A Communications Interface for SAOimage

Donald H. Gudehus

CHARA, Georgia State University, Atlanta, GA 30303

Electronic mail: gudehus@chara.gsu.edu

Published in *Publications of the Astronomical Society of the Pacific* **107**, 199 (1995)

A Communications Interface for SAOimage

Donald H. Gudehus

ABSTRACT.

An interface to the X window pseudocolor display program, SAOimage, has been written to allow the easy display, visualization, and analysis of astronomical data. A user program can now be written with little effort to send images, cursors, vectors, and look-up tables to SAOimage, as well as to read back cursor coordinates. A special feature of the enhancement package is the ability to load several images into the display at the same time, with correct reading of the mouse pointer coordinates in each image. It is also possible to send successive frames of data in a slow video mode to visualize time-variable phenomena or to aid in the real-time acquisition of data. All the source code and demonstration programs are available over anonymous ftp.

Keywords: methods: data analysis – techniques: image processing

1. INTRODUCTION

1.1. The SAOimage Program

SAOimage (Worrall et al. 1990; VanHilst 1991) is a software package which can display pseudocolor images on hardware platforms which support the X window system. Computers running OpenVMS and most flavors of the Unix operating system are currently supported. It is thus possible to run SAOimage on computers which use a variety of processors available from several manufacturers. In addition, the display can be viewed and controlled on computers using other operating systems, providing that an X window display server is used. For example, with SAOimage running on a VAXstation, the display could be viewed and controlled on a Macintosh running MacX. Besides pseudocolor display, with an image area mainly limited by the computer's memory, SAOimage provides a rich supply of interactive controls such as zoom, pan, pseudocolor look-up table (LUT) control, and cursor positioning. SAOimage can also read from data files and LUTs, write cursor information to disk, and send displayed images to a printer. SAOimage can therefore replace bulky and noisy dedicated hardware image displays and provide intelligent control as well. Because SAOimage is in the public domain, it can provide a cost effective solution to many image display needs.

On the the other hand, a dedicated hardware image display usually has more bits for pseudocolor display (SAOimage can display 200 levels), has true overlay planes, can draw vectors and do rectangular fill at high speed, and has multiple hardware cursors under convenient and precise control of joysticks. More importantly, a hardware display can easily be placed under almost complete control of a user's program.

The purpose of this paper is to report the development and availability of an easy-to-use communications interface which increases the usefulness of SAOimage for the display and visualization of scientific data.

1.2. Desirable Enhancements to SAOimage

1.2.1 Data Input

From the perspective of a researcher requiring the display of scientific data, several needs stand out prominently. One would like, for example, to be able to display in pseudocolor a two-dimensional array of data created or opened by a user computer program. While SAOimage does accept input from the IRAF package, available from the National Optical Astronomy Observatories (NOAO), a custom user application is not easily interfaced to IRAF, and even if this were done one would then have the overhead and constraints of that package. In addition, IRAF uses an unusual file structure consisting of a header file and a data file, located in different directories, an arrangement not compatible with some existing databases. In a future version, however, IRAF is expected to be able to accept Flexible Image Transport (FITS) format files. SAOimage will accept non-IRAF formats, but this can only be carried out by specifying on SAOimage's command line the file name, pixel width, pixel height, data type, byte offset, and possibly a byte swap directive (if bits and bytes do not both increase in significance as memory location increases). A user program in communication with SAOimage would be expected to have these parameters readily available once a file was opened or an array created, so that the inconvenience of repetitive and unnecessary typing could then be eliminated.

1.2.2 Data Display

SAOimage displays two-dimensional data in pseudocolor or grayscale on multibit-displays, or in dithered mode on 1-bit displays. Computer memory is reserved for the display area by specifying in a setup file the desired size of a display buffer. Usually this size is matched to the size of the image to be displayed. In a more general implementation of a display program, one would like to be able to easily change the size of the buffer during program operation, display images of various sizes within this buffer, be able to display more than one image at a time within the buffer, and be able to display images in rapid succession to simulate video.

