Astronomical Data Analysis Software and Systems IV
ASP Conference Series, Vol. 77, 1995
R. A. Shaw, H. E. Payne, and J. J. E. Hayes, eds.

IMPORT/EXPORT: Image Conversion Tools for IRAF

M. Fitzpatrick

IRAF Group, NOAO[1], *PO Box 26732, Tucson, AZ 85726*

Abstract. We present a set of IRAF[2] tasks for converting foreign image formats to and from the native IRAF format. Traditionally, this has been done by writing individual one-to-one conversion utilities, either as single tasks or as procedures within a multiple-format conversion tool. This approach often results in the unnecessary duplication of code for formats that are actually very similar.

1. Introduction

A survey of the various image formats in use today was done to find the factors common to most formats. It was then possible to write a description of each format into a database, using parameters that describe the layout of the pixels, and expressions that determine the values of these parameters on an image by image basis. Using this database allows one piece of code to read multiple formats. In cases where a format does not fit the general raster model but should be supported because it is particularly useful or popular, or where more detailed processing is required, specialized code can be written. The database approach also means that a new image format can usually be added *by the user* without modifying the existing code.

When exporting IRAF images to other formats the only built-in knowledge of the format is the structure of the image header. Since the database only uses a fraction of what may be in an image header, it cannot be used to define completely the output file. Still, the code for writing the image data is general in nature, using format parameters set internally. This usually results in only a page or two of specialized code for a specific format, most of which is for writing the header. A user-defined file of header information can be prepended automatically. Since the task parameters define the pixel layout fully, there is some minimal capability for producing formats not fully integrated into the task. Table 1 shows the list of currently supported formats.

At the heart of the conversion tasks is the vector expression evaluator. In *IMPORT* this is not only used for computing the converted pixels, but also

[1] National Optical Astronomy Observatories, operated by the Association of Universities for Research in Astronomy, Inc. (AURA) under cooperative agreement with the National Science Foundation

[2] Image Reduction and Analysis Facility, distributed by the National Optical Astronomy Observatories

Table 1. Currently supported or recognized image formats.

Image Format	IMPORT Task	EXPORT Task
Raw Binary	supported generic model	supported generic model
ASCII Text		all task options
AVS X Image	24-bit RGB plus alpha channel	...
Encapsulated PostScript	...	8-bit grayscale, 24-bit color
EXPORT task raw binary	all types	...
FITS	simple images only	...
GIF	8-bit CLT	8-bit CLT
IRAF OIF image	all IRAF data types	all data types
PDS	8-bit PDS3 images only	...
PGM Format	raw file only	raw file only
PPM Format	24-bit raw file only	24-bit raw file only
SGI RGB image	8-bit no CLT, 24-bit RGB	8-bit w/opt CLT, 24-bit RGB
Sun Rasterfile	8-bit w/opt CLT, RGB/RGBA	8-bit w/opt CLT, RGB/RGBA
Sun Taac	8-bit no CLT or 24-bit RGB	...
JPL Vicar	byte, short, int or floating point	...
X10 Window Dump	8-bit w/opt CLT	...
X11 Window Dump	8-bit w/opt CLT, RGB/RGBA	8-bit w/opt CLT, RGB/RGBA
CMU WM Raster	recognition only	...
Fuzzy Bitmap	recognition only	...
JPEG	recognition only	...
NCSA HDF	recognition only	...
Utah RLE Toolkit	recognition only	...
TIFF	recognition only	...

to evaluate the database expressions that define the image. A wide variety of standard math functions is available, as well as specialized functions for reading header values, combining or separating image colors, or defining the layout of images. Expression operands may include user-defined image *tags* that refer to a line of pixels in the image, image header parameters, or scalar values. Operators permit concatenation and replication of pixels, and boolean operations used to select pixels, as well as the standard arithmetic operators.

