaxy positions from the 6dF Galaxy Redshift Survey second data release (Jones et al. 2005). An interactive 3D visualization is advantagous in this case, so that the wedge can be rotated and zoomed. Without interactivity, it is difficult to interpret the details of the large-scale structure from a single viewpoint. This conveniently highlights the limitations of a static, 2D publishing and presentation medium for n-dimensional scientific data!

Cosmological simulations generally use N-body techniques to efficiently calculate the gravitational forces between an ensemble of dark matter, gas and star 'particles'. Each particle often comes with full information on mass, spatial position (x, y, z) and velocity (v_x, v_y, v_z) over a range of time steps (or snapshots). Derived properties include energy (kinetic and potential), temperature and density, which may be calculated on a per particle basis or on a grid. Dubinski (2008) provides examples of standard technqiues for representing N-body data, but with an emphasis on pre-rendered sequences rather than real-time interactivity.

---

[4] http://blog.papervision3d.org/.
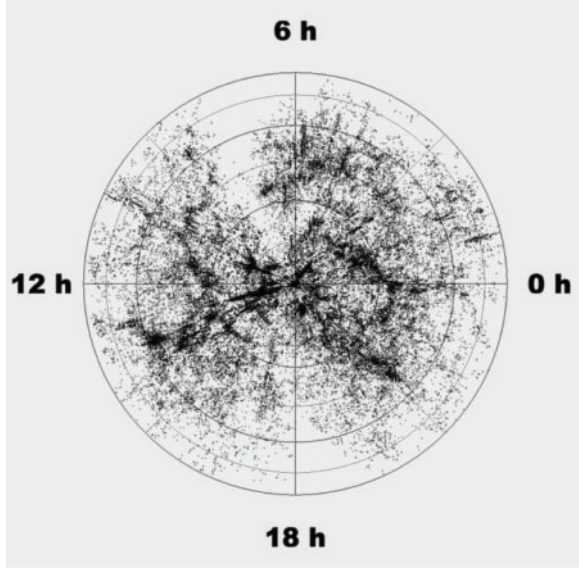
Figure 2 A cone or wedge diagram is the standard representation for galaxy positions from a redshift survey, usually projected along the line of the celestial poles. The data is from the 6dF Galaxy Survey (Jones et al. 2005). 49 377 galaxy positions, with $0 \leq v \leq 15\,000\,\mathrm{km\,s^{-1}}$ and Dec. $\leq 0°$, are shown as black points in this projected view. Declination lines are drawn at intervals of 15°.

3.2 Visualization Requirements

There are four main data types that a cosmological visualization may need to represent:

1. *3D points*: (RA, Dec., z) triplets for redshift surveys and (x, y, z) positions for N-body simulations.
2. *3D vectors*: (v_x, v_y, v_z) velocity data from cosmological simulations, which have a spatial location and a direction, or flow lines representing the motion of particles over time;
3. *2D and 3D mesh-based data*: density, temperature, etc. suitable for display as pixel-based images (2D) or with volume rendering (3D). Volume rendering (Drebin, Carpenter & Hanrahan 1988) combines gridded data with lighting effects to give a virtual solid. Volume rendering requires an opacity to be defined for each volume element (voxel). In general, this is a combination of absorption and emission, however, for cosmological datasets we usually only deal with emission: N-body particles or galaxies are either present or they are not.
4. *non-mesh data*: a practical example is the isodensity surface, the 3D analogue of a contour line in 2D. These may make use of gridded data as a calculation aid, but are not constrained to align with the mesh.

The treatment of the first three-dimensions is usally achieved through a direct mapping of (x, y, z) positions to screen coordinates for viewing on a 2D screen or with a 3D stereoscopic display. Higher data dimensions are visualized through a combination of techniques including:

- *colour*: whether plotting points or mesh data, an appropriate colour map can help to highlight significant features in a dataset. Used carelessly, though, colour is also the best way to ruin the effectiveness of a visualization (e.g. Tufte 1990, 2001);
- *point sizes*: are most suitable when there is a natural size-ordering (e.g. mass, magnitude), rather than for a morphological or descriptive quantity (e.g. galaxy type, survey date). The latter are often better distinguished through the use of symbols or *glyphs*;
- *splats*: are texture billboards, often with a gaussian profile, which can be used as an alternative to volume rendering, particularly when there are a large number of particles (Westover 1989, 1990). Splats can demonstrate physical scales by choosing an appropriate texture size. For an example from cosmology, see Hopf & Ertl (2003); and
- *animation*: usually reserved for visualizing variation with time, although it can also show the variation of a quantity, such as cycling through a sequence of density isosurfaces.

S2PLOT provides support for each of the data types and techniques described above, and so is appropriate for generating a range of cosmological visualizations. To provide the requirements of interchange, a move to other output formats requires that all of these data types are supported. Using examples from cosmological visualization, we now demonstrate how this has been achieved.