The version of SAOimage used as the basis for the present enhancement, version 1.07, did not fully possess these more general capabilities and did not always behave as expected when images of various sizes were loaded. For example, if the screen, or desktop, width of SAOimage was larger than the buffer width, the image would initially be aligned with the left side of the desktop, be rolled over, and display incorrect coordinates. Operation of the zoom or pan controls would then correct the display. Some users make their screen width the same as their chosen buffer width to counteract this effect, but this is inconvenient and also restricts the viewing area when the image is zoomed. Another problem was that if a single line of data with a number of pixels less than the width of the buffer was sent to SAOimage, the image initially appeared only in the pan box and zoom boxes, until the zoom or pan controls were used.

The greatest SAOimage limitation was that many independent images of arbitrary size could not be displayed at one time. Multiple image display is scientifically useful when comparing exposures made under different conditions (different filters, exposure times, seeing, etc.), comparing images at different stages of processing (before and after flatfielding, removal of fringes, removal of overlapping objects, etc.), and comparing an image of data with an image of a model. This kind of comparison complements blinking, which SAOimage incorporates, but is done in space rather than time. The ability to load multiple images is also important if it is desired to build up a mosaic of a field from several smaller images. The small fields of CCD exposures make this capability very attractive. While a single file of larger size could be constructed by piecing together smaller files, this would require additional disk storage as well as additional labor. Although the current version of IRAF does allow one to mosaic several images into the display, the coordinates read back are all relative to the last image loaded, and the position on the frame is specified in inconvenient normalized coordinates, rather than frame pixels. Also if one attempts to load a single vertical line of data, a much larger patch of data is actually written to the display.

Because of the overhead associated with loading an image, i.e., entering the filename, and perhaps other information, on the command line, simulated video was also not feasible with the original standalone version of SAOimage.

Several other anomalies associated with the display of data were noted in version 1.07. Among the more important were *i*) when a 32-bit floating-point number contained in a FITS file was displayed from a VAXstation, the displayed pixel intensities were too small by a factor of 4; *ii*) when a FITS file was loaded prior to an array-type file, the intensity scaling and bias from the FITS file were retained, causing the displayed intensities to be incorrect.

1.2.3 Coordinates and Overlays

Image displays have three coordinate systems which will be referred to here as the screen, frame, and file systems. For SAOimage, the screen coordinates are referenced to the SAOimage desktop, the size of which is controlled in a setup file; the frame coordinates are referenced to the buffer; and the file coordinates are referenced to the file being displayed. The screen coordinates have their origin at the upper left of the screen, and for a given point in the file or frame depend on zoom and pan. The frame coordinates have their origin at the upper left of the frame, and are likewise dependent on zoom and pan for a given screen coordinate. The file coordinates have their origin at the lower left of the file and, relative to the frame, are dependent on the offset and expansion factor used when loading the file.

As the mouse pointer is moved over the field of a display device, a knowledge of each of the above three coordinates is important for effective image processing. The screen coordinates are usually used as a controlling input for the color LUT. Typically, a decreasing vertical position inceases the number of color assignments per input intensity interval, and a changing horizontal position shifts the color assignments. The file coordinates are needed for reading an intensity value from the file. In the original version of SAOimage, only the file coordinates were returned, and the screen coordinates were only used internally for LUT control. If several images are loaded into the buffer however, the frame coordinates are necessary for the determination of which image the cursor is located in. From this information, the communicating program can then determine the actual file coordinates. The screen coordinates can also be of use to the communicating program if more precise control of the LUT is desired. For example, the user may wish to return to a previous LUT setting under program control.

