S2PLOT: a Straightforward Library for Advanced 3-dimensional Scientific Visualisation

David G. Barnes, Christopher J. Fluke

Centre for Astrophysics and Supercomputing, Swinburne University of Technology, Hawthorn 3122, Australia

Abstract. S2PLOT is a user-oriented programming library for generating and exploring 3-dimensional (3-d) scientific plots and diagrams. It provides a lightweight interface—inspired by the simple yet widely-used PGPLOT—to produce hardware-accelerated visualisations of point, line, image and volumetric data. S2PLOT provides C and FORTRAN interfaces, and supports monoscopic, stereoscopic and curved (eg. dome) display devices. PGPLOT-savvy astronomers can usually write their first S2PLOT program in less than ten minutes. In this paper, we introduce the latest S2PLOT version and highlight major new additions to the library, including volume rendering and isosurfacing of astronomical data. We describe a simple extension that enables the embedding of large-area FITS images directly into S2PLOT programs using standard World Coordinate Systems, and we introduce the Python interface to S2PLOT.

1. Introduction

S2PLOT is a programming library written in the C language, providing routines for 3-dimensional (3-d) scientific graphics (Barnes et al. 2006). It provides interactive, qualitative and quantitative functions, and can render to desktop, stereoscopic and curved display devices. Advanced graphical entities such as volume renderings and isosurfaces can be generated and displayed with just a few function calls. S2PLOT version 2.0 features an integrated presentation tool (s2slides) for giving scientific talks featuring realtime 3-d graphics, and a save-to-VRML feature which produces output suitable for converting S2PLOT visualisations to 3-d figures in PDF files (Barnes and Fluke, 2007).

Radio, optical and X-ray observations routinely collect data with three (2-d angular position, 1-d frequency) or more dimensions, while spatial simulation codes universally generate data with six (3-d position, 3-d velocity) or more dimensions. The primary motivation for S2PLOT is to make it easy for astronomers to use 3-d graphics to explore and analyse such multi-dimensional data. To this end, around one-third of the S2PLOT functions are directly related to the equivalent 2-d functions from the widely-used PGPLOT library, making S2PLOT simple to use for astronomers who have previously used PGPLOT. S2PLOT is also motivated by our strategy of making 3-d graphics worthwhile for astronomers: this we are doing by focussing on the surveyed needs of astronomers (Fluke et al. 2006), extending the library to include selected scientific and astronomical functions, and providing tools for the inclusion of 3-d graph-

1PGPLOT is written by T. Pearson; http://www.astro.caltech.edu/~tjp/pgplot
ics in oral and written presentations. In this paper, we describe the following features of S2PLOT: realtime volume rendering, interactive handles, and FITS images as textures.

2. Realtime volume rendering

Volume rendering is a well-known method for visualising 3-d data volumes (eg. Drebin et al. 1988). It entails casting rays through a data volume, applying per-volume element (voxel) absorption to the light as it passes through the volume, and optionally applying per-voxel emission characteristics. True volume rendering can dramatically improve comprehension of a complicated density field but is a slow operation. S2PLOT realises realtime frame rates for moderate size volumes (eg. $64 \times 64 \times 64$ voxels) by layering transparent textures on top of each other. While not a true volume rendering (emission is not supported, and highly oblique angles through the volume exhibit mild over-absorption), the overall impression is very close to standard volume rendering. Figure 1 illustrates how texture planes are chosen and stacked for volume rendering in S2PLOT.

Two basic light transmission models are available in S2PLOT: the classical painter’s algorithm which allows opaque foreground features to obscure the background, and a non-absorbing model in which rendered pixels only get brighter as foreground texture planes are added to the image. The best model for a given data volume generally depends on the complexity of the volume and the noise level.

3. Interactive handles

Quantitative analysis of multi-dimensional volumes requires some way to tag objects or regions of interest. Our approach in S2PLOT is to provide handles—entities within the visualisation that can be clicked on and (optionally) moved by the user. Handles in S2PLOT are drawn as rectangular textures that always face the camera. The programmer receives feedback by way of a callback function.
when a handle is clicked, and apart from responding to the click with some operation (eg. fetch information on the selected item), can easily change the texture or colour of the handle to reflect a state change. Dragging is effected similarly, with a separate callback returning the new 3-d coordinates of the handle. Handles can be dragged in the plane of the viewing screen, and moved orthogonal to the screen plane (in and out of the space) using the scroll wheel. Possible uses of handles include:

- tagging catalogue objects, with clicks leading to further object information being displayed, or object identifiers being exported to a PLASTIC hub;\(^2\)
- as control points for 3-d regions of interest, with dragging enabled to move and reshape the region; and
- as basic user interface elements such as sliders, by using screen coordinates (2-d drawing) in combination with handle dragging constrained to one axis.

4. FITS images

Spectral line data cubes can be displayed using S2PLOT’s built-in volume rendering and isosurface functions. Even though such data are not physically 3-d, there are merits to treating them as if they are (eg. Borkin 2008; Goodman 2008; Oosterloo 1994). 3-d FITS images can also be displayed as sequences of channel maps or position-velocity slices. There is one clear benefit of showing these slices within a 3-d system: the location of the currently displayed slice is immediately apparent from the placement of the slice within the projected edges of the data cube. The viewer does not need to refer to a slice number or velocity value to ascertain the location of the displayed slice.

For 2-d FITS images, especially large-area images of many tens of square degrees, S2PLOT can be used to re-project the image directly back on to a virtual sky for viewing. This is accomplished by calculating a grid of vertex coordinates using the world coordinate system in the FITS image, and applying the FITS image as a texture to this vertex grid. Whereas the wholesale regridding of an image can be a time-consuming task, the OpenGL internals of S2PLOT make the re-projection to the virtual sky a realtime operation. Figure 2 illustrates how a vertex grid is used to place a FITS file on a curved sky.

5. Additional features

S2PLOT version 2.0 includes a Python interface to the library. The Python module includes all of the callback and advanced texture-handling functions of S2PLOT. With many large and popular astronomy software packages now using or including Python interfaces, we hope the availability of a Python S2PLOT module will make 3-d graphics even more accessible to the community. S2PLOT also now includes a save-to-VRML feature which can be used to export VRML files for import into Adobe Acrobat 3D. This commercial software can embed interactive, 3-d figures in PDF documents. Simple JavaScript fragments can be added to the figures to provide high-level controls of the visualisation, or implicit

---

\(^2\)Platform for Astronomical Tool Interconnection; http://plastic.sourceforge.net/index.html
Figure 2. Fast re-projection of a large-area 2-d FITS image to a curved virtual sky in S2PLOT: (a) vertex grid plotted on “sky shell” after transformation to world coordinates; (b) with FITS image textured onto the vertex grid. Magellanic Stream neutral Hydrogen emission image from Putman et al. (1998).

Acknowledgements. We thank Nick Jones for developing the Python S2PLOT module.

References

Oosterloo, T. 1995, PASA, 12, 215
Putman, M.E., Gibson, B.K., Staveley-Smith, L., et al. 1998, Nat, 394, 752