4 Cosmological Examples

In this section, we present several specific examples from cosmological visualization. We look at how our approach permits interchange between S2PLOT desktop applications, S2SLIDES presentations and S2WEB. For 3D PDF demonstrations of interactive volume rendering of a dark matter halo and a wedge diagram based on the CfA2 redshift survey, we refer the reader to Figure 5 of Barnes & Fluke (2008) and Figure 6 of Fluke & Barnes (2008) respectively.

4.1 Wedge Diagram

As discussed in the previous section, the wedge diagram is the most common representation of galaxy locations from redshift surveys. Therefore, it serves as an important first demonstration of our approach to interchange.

A static view of the 6dF Galaxy Survey ($0 \leq v \leq 15\,000\,\mathrm{km\,s^{-1}}$, Dec. ≤ 0), projected along the celestial poles, was shown in Figure 2. An oblique view of the same dataset is shown in Figure 3, obtained as a screenshot from an interactive S2PLOT desktop application. The figure shows density isosurfaces overlaid on the galaxy locations to help highlight the large-scale filamentary structures. The isosurfaces were generated by smoothing galaxy locations onto an 80^3-cell cubic mesh using the triangular-shaped cloud algorithm (TSC; Hockney & Eastwood 1988). The number count, n_{ijk}, in each cell is rescaled as $\log_{10}(1 + n_{ijk})$, and then linearly normalized to lie in the range 0–100. The final 3D mesh is converted to an isosurface using the S2PLOT ns2cis() function, and displayed with the ns2dis() function. Both 25% (red) and 75% (yellow) contour levels are shown. No attempt has
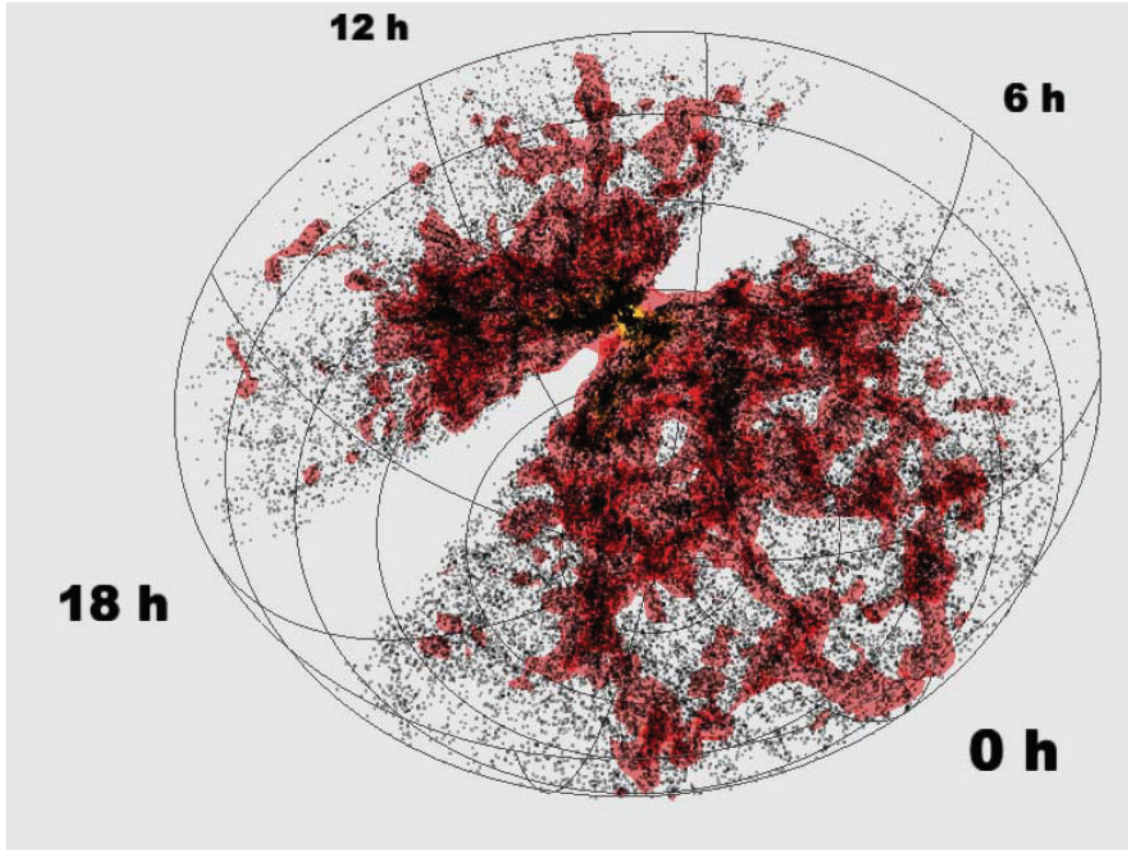
Figure 3 Oblique view of the 6dF Galaxy Redshift Survey (49 377 galaxies with $0 \leq v \leq 15\,000\,\mathrm{km\,s^{-1}}$ and Dec. $\leq 0°$). Density isosurfaces are overlaid on the galaxy locations, to help highlight the filamentary large-scale structure. Declination lines are shown at intervals of $15°$.

been made to correct for the non-uniform volume filling effect of the spherical (RA, Dec., z) coordinate system when smoothed onto a Cartesian grid in (x, y, z)-space.