Dedicated image displays usually have overlay planes which can be written to in order to display erasable patterns of various kinds without disturbing the underlying data. SAOimage does not have an independent overlay plane (a limitation of X window), but allows the overlay patterns (called cursors in SAOimage) to share the available bits with the data. While the cursors available in SAOimage are quite sophisticated, giving the user control over the sizes, shapes, and orientations of circles, ellipses, rectangles, arrows, annului, and polygons, their application to scientific analysis required some enhancements.

Three typical user operations illustrate the requirements of overlays for astronomical image processing. When interactively editing regions of contamination or bad pixels, the user defines an area to be revalued with the mouse cursor. The region is then illuminated with an outline which can be erased if the user decides not to proceed. Upon proceeding, the outline is made nonerasable so that the user can keep track of which areas have been edited. In aperture photometry the user draws three concentric circles to define the object area and the sky annulus. While these sizes should be consistent from object to object, some initial trial values need to be tested. Thus, until the user decides to proceed, the circles need to be erasable. After a measurement is made, the annuli are retained in a set of semipermanent annuli. The third operation is simply the display of objects in a list. Circles are drawn around each object's location to allow identification and inspection of the listed objects. Here many semipermanent circles of possibly different radii are drawn on the screen.

1.2.4 Look-up Table

Several color look-up tables with interactive control of color mapping parameters are included with SAOimage. Two additional desirable features of look-up tables for astronomical display are the incorporation of rollover, i.e., a repeating sequence of pseudocolors for increasing input value, and control from a user program. Since SAOimage did not fully incorporate these features, it was deemed important to include them in the new version.

1.3 Other Communication Applications

After the present communications interface was written, the author became aware of some other applications besides IRAF which can communicate with SAOimage. The NOAO program Faint Object Classification and Analysis System (FOCAS), which is separate from IRAF, communicates via the IMTOOL protocol to display images and overlays. Also, Jim Wright, at the California France Hawaii Telescope Corp. has developed some preliminary modules which use IMTOOL on the UNIX operating system. These are used for telescope and instrument operation and for designing masks for multi-object spectroscopy.

2. IMPLEMENTATION OF ENHANCEMENTS

The desired enhancements described in section 1 were carried out at three levels. The first level involved the modification and creation of modules within SAOimage itself to carry out the basic display of data and cursors, the feedback of information, and the repair of various bugs. The second level required the creation of new subroutines of a general nature which can be called by a user application to carry out the desired operations. The third level of development was the incorporation of the second-level modules into the MIIPS (Gudehus 1989) package and six demonstration programs. The language of SAOimage is C and that of the second- and third-level programs, FORTRAN, a language familiar to scientists. The enhanced version of SAOimage is known as version 1.07E2 and required the modification and/or creation of 29 first-level SAOimage files. In addition 18 second-level files, excluding the demonstration programs and MIIPS modules, were created. Development was first undertaken for the OpenVMS operating system and about a year later the package was exported to the UNIX operating system. Overall, development under OpenVMS was found to be considerably easier.

2.1. Communication

Under OpenVMS, interprocess communication is carried out by means of Mailboxes. A user program obtains separate channels for sending and receiving information to and from SAOimage, and streams of bytes are then sent on those channels. Under UNIX, the equivalent terms are Pipes and Sockets, the latter being used for communication between nodes. Because the receiving process expects a fixed number of bytes, that number must be set on the previous transmission, with the initial transmission number being 16 bytes, by default. Because of SAOimage's historical relationship with IRAF, a number of modules already existed to encode and decode the byte streams. The convention employed was originally designed for communication with an Integrated Imaging Systems IIS M70 Image Display, and is based on IRAF's IMTOOL protocol. While this protocol possesses some extra overhead because of its many bit masks and bit sum, it was thought easier to adapt it for use with the enhanced version than rewrite a large amount of code. Because there is no formal documentation on these protocols, it was necessary to examine the btye stream in conjunction with the associated code under various conditions in order to understand how it worked.