2. The IMPORT Task

The *IMPORT* task can also be used simply to print information about a file, or list the pixels. By default, the database of formats will be used to identify the file before processing; task parameters can be used to override certain database values or to define the format completely. When a text listing or image is the requested output type, an expression parameter can be used to process the pixels before writing the result. Expressions typically convert RGB values to grayscale, apply a gamma correction, flip an image, or reverse the channel order.

2.1. The Generic Data Model

With a few exceptions, the following model features apply well to a large number of commonly used image formats: (1) pixels are even multiples of 8 bits, short integers may be signed or unsigned, floating point may be native or IEEE, (2) any padding is also a multiple of 8 bits, (3) no compression is used in storing the data, (4) multiple bands of pixels may either be stored sequentially, interleaved line-by-line, or pixel-by-pixel, (5) padding may come before or after the image, before or after each line in the data, or before or after groups of pixels, (6) byte-order of integers may be defined, and may require byte-swapping on a particular host, and (7) pixels are not required to be all the same type.

The task can read any format meeting the above guidelines without changing the code. New formats may be added by appending their own format database or by specifying the task parameters directly.

2.2. The Format Database

The format database is composed of records of *keyword* = *expression* pairs. Expressions often use the specialized I/O functions to read the image header at a specific offset, e.g., to get a magic number that identifies the format or to get an image dimension at a known byte offset. The records are examined sequentially until an expression that identifies the format is evaluated as true. Each expression in the record is then evaluated to set the keyword value, the keywords mirror the task parameter and define, for instance, the amount of header to skip before reading pixels. The expressions are evaluated for each image individually, so there is no requirement that each image in a list be the same size or, indeed, the same format.

3. The EXPORT Task

The *EXPORT* task can easily convert an IRAF image to any of the supported formats. Again, the generic data model is used as a basis for the image conversion code, but there is additional support for formats that meet special needs. Aside from raw binary data, the task can also write the pixels as ASCII text, for input to other programs or for debugging. Support for Encapsulated PostScript allows for direct conversion of images for hardcopy output or for inclusion in documents (such as this one). GIF output allows for data to be processed by common X utilities or to be presented on the WWW. Lastly, images may be written out as new IRAF images if the user wants to combine several images into a single

Figure 1. An example mosaic produced by the EXPORT task showing three BVR images of the Trifid Nebula. The images were converted directly from IRAF format to Encapsulated PostScript using the EXPORT task.

file for further processing (e.g., annotating the image using the *TVMARK* task) before writing it out in some final format.

Where possible, the output may be color images—something not supported directly by IRAF image model itself. By combining three images which may represent the three color bands, an output file may be written as an RGB color image. Since formats such as GIF do not support RGB data, an expression function is provided that will automatically compute an 8-bit colormap from three images using a Median Cut Algorithm and Floyd-Steinberg dithering. This colormap can be output to any file that supports it. When only one image is available, but color output is desired, an artificial colormap can be specified. There are currently eleven colormap selections built in to the task; a user-specified file may also be used.

The *EXPORT* task also provides several ways of scaling the dynamic range of the IRAF image to the allowed range of the output format. These include: (1) the *zscale* algorithm used by the *DISPLAY* task with default values to give an optimal scaling, (2) user-specified *z1* and *z2* values, (3) colormap scaling using display brightness and contrast values, (4) a linear transformation function, and (5) any arithmetic expression or function (e.g., sqrt(), log()). Automatic datatype conversion will be done unless something specific is requested.

Perhaps the most useful feature of the task is the ability to mosaic any number of input images into a single output image. Expression functions can be used to place images side-by-side or from top-to-bottom. In Figure 1, for example, three IRAF images were written directly to the EPS file included in this paper. Each image was intensity scaled independently. The expression syntax specified the image layout as well as the spacing between the images. In such mosaics, images do not have to be the same size; the size of the final image will be determined automatically. Writing an image mosaic as a new IRAF image allows the user, for instance, to mark object stars and output the marked image as PostScript.