The VRML output is saved into a file from the S2PLOT program when the user presses the key combination Shift-W. The conversion to X3D format is syntactic only, and we use the freely-available 'Vrml97toX3DNist' convertor provided by the National Institute of Standards and Technology[5]. Textures are written by S2PLOT in TGA format and must be converted to PNG format: we use the ImageMagick CONVERT program.[6] The resultant X3D file and PNG texture file(s) are then zipped together into one file. This file can be placed on the Web and an existing S2WEB FLASH program pointed to the zip file, or the zip file can be embedded in the FLASH program itself. For this paper, we do the latter.

The S2WEB version of the 6dF Galaxy Survey can be viewed online[7]. Mouse control of camera orientation, and key presses for functionality such as zooming, toggling autospin, and changing the spin speed are consistent between the Web and desktop applications. This is a critical part of our interchange philosophy — that there is a common user interaction experience. Of additional

interest here is the use of textured billboards for the RA labels, which retain their orientation towards the viewer regardless of the camera orientation. Line thickness, such as for the grid, is only a global setting in VRML, and correspondingly X3D. Therefore S2PLOT programs with S2WEB (or PDF) as a target should not use thick lines. A work-around is to use thin cylinders instead of lines.

4.2 Merger Tree

Semi-analytic models of galaxy formation (White & Frenk 1991; Cole et al. 2000; Croton et al. 2006) make use of halo merger trees, usually generated from dark matter cosmological simulations. Starting with a catalogue of halos at $z = 0$, the halo constituents are traced backwards in time within the simulation, so that the history of mergers can be reconstructed. Each progenitor halo can then be tagged with a galaxy, and used as an input to a (prescriptive) galaxy formation model. In this example, we use data from the Millennium Simulation (Springel et al. 2005) and the galaxy model of De Lucia & Blaizot (2007).
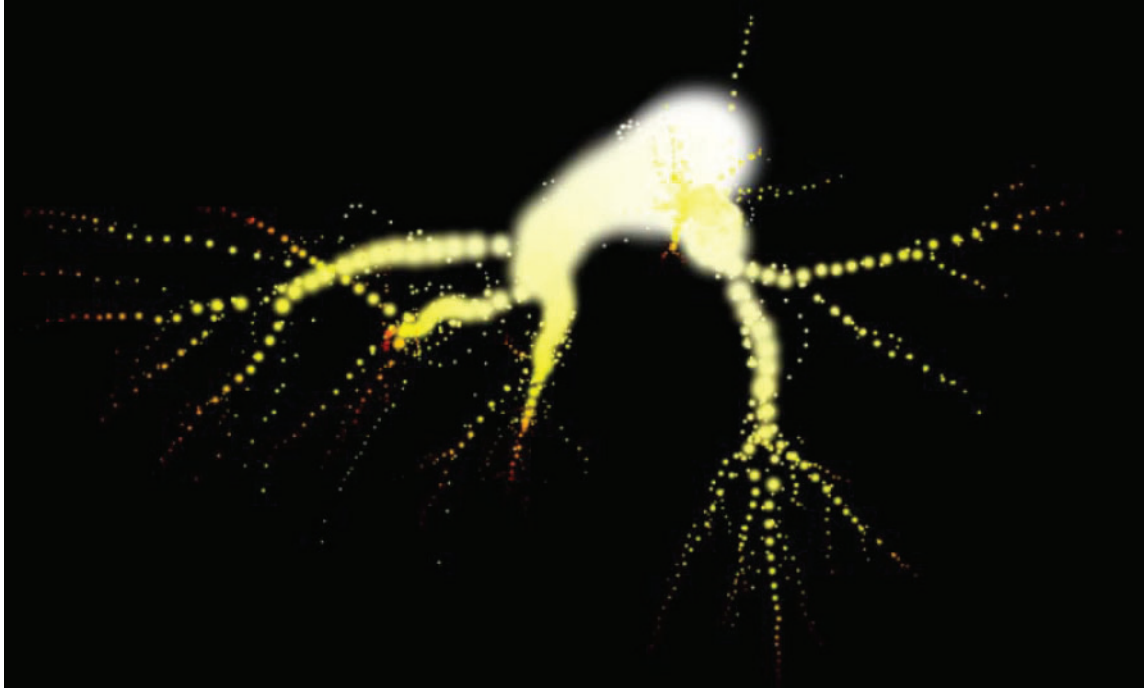
The S2PLOT application reads a text file generated by an SQL query of the Virgo Millennium Database,[8] returning the list of progenitors for a given halo. The application is able to plot any combination of three (numerical) parameters in the 3D space, and then applies a colour map based

---

[5] http://ovrt.nist.gov/v2_X3D.html.
[6] http://www.imagemagick.org/.
[7] http://astronomy.swin.edu.au/s2plot/interchange/6dF.