There are two new second-level routines associated with communication. SAOSTART establishes the size of the display buffer, either via a user prompt or by the user predefining its size, and acquires the communication channels (when operating under OpenVMS). SAOFIRST establishes the zero points of the frame coordinate system and erases the screen of all data and overlays. The helper subroutines IMTOOLCHECKSUM and SAOERAS calculate the checksum of the IMTOOL header packet, and erase the screen, respectively. Many of the second-level modules communicate information to each other through the COMMON block SAOFACTS, which includes the input and output channel numbers, the buffer and its identifying code, and the maximum number of columns and rows that can fit into the buffer. The labor of coding the declarations for SAOFACTS can be avoided by simply including the file SAOFACTS.FOR in the user program, if so desired.

2.2. Display of Multiple Images

SAOimage in its original form did not easily support loading multiple images. This is because several lines of data, which are the full width of the buffer, were required to be loaded into the frame buffer at once. In this situation, the byte stream writes over all buffer locations between the starting and ending pixel locations. That is, any images on the same horizontal level as the newly loaded image are overwritten. Writing several lines at once, as in a block mode of transfer, is highly efficient and quickly loads an image. However, to load multiple images of arbitrary size at arbitrary locations in the display buffer requires that one line, only as wide as the image, be loaded at a time. Thus, versatility in this area is gained at the expense of speed. Besides the required changes in SAOimage, a new second-level subroutine, SAOLOAD, was created to allow easy loading of images, either in the original high speed block mode or as single arbitrary-length lines, depending on the setting of the MODE variable. The position of each image is specified by the frame coordinates of the lower left corner of that individual image. An additional second-level subroutine, SAODELS, is similar to SAOLOAD, but opens a Standard Astronomical Data (SAD) format file, the native format of the MIIPS package.

One minor software bug, revealed when the multiple image capability was added, was that if previous images were being displayed at nonunity zoom, the new additional image would be displayed at unity zoom along with the original nonunity zoom images. This problem was solved by simply redrawing the screen at unity zoom before loading a new image, an approach which also speeded up the loading process.

The standard engineering convention for frame and screen coordinates is to reference the beginning pixel's corner as (0.0, 0.0). For file coordinates this convention is sometimes also used, but for scientific usage the corner is usually referenced as (1.0, 1.0). This is the SAD format convention. Since IRAF references the file corner as (0.5, 0.5), it is necessary to keep these conventions clear. SAOimage originally employed the switches "zero" and "one" to distinguish the engineering and IRAF conventions. In the enhanced version, the engineering, IRAF, and scientific conventions are instead set by the more intuitive switch values of zero, half, and one, respectively.

2.3. Cursors

To allow remote control drawing of cursors, the communications protocol was modified to recognize a new control value which when given, would call up new SAOimage subroutines to carry out cursor drawing. The second-level subroutine SAO_DRAW can by this method be called to draw an erasable cursor, a nonerasable cursor, or an erasable cursor region (set of cursors of the same size and color). A cursor is considered nonerasable if it is written into the buffer containing the displayed data. It can only be removed by reloading the image. In addition to drawing cursors, SAO_DRAW can erase any of the erasable cursors. Because these cursors are independent of the display buffer, loading a new image will not erase them. Thus the module SAO_ERASE_CURSOR, called by SAO_DRAW, was created to handle this task. The cursors are available in the shapes of dots, circles, ellipses, rectangles, and arrows, as single shapes or annuli, and in at least 6 colors. Vectors, useful for graphics and polygons, can be drawn by calling either SAO_DRAW or the specialized subroutine SAO_VECT. Since vectors and nonerasable cursors are drawn into the frame buffer, the number of available colors is potentially large and will depend on the value written and the the current color look-up table. The desired locations are given in the call as file coordinates, but one must also specify which picture in the frame is to be written to since multiple images are supported.

