Astronomical Data Analysis Software and Systems VIII
ASP Conference Series, Vol. 172, 1999
D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds.

SAOTk: Tcl/Tk Widgets for Imaging and Data Visualization

William Joye, Eric Mandel

Smithsonian Astrophysical Observatory, Cambridge, MA 02138

Abstract. *SAOTk* is an integrated set of *Tcl/Tk* widgets for astronomical imaging and data visualization. The widget set is composed of an Image, Panner, Magnifier, and Colorbar widget. In addition to "classical" support for imaging *FITS* data, manipulating colormaps, region marking, and coordinate readout (including WCS), *SAOTk* widgets also support arbitrary image scaling and rotation, advanced printing and graphics, image mosaics, and shared memory.

The *SAOTk* widgets will be incorporated into the next major release of *SAOtng*, a release that will allow the astronomical community to begin moving away from its primary dependence on *Unix* and *X11*. They also can be incorporated into any standard *Tcl/Tk* environment to build custom data analysis and visualization applications.

1. Introduction

For more than fifteen years, software developers at Smithsonian Astrophysical Observatory have chased the elusive goal of providing a perfect image display program to the astronomical community. We started our quest in the early 1980's by developing *SAOimage*, one of the very first astronomy programs to utilize the *X Window System* within the *Unix* workstation environment. *SAOimage* broke new ground in astronomical image display by providing an intuitive and easy to use graphical menu/command button interface. It has since been used by thousands of astronomers around the world.

In the early 1990's, we introduced *SAOimage: The Next Generation (SAOtng)*. The aim of this new offering was to move beyond the concept of a static and stand-alone image display. We wanted to allow users to extend the program beyond the functionality that we had compiled and linked into it. We also wanted to give users the ability to control the program from external programs, scripts, etc. Toward this end, we developed the *X Public Access (XPA)* mechanism, which enabled *SAOtng* (and other programs) to define named "public access points" for communication of data and commands with other programs. *XPA* offered public access to traditionally internal *SAOtng* algorithms such as image file loading and scaling. It also supported user-configurable analysis of image data and bi-directional communication with other processes.

In this new phase of our work, we aim to move beyond the bounds of any particular image display program and even beyond the bounds of the *Unix* operating system that has been so central to astronomical software over the past decade and a half. We therefore have developed *SAOTk*, an integrated set of *Tcl/Tk* widgets for astronomical imaging and data visualization.

Table 1. SAOTk Widgets

Image	FITS viewer for a single image or a mosaic of images
Colorbar	control of imaging color environment
Panner	thumbnail image for position control
Magnifier	magnified view of a section of the image

The *SAOTk* widgets will be incorporated into the next major release of the *SAOimage* line of image display programs, a release we have code-named *DS9*. The availability of these widgets, along with *DS9*, will allow the astronomical community to begin moving away from its primary dependence on *Unix* and *X11*. The widgets themselves are designed to be incorporated easily in any standard *Tcl/Tk* environment to build custom data analysis and visualization applications.

2. The SAOTk Image Widget

The Image widget is used to display *FITS* data. It supports both *FITS* images and binary tables. Multiple extension *FITS* files are also supported. The widget supports arbitrary zoom, rotation, orientation, and panning. The *FITS* data are maintained in raw form. `BSCALE`, `BZERO`, and `BLANK` keywords are supported.

The Image widget is designed to support rendering of large *FITS* images and binary tables. For images, the rendering time is constant, and is based on the size of the screen, not the size of the data. Thus, the widget is capable of rendering very large images (8k $\times$ 8k or larger), or a large mosaic of images (such as 36 images, each 2k $\times$ 4k). Since all rendering is done directly from the raw *FITS* data, memory requirements are very small.

For *FITS* binary tables, the rendering time is based on the number of filtered rows or events. Since the image is binned *on the fly*, and then rendered to the screen, memory requirements again are very small. Of course, if the data are sorted by position, rendering time is much faster, since the widget can calculate which segments of a table contain the events that need to be processed.

The Image widget supports display of a mosaic of images, where each image segment is a separate *FITS* image. These segments may be stored in one multi-extension file, or may be located in a number of different files. The widget uses *FITS* mosaic keywords to place each image in its proper place in the overall image space.

One special feature of the Image widget is the ability to rotate, orient, and zoom an image to align its display to an arbitrary WCS. This feature is very useful for making comparisons between images from different instruments or telescopes that have different coordinate systems or WCS parameters.

As with all 8-bit, Pseudo-Color displays, each *FITS* pixel data value is scaled to a display value between 0 and 255. This scaling is done *in real time*, so that the widget may be used for real-time applications. The mapping from the raw data to the scaled value is based on the start/stop clip values and the scaling algorithm. Clipping values can be defined using minimum/maximum data values, *IRAF* zscale, or user-specified values. Scaling algorithms currently include `linear`, `ln` (from *SAOimage*), `log` (from *IRAF*), and `sqrt`. Users can easily add new scaling algorithms.