[8] http://www.g-vo.org/Millennium.

Figure 4 Dark matter halo merger tree from the Millennium Simulation. Progenitor halo positions are mapped to (x, y, z) coordinates in 3D space, colour represents the redshift at which each progenitor is bound to the final halo ('heat': red = high redshift, white = low redshift), texture splat size is scaled by virial radius of each sub-halo.

on the fourth parameter. In Figure 4, we show (x, y, z) spatial coordinates for the progenitor halos, and colour by the redshift at which each progenitor is bound to the final halo ('heat' colour map: red = high redshift, white = low redshift). Rather than using points to mark each progenitor, we use textured splats, with sizes scaled by the virial radius of each sub-halo.

Export to VRML and conversion to Web format is the same as for the 6dF case, and the S2WEB version is available online[9]. In this example, some additional steps are required to generate the appropriate texture splats. Within S2PLOT, a single texture image is used, and the colour is controlled within the software. A current limitation in PAPERVISION3D's material attributes for rendering is that only the first colour applied to the texture will be used. Accordingly, we post-process the VRML file prior to conversion to X3D, and where a single texture is used more than once in different colours, we create multiple copies of the texture so that PAPERVISION3D and S2WEB code correctly shades the textures. Typically, coloured billboards are very small occupying a few tens of pixels, and so the penalty in total zip file size for this approach is small. We intend to build this capability directly into the S2WEB code shortly.

4.3 Galaxy Merger

Investigating the dynamics of mergers between galaxies through numerical simulations has become critical to

understanding how galaxies evolve over $\mathcal{O}(\text{Gyr})$. Interactive viewing of the galaxy merger shows the different ways that such a system might be seen on the sky. By varying view angle and merger parameters, such as relative masses of the galaxies or distance of closest approach, the goal is to reproduce known merging systems (such as the Antenna galaxies NGC 4038/9, 90 million light years distant in the constellation Corvus, or the more distant merging pair NGC 2207 and IC 2163 in Canis Major) and hence gain insight into their dynamical evolution.

We use here a sample case distributed with the GADGET-2 gravity-plus-smooth-particle-hydrodynamics code (Springel et al. 2001; Springel 2005). This test case simulates a merger between two equal-mass spiral galaxies, using 30 000 particles per galaxy. An S2PLOT application has been developed to enable interactive visualization of the merger, including using isosurface (Figure 5; top row) and volume rendering (Figure 5; bottom row) techniques to help highlight and identify specific features. The application can read multiple GADGET-2 datafiles, cycling between the series of steps in the merger; three representative stages in the merger are shown. As with the 6dF wedge diagram, isosurfaces are built by smoothing particle positions (in this case, onto a 100^3-cell cubic mesh), and the number counts are log-normalized. Density isosurface levels of 20% (red) and 65% (yellow) are shown. A lower resolution mesh (80^3) was used for the volume rendering, and a 'rainbow' colour map (blue-green-yellow-red from low to high density) is applied over the data range.

Parameters for the visualization were adjusted within the S2PLOT program, and the isosurface model was exported to VRML and converted to X3D. Since