The inverse operation of reading back coordinates at the mouse pointer's location is handled by the second-level subroutine SAO_CURN. Desktop, frame, and file coordinates are returned. Since the picture boundaries relative to the frame are stored in a COMMON block, the number of the picture and the file name at the mouse pointer location are easily determined and the intensity value from the file can then be read.

2.4. Look-up Tables

SAOimage has several built-in LUTs. These are read from a definition file which is easily edited if new tables are desired. Besides incorporating the ability to select the LUT from a remote process, 3 new LUTs with roll-over were created for the enchanced version. The wide dynamic range available from CCD images and the large range of surface brightness possessed by galaxies and other astronomical objects makes the rolling over of pseudocolors mapped to pixel levels a necessity for revealing subtle level changes in an image. A new black-and-white LUT has a sawtooth mapping function with 7 cycles, and two color LUTs have triangle mapping functions with 7 cycles. For example, in one of the color LUTs, magenta, blue, cyan, green, yellow, and red are each represented by triangle functions, with each color offset from the adjacent one by one-half the width of the base of the triangle. Because there are 7 cycles, each color appears 7 times in the LUT. Any of the 11 available LUTs can be made active by specifying the LUT index and calling the subroutine SAOCONVLT. A user-defined LUT can be created in real time by specifying the desired red, green, and blue 8-bit values and downloading the information to SAOimage with the subroutine SAO_LOAD_VLT.

3. DEMONSTRATION PROGRAMS

Six demonstration programs have been written to showcase some of the capabilities of the communications interface. The first, SAODEMO, begins by writing several hundred colored dots onto the display at random locations with random colors. Then short moving trains of dots of variable length are allowed to randomly travel over the display. This is followed by erasable elliptical cursors, erasable elliptical annuli, circle regions, nonerasable ellipses, erasable rectangles, rectangle regions, and nonerasable rectangles. Then 4 simple images are loaded into the display at 4 different locations, and the LUT is cycled through its range.

The second program, SAODEMO2, simulates the focusing of a star. An artificial star image of varying width and central intensity is sent to SAOimage in 16 successive frames and cycled 6 times. This program makes clear that SAOimage can be used in conjuction with data acquisition hardware to allow real time visualization of phenomena which can be captured in a two-dimensional format. The proper focusing of a star image on a CCD, for example, could be done in an effective way by capturing each frame or subframe of the CCD, sending the frame to SAOimage, calculating the full-width half-maximum (FWHM), and displaying a plot of a cut through the image in two perpendicular directions, either on xterm or on SAOimage itself. SAOimage has been incorporated into the MIIPS plot package (Gudehus 1988) hence the capability of combining display and plotting on one device for such an application already exists.

The third program, SAODEMO3, is a simulation of a revolving binary system. An artificial binary system with 20 frames per period is cycled 8 times. In this simulation the display buffer size is 90 by 90 pixels. Because SAOimage's transmission buffer size is 8192 bytes, the entire frame can be loaded in one block. On a VAXstation 4000 VLC, a frame rate of about 9.4 frames per second was attained with this example. Thus, effective visualizations of time-variable astronomical observations and astrophysical simulations can be done with SAOimage. While larger areas could be loaded in one block if the transmission buffer size were increased, this has not yet been attempted.

The fourth program, SAODEMO4, demonstrates the readback of independent cursor positions from each of 4 images. Upon pressing the carriage return, the screen, frame, and file coordinates are displayed, along with the image name and intensity value. The demo ends with an expanding rotating square.

The program SAODEMO5 demonstrates vector graphics. A set of randomly colored and oriented vectors is written to the display. The user is given a choice of displaying each vector as soon as it is sent or after all have been sent. The latter alternative is considerably faster.

The last program, SAODEMO6, combines cursor positioning and vector draws. Each time the user presses the carriage return, a vector is drawn to the current mouse pointer location. Such a sequence can serve as the basis for delineating polygonal areas to be edited or highlighted.