If the data are contained in a *FITS* binary table, the events are filtered and binned *in real time*, based on the current pan, zoom and rotation values. Events may be filtered using any column, and spatial region masking also is supported.

The Image widget can read *FITS* data from a file, `STDIN`, or a *TCL* Channel. It also supports a number of different memory management options: memory can be allocated as needed (and swapped to disk), it can be mapped directly to the *FITS* file *(mmap)*, or the *FITS* data can be accessed from a shared memory segment. The default is *mmap*.

The Image widget supports a variety of coordinate systems, as defined in the *FITS* header: `image, ccd, detector, amplifier,` and WCS `(fk4, fk5, galactic, ecliptic, linear).`

The Image widget supports *PostScript* level 1 and level 2 printing. This is not a screen capture process, but a full-featured *PostScript* driver. A variety of document sizes are available, and grayscale and color imaging are supported. Level 2 output is compressed via `RLE` and encoded via `ASCII85`, so the size of the output *PostScript* data is usually very small.

The Image widget also supports a variety of markers or regions, including `circle, ellipse, rectangle, polygon, line,` and `point`. A marker's position is attached to the image and may be specified in any of the supported coordinate systems. Each marker has a number of properties, including color, text, and name. Each marker also allows specification of behavior such as whether the marker can be selected, edited, moved, rotated, deleted, etc.

3. The SAOTk Colorbar Widget

The Colorbar widget controls the color environment for the application. Currently, it requires an 8-bit Pseudo-Color visual, though this need not be the default visual. This requirement will be removed in a future release.

The Colorbar widget supports a variety of colormap formats, including colormaps from *SAOimage, SAOtng/ximtool,* and other display programs. There are a number of built-in maps, and new maps may be loaded, edited, and saved.

The widget will request a minimum number of colors in the default colormap. If these are not available, it will use a private colormap. The private colormap will be populated with colors from the default colormap to minimize color flashing.

4. The SAOTk Panner and Magnifier Widgets

The Panner and Magnifier widgets are designed to work with the Image widget. Each Image Widget can be assigned to work with one Panner widget and one Magnifier widget.

The Panner widget displays a thumbnail image of the data being displayed, in order to support of fine position control. It also overlays an image orientation compass and a WCS orientation compass. The WCS compass may be positioned at the center of the image or at the tangent point.

The Magnifier widget provides a magnified view into the data. Two magnification algorithms are available: `Fast` and `More Accurate`. The later is especially useful for non-integer zoom and rotation values. The magnification value may be user-specified, but usually is a power of two.

5. DS9: Putting It All Together

DS9 is our code-name for the next version of the *SAOimage* family of display programs. It is a *Tcl/Tk* application which utilizes the *SAOTk* widget set. It also incorporates a new version of *XPA* (an implementation which is independent of the *X Window System*) to allow external processes to access and control its data, GUI functions, and algorithms. *DS9* will support all capabilities in the current version of *SAOtng*, including direct display of *FITS* images and binary tables, multiple frame buffers, region/cursor manipulation, many scale algorithms and colormaps, and easy communication with external analysis tasks. It is highly configurable and extensible to meet the evolving needs of the astronomical community.

DS9 is a stand-alone application that requires no installation or support files. Preliminary versions of *DS9* exist for Sun Solaris 2.5/2.6 and Linux. Other Unix platforms will be supported in the near future. Ports to Apple Macintosh and Microsoft Windows also will be available in the next year. All versions and platforms will support a consistent set of capabilities.

DS9 supports advanced features such as multiple frame buffers, mosaic images, tiling, blinking, geometric markers, colormap manipulation, scaling, arbitrary zoom, rotation, pan, and a variety of coordinate systems (including Image, CCD, Detector, and WCS). *DS9* also supports *FTP* and *HTTP* access.

The GUI for *DS9* is user-configurable. GUI elements such as the coordinate display, panner, magnifier, horizontal and vertical graphs, and colorbar can be configured via menus or the command line.

An exciting new feature of *DS9* is the ability to display multiple images in tile mode, where each image is scaled, oriented, and rotated to match WCS parameters. By locking cross-hairs, features between images are easily compared.

6. Conclusion

More information about the *SAOTk* widgets, *DS9*, *XPA*, or *SAOtng* (including our latest software offerings) can be found on the World Wide Web at:

`http://hea-www.harvard.edu/RD/`

While our goal of developing a perfect image display program has been elusive thus far, we are confident of achieving it the next release.

Acknowledgments. This work was performed in large part under a grant from NASA's Applied Information System Research Program (NAG5-3996), with support from the AXAF Science Center (NAS8-39073).