---

[9] http://astronomy.swin.edu.au/s2plot/
interchange/tree.
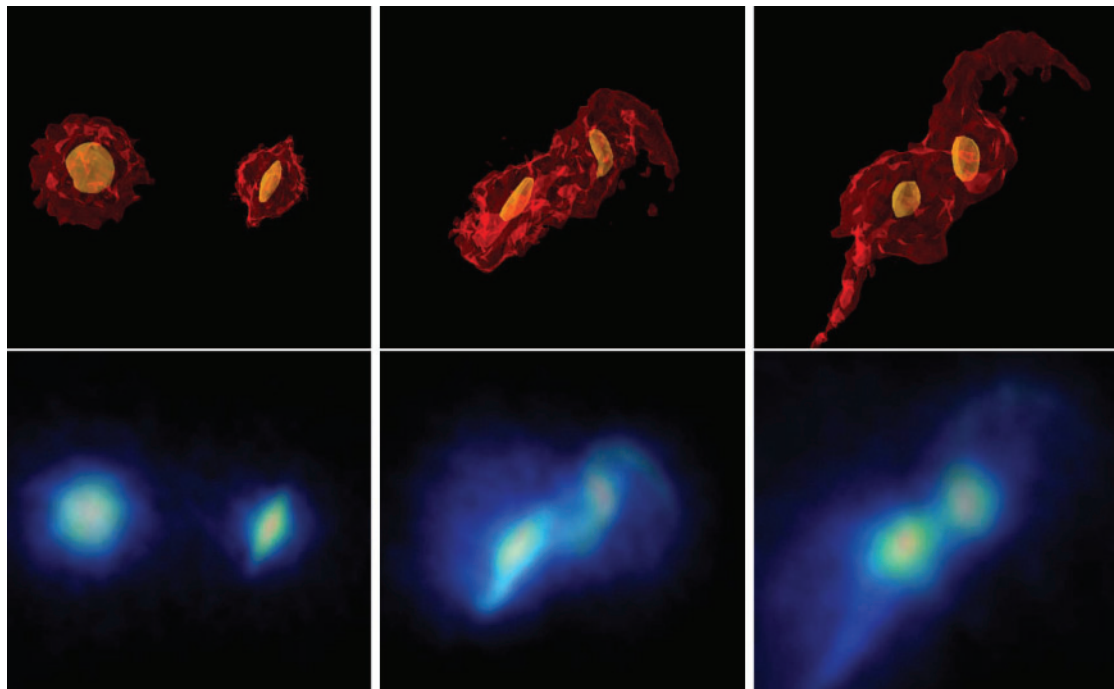
Figure 5 Three steps in a merger between two equal-sized galaxies. The GADGET-2 simulation comprises 60 000 particles that are smoothed onto a 100^3-cell cubic grid using the TSC algorithm. The counts in each cell, n_{ijk}, are scaled by $\log_{10}(1 + n_{ijk})$ and then normalized to lie in the range 0–100. (top row) Density isosurface levels of 20% (red) and 65% (yellow). (bottom row) Volume rendering with a linear 'rainbow' (blue-green-yellow-red from low to high density) colour-map.
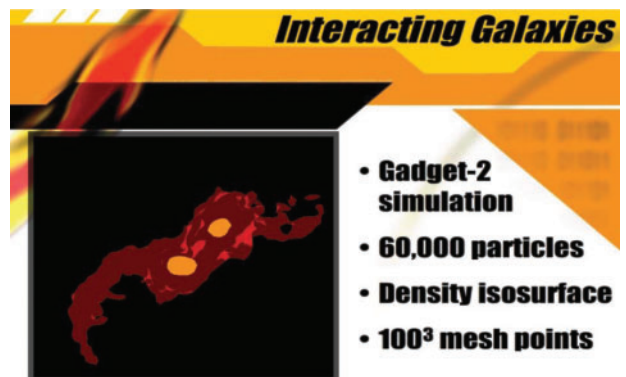
PAPERVISION3D is a software-only renderer (i.e. not hardware accelerated), it is not currently feasible to export the full resolution geometry for either isosurfaces or volume rendering. Instead, there is a trade-off between download speed (due to file size of the geometry), the interaction rate, and the level of detail displayed. As a compromise, the S2WEB versions use a 60^3-cell mesh for the isosurface calculation and a 40^3-cell mesh for the volume rendering.

An example of using isosurfaces to display a galaxy merger is available online[10]. For this example, we have pre-loaded 5 GADGET-2 snapshots. Within the S2PLOT code, these are given unique FRAME* number identifications for the VRML output. S2WEB determines which of the frames should be displayed and the user can step through the sequence by pressing the spacebar.

A particular application of implicit user-control of the S2PLOT model tree is found in axial-slice volume rendering: a 3D gridded dataset is used to generate three sets of textures, one set along each lattice axis of the volume (e.g. Lacroute & Levoy 1994). Each set is contained within its own branch of the model tree (using the predefined VRSET* strings with the pushVRMLname function) and then Adobe 3D JAVASCRIPT (for PDF) or Adobe ACTION-SCRIPT 3 (for FLASH) code is written which selects the best slice set to draw for the current camera location and view direction. The FLASH application can be viewed online[11].

---

[10] http://astronomy.swin.edu.au/s2plot/
interchange/iso.

[11] http://astronomy.swin.edu.au/s2plot/
interchange/vr.

Figure 6 Displaying the interactive view of a GADGET-2 galaxy merger via the S2SLIDES application for use in an academic presentation. The S2PLOT desktop application is overlaid by S2SLIDES on top of a background image. The background image can be created by any image or presentation editor; in this case Microsoft POWERPOINT was used, and the image was exported as a TGA-format file.

Figure 6 shows an example of an S2SLIDES presentation using the galaxy interaction visualization application. The same S2PLOT code that is used in the desktop application is executed as a child process by S2SLIDES. The position of the application window on the background slide is customisable, and a fullscreen mode is available via a keystroke.

While Microsoft POWERPOINT and Apple KEYNOTE do permit the use of embedded interactive datasets on slides through a VRML plug-in, they only provide a limited interaction capability, and do not support advanced displays. One of the main advantages of S2SLIDES is that, as an

S2PLOT application itself, it can be used with advanced displays. This reusability means that a presentation can be given in 2D via a laptop, but if a stereoscopic display is available, the presentation can be given in stereo simply by changing an environment variable for the display type.