In Figure 1, we give a short sample of code which illustrates the ease with which a single image can be loaded into SAOimage. Following the INCLUDE statement, which declares the variables and arrays used by the various subroutines, the display buffer size is set at 512 by 512 pixels. For simplicity, this exactly matches the declared size of the array, but is in general allowed to be larger. The subroutines SAOSTART and SAOFIRST initialize SAOimage for operations such as image loading or cursor drawing and feedback. Finally, SAOLOAD is called to load the filled array.

```
      PROGRAM SAMPLE
C
C       This sample program demonstrates loading an array
C   of data into SAOimage.
C
      INCLUDE 'SAOFACTS.FOR'
      CHARACTER40 ARRAY_NAME
      DIMENSION ARRAY(512,512)
C
      IFB=1 !  Calls up the 512X512 buffer;  no user prompting
      CALL SAOSTART
      CALL SAOFIRST
C   Fill the array
      NCOLS=512
      NROWS=512
      DO J=1,NROWS
         DO I=1,NCOLS
            ARRAY(I,J)=I+J
         END DO
      END DO
C   Position the lower left corner of the array at the
C   lower left corner of the frame.
      IXFRAME=0
      IYFRAME=511
      BRIGHTNESS_MIN=1
      BRIGHTNESS_MAX=1024
      ARRAY_NAME='Sample_image'
      NPICTURE=1
C   Load SAOimage in 'fast mode'
      MODE=0
      CALL SAOLOAD(ARRAY,NCOLS,NROWS,IXFRAME,IYFRAME,
      1BRIGHTNESS_MIN,BRIGHTNESS_MAX,ARRAY_NAME,NPICTURE,MODE)
C
      END
```

Fig. 1

A sample program which demonstrates the simplicity of displaying an image with the enhanced version of SAOimage and its associated modules.

4. SUMMARY

The set of subroutines described here together with the enhanced version of SAOimage, v. 1.07E2, greatly increase the usefulness of an already powerful image display program. It is now possible to easily display bitmapped images of any size in pseudocolor from a user's program. Furthermore, multiple images of any size may be displayed in the frame at one time, with the mouse pointer correctly returning the screen, frame, and file coordinates of pixels in any image. Several types of cursors useful in astronomical image processing may be written to the display, and look-up tables loaded, all under control of a user's communicating program. When only one image at a time needs to be displayed, the rate of loading is fast enough to allow slow video displays of time-variable phenomena.

All of the source code for the sample programs, the code for compiling and linking user applications, the source code for the enhanced SAOimage, as well as the executables, are available by anonymous ftp at vms.ucc.okstate.edu. The SAOimage files are in the directory DUA4:[MIIPS.X_WINDOWS.SAOIMAGE], the OpenVMS communications files are in [MIIPS.DISPLAY], and the UNIX communications files are in [MIIPS.UNIX.DISPLAY]. Interested parties should first read the files [MIIPS.DISPLAY]SAO_INSTRUCTIONS.TXT (for OpenVMS) and [MIIPS.UNIX.DISPLAY]SAO_INSTRUCTIONS.TXT (for UNIX).

The author would like to express his deep appreciation to Mike VanHilst for his many helpful and encouraging suggestions during the lengthy development of this project and his thanks to the referee Bob Abraham for his useful feedback and suggestions. The author also thanks Oklahoma State University for making their anonymous ftp server available.

Note: Since this paper was written, the ftp server at Oklahoma State University has ceased to be available. The software can instead be obtained by going to http://www.chara.gsu.edu/~gudehus/miips.html and clicking on the link to "Enhanced" SAOimage and the Communications Package.

References

Gudehus, D., 1988, BAAS, 20, 708
Gudehus, D., 1989, BAAS, 21, 784
VanHilst, M., 1991, *User Manual for SAOimage*, Smithsonian Astrophysical Observatory: Cambridge
Worrall, D. M. et al., 1990, BAAS 22, 821