Some S2PLOT programs may load large data when they run. In order to alert the presenter and audience that something is happening, we utilize a spinning logo while applications are starting up.

5 Future Work

We have completed most of the work involved in writing the S2PLOT model tree structure to VRML 2.0 format, and we have demonstrated most of the capabilities required for the implementation of a common user interface across desktop, e-print (PDF) and Web (FLASH) targets. The latest version of S2PLOT (2.4, October 2008) can export nearly all of the S2PLOT primitives to VRML, including transparent facets, billboards and interactive handles.

A binary distribution of the S2WEB FLASH application and a template HTML file is provided on the S2PLOT website. We plan to release the source code for the S2WEB FLASH application in December 2008. We will include the code that supports interactive handles using the HANDLE* naming convention in the VRML file; provide animation capabilities using the FRAME* naming convention; and support axial-slice volume rendering using the VRSET* naming convention. Users will be able to build the standard S2WEB application using the free Adobe FLEX 3 SDK, or make simple modifications to the code for custom animation and interaction. The combination of the S2PLOT programming library and the S2WEB FLASH application yields a free and straightforward technique for astronomers and other scientists to generate 3D visualizations on the desktop and transfer them to the Web.

The prospects for future development of the integrated S2PLOT solution for 3D analysis, collaboration, publication, education and outreach are excellent. Specific areas for improvement and development are summarized below.

Based on PAPERVISION3D and FLASH PLAYER 9, S2WEB is not hardware accelerated. Accordingly, its rendering speed, or equivalently frame rate, depends on both the size of the FLASH window within the Web browser, and the complexity of the 3D geometry being displayed. Volume rendering of data sets larger than around 32^3 pixels, splatting of more than a few thousand billboards, and the display of complex transparent isosurfaces, can become slow. However, Adobe FLASH PLAYER 10 now offers 3D graphics accelerated by hardware and is presently in beta release. It is likely that PAPERVISION3D will be modified to provide hardware accelerated rendering, thereby improving the usability of S2WEB without further work from us. While we wait for this upgrade though, we intend to assess the merits of re-writing S2WEB directly for the FLASH PLAYER 10, without using the PAPERVISION3D classes.

Our S2PLOT philosophy has been to support multiple, advanced display devices from a single library code base.

It should be reasonably straightforward to generate real-time, stereoscopic 3D figures within Web browsers, by interleaving the left and right eye views when a suitable desktop stereo display is available.

Integration of Web-deployed 3D assets with our in-house Content Management System (CMS) would enable straightforward inclusion of 3D figures and diagrams in Swinburne Astronomy Online[12] course material and the COSMOS Encyclopedia of Astronomy[13]. For example, the COSMOS entry on hierarchical merging (Figure 7) contains a version of the interactive halo merger tree as described in Section 4.2. With regards to learning and teaching benefits, interactive presentation and exploration of scientific data promotes student-centred active learning. There is a natural opportunity to provide the required interactivity via 3D enabled Web sites, including presentation of conceptual models, geometrical models, and datasets from both observation and numerical simulation. Interactive applications can be enhanced through features such as dynamics, selection, feedback, etc., all of which are feasible with our FLASH/ACTIONSCRIPT 3 approach.
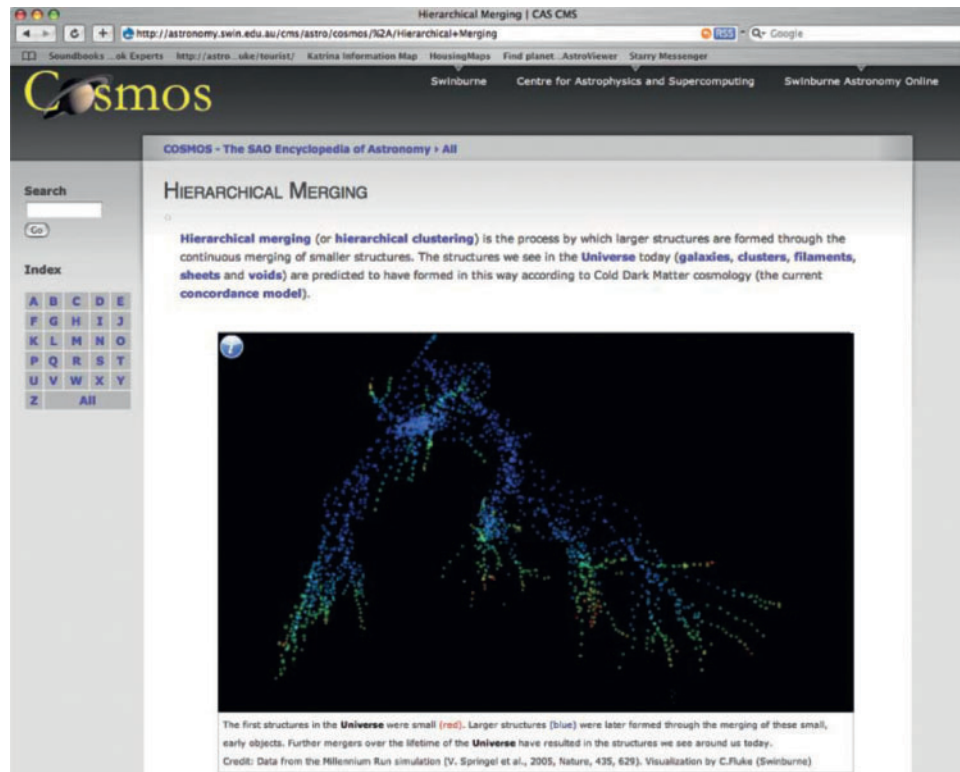
Our 3D PDF solution requires commercial software, specifically Adobe ACROBAT 3D or Adobe ACROBAT 9 PRO EXTENDED. We are cogniscent of the risks of building upon proprietary (and often expensive) software products, however there are few alternatives at this stage:

- We have examined the Universal 3D (U3D) File Format.[14] If we could write suitable U3D files, then these could be embedded in PDF files as 3D annotations using the excellent MOVIE 15 LATEX style created by Alexander Grahn. Unfortunately there is no good reference implementation of the U3D standard, and its originators appear to have lost interest in maintaining or even using the U3D format. Furthermore, we have been unable to preserve important visualization attributes (such as point colour) when experimenting with U3D files using the Adobe software products.
- The better format for storing 3D models in PDF files is Adobe's PRC format. We are not aware of any reference implementations of the PRC format, and the specification has historically been very hard to locate. While we cannot write PRC format directly from S2PLOT at this stage, it may be possible in the future if Adobe are forthcoming with an open specification of the file format. In the meantime, what *is* possible is to use an Adobe product *once* to import S2PLOT-generated VRML, save the resultant PDF file, and then extract the model data in PRC format using a tool like pdftk. Once the PRC data is extracted, it can be easily added to documents generated with pdflatex using the aforementioned MOVIE 15 LATEX package.

---

[12] SAO: `http://astronomy.swin.edu.au/sao`.
[13] COSMOS: `http://astronomy.swin.edu.au/cosmos/`.
[14] Standard ECMA-363 Universal 3D File Format: `http://www.ecma-international.org/publications/standards/Ecma-363.htm`.

Figure 7 Embedding an interactive merger tree visualization in a COSMOS encyclopedia entry. A rainbow colour map is used to indicate progenitor redshifts (red = earliest, blue = latest), and there is no scaling of splat sizes.

6 Concluding Remarks

We have described our S2PLOT-based system for creating 3D visualizations, and our extensions which use VRML as an interchange format to enable the deployment of almost identical 3D visualizations across multiple targets: the desktop (S2PLOT application); the electronic article (3D PDF); academic presentation via S2SLIDES, or alternatively and more portably, using 3D PDF documents created with the beamer or prosper LATEX packages; and the mainstream Web (FLASH S2WEB application). The two key standards that our system builds on are the formal 3D model interchange format (VRML and X3D) and the informal user interface: common camera and rendering controls across all output targets.

We have used cosmological visualization as a specific example of the requirements and benefits of interchange between different target modes. S2PLOT provides the basic set of graphics primitives necessary to represent wedge diagrams from galaxy redshift surveys and large-scale structure from cosmological simulations. Advanced functionality is also available, such as volume rendering, splatting and isosurfaces. We have demonstrated how cosmological visualization applications built on these graphic capablities may be interchanged and accessed interactively at the desktop, on the Web, and during presentations. It is evident that this approach applies equally well across other fields of astronomy research.

While a range of standards-compliant (e.g. Wells, Greisen & Harten 1981; Ochsenbein et al. 2004) processing, analysis and visualization tools exist, astronomers tend to use their own data formats for intermediate steps of processing (particularly for theory and simulations). This avoidance of data standards makes interchange of data and intermediate results between collaborators a complex process. It is often only at the completion of a research project that astronomers may chose to 'publish' their data using an interchange format, and increasingly this includes Virtual Observatory defined standards (Norris et al. 2007). In most cases, it is easier to share still frames or animated sequences from data as intermediate products.

Our approach provides a pathway for the sharing of intermediate and final data products in an interactive, 3D format. The goal here is that interaction with the Web-based version should be as similar as possible to the application that was used to explore the data initially on the researcher's desktop. Using S2WEB, interactive models could be shared online for a 'quick look', in the same way that astronomers regularly put images on a personal or project Web site.

A secondary benefit of interactive visualization in the presentation and publication modes is the additional level of scrutiny available. Rather than selecting a particular 'best' view for a 2D still image or animation sequence, the viewer is able to explore the dataset for themselves,

either confirming or rejecting the conclusions as originally proposed.

We believe that 3D model interchange format need to be augmented with a common user interface before reasonable uptake by the community, who have limited time to learn new tools and techniques, can be expected. The user interface includes interactive functionality such as the control of viewpoint that has never been required for classical 2D graphics, but is axiomatic to 3D visualization. By simplifying the deployment of interactive, 3D visualization tools that share a common user interface across multiple media (desktop, e-print, presentation, Web), we aim to properly enable and stimulate the strategic uptake of 3D visualization techniques in astronomy.

Acknowledgments

References

Abt, H. A., 2003, in Information Handling in Astronomy - Historical Vistas, Astrophysics and Space Science Library, Ed. Heck, A. (Dordrect: Kluwer Academic Publishers), 285, 127

Barnes, D. G., Fluke, C. J., Bourke, P. D. & Parry, O. T., 2006, PASA, 23, 82

Barnes, D. G. & Fluke, C. J., 2008, NewA, 13, 599

Beeson, B., Lancaster, M., Barnes, D. G., Bourke, P. D. & Rixon, G. T., 2004, in Optimizing Scientific Return for Astronomy through Information Technologies, Eds. Quinn, P. & Bridger, A., SPIE, 5493, 242

Cole, S., Lacey, C. G., Baugh, C. M. & Frenk, C. S., 2000, MNRAS, 319, 168

Croton, D. J. et al., 2006, MNRAS, 365, 11

Crutcher, R. M., Plante, R. L. & Rajlich, P., 1998, in Astronomical Data Analysis Software and Systems VII, ASPC 145, Eds. Albrecht, R., Hook, R. & Bushouse, H., 3

De Lucia, G. & Blaizot, J., 2007, MNRAS, 375, 2

Drebin, R. A., Carpenter, L. & Hanrahan, P., 1988, Computer Graphics, 22, 65

Dubinski, J., 2008, NJPh, 1015002

Eichhorn, G., Kurtz, M. J., Accomazzi, A., Grant, C. S. & Murray, S. S., 1994, BAAS, 26, 1371

Fluke, C. J. & Barnes, D. G., 2008, AER, 7, 113

Genova, F. et al., 1996, VA, 40, 429

Globus, A. & Raible, E., 1994, Computer, 27, 86

Hockney, R. W. & Eastwood, J. W., 1988, Computer Simulation Using Particles (New York: Taylor & Francis Group)

Hopf, M. & Ertl, T., 2003, in VIS '03: Proceedings of the 14th IEEE Visualization 2003, 15

Jones, D. H., Saunders, W., Read, M. & Colless, M., 2005, PASA, 22, 277

Kurtz, M. J., Eichhorn, G., Accomazzi, A., Grant, C. S., Murray, S. S. & Watson, J. M., 2000, A&AS, 143, 41

Lacroute, P. & Levoy, M., in SIGGRAPH '94: Conference Proceedings, Ed. Cunningham, S. (New York: ACM), 451

Murray, S. S., Brugle, E. W., Eichhorn, G., Farris, A., Good, J. C., Kurtz, M. J., Nousek, J. A. & Stoner, J. L., 1992, in Astronomy from Large Databases II, European Southern Observatory Proceeding, Eds. Heck, A. & Murtagh, F., 387

Norris, R. P. et al., 2007, HiA, 14, 673

Ochsenbein, F. et al., 2004, in Towards an International Virtual Observatory: Proceedings of the ESO/ESA/NSA/NSF Conference, Eds. Quinn, P. & Górski, K. M., 118

Perryman, M. A. C. & ESA, 1997, The HIPPARCOS and TYCHO catalogues, SP1200

Plante, R. L., Rajlich, P. J., Pietrowitz, S., Xie, W. & Qamar, A., 1999, in Astronomical Data Analysis Software and Systems VIII, ASPC, 172, Eds. Mehringer, E., Plante, R. & Roberts, D., 433

Quinn, P. J. et al., 2004, in Optimizing Scientific Return for Astronomy through Information Technologies, Eds. Quinn, P. & Bridger, A., SPIE, 5493, 137

Springel, V., Yoshida, N. & White, S. D. M., 2001, NA, 6, 79

Springel, V., 2005, MNRAS, 364, 1105

Springel, V. et al., 2005, Natur, 435, 629

Tufte, E. R., 1990, Envisioning Information (Cheshire, CT: Graphics Press)

Tufte, E. R., 2001, The Visual Display of Quantitative Information (2nd edition; Cheshire, CT: Graphics Press)

Wells, D. C., Greisen, E. W. & Harten, R. H., 1981, A&AS, 44, 363

Westover, L., 1989, in Proceedings of the Chapel Hill Workshop on Volume Visualization, 9

Westover, L., 1990, Computer Graphics, 24, 367

White, S. D. M. & Frenk, C. S., 1991, ApJ, 379, 52