Astronomical Data Analysis Software and Systems VI
ASP Conference Series, Vol. 125, 1997
Gareth Hunt and H. E. Payne, eds.

The FITS List Calculator and Bulk Data Processor

Elizabeth B. Stobie and Dyer M. Lytle

NICMOS Project, Steward Observatory, University of Arizona, Tucson, AZ 85721, E-mail: bstobie@as.arizona.edu, dlytle@as.arizona.edu

Abstract. The FITS List Calculator and Bulk Data Processor (FLC) is an IDL program with a graphical user interface developed for processing large groups of Near Infrared Camera and Multi-Object Spectrometer (NICMOS) test data efficiently. The program is built around the simple concept of defining a new IDL variable type, the file list. Lists of files may be assembled with assigned variable names and displayed with subwidgets. These variables may then be arguments to predefined functions and procedures in the main widget as well as IDL functions and procedures. They may also be used in arithmetic expressions. Images and arrays may be defined by direct loading or as the result of calculation. When processing is repetitive, scripts may be written to process many groups of data in exactly the same way.

1. Introduction

A total of 126,370 observations were produced during the NICMOS instrument characterization tests in the Brutus thermal vacuum chamber at Ball Aerospace during the summer of 1996. These observations provided performance information as well as preliminary calibration reference data for the instrument. Software was needed to process large datasets in routine ways with minimal effort. FLC was developed for this purpose by the authors in a time period of approximately two months.

2. Data Types

There are four basic data types used in FLC:

1. lists
2. images
3. arrays
4. scalars

The user may customize the number of lists, elements in a list, images, and arrays by editing the user_defs file. Structures are allocated at run time for the maximum number of lists, images, arrays, and scalars. One must be careful of the number of images allocated as they are the largest memory hog. Figure 1 shows the data structures used to hold lists and images.

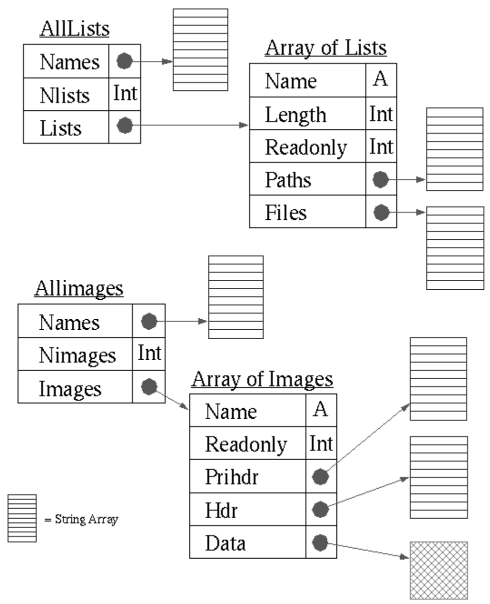

Figure 1. List and Image structures in FLC.

2.1. Lists

Lists contain lists of FITS filenames (no other filetype is supported), which are stored in memory. The files themselves are stored on disk and read in one at a time when needed. List names may be used in FLC functions and procedures or used in arithmetic expressions. The default limits are 100 file lists with 100 filenames in each. They are stored as IDL structures and these structures are, in turn, stored in a parent IDL structure containing all lists.

2.2. Images

Images may be produced as the result of arithmetic operations, functions, or defined procedures or may be directly loaded from FITS files residing on disk. The default size of the image data is 256×256, optimized for NICMOS data. The image header is also loaded into memory. For files with image extensions the primary header and the first image extension header are stored separately. Any subsequent image extensions are ignored. Newly created images may be saved to disk in FITS format. The default number of images is 50. The IDL structures for images are similar to the structures for lists except that the image headers and image data arrays are kept in the structure.

3. Display

There are two display functions in FLC. The first, shown in Figure 2, is for detailed examination of individual images. The window includes a scrolling text region for examining the image headers, a histogram/plot display that can show a histogram of the entire image, a histogram of an individual quadrant, or plots

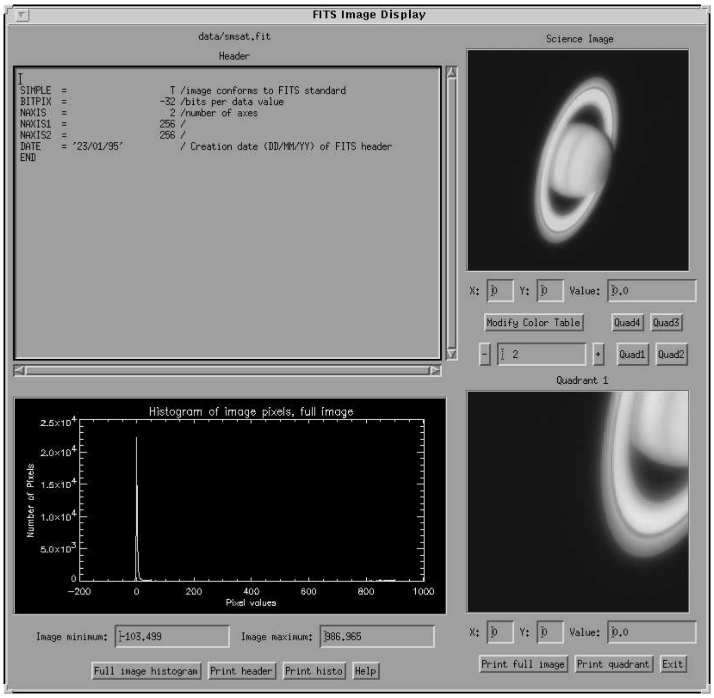

Figure 2. Image display in FLC.

of cross sections of the image, vertical or horizontal. Finally there are two image displays, one showing the entire image and one showing only a quadrant of the image. The image displays also have (x,y) value readouts below them.

The other display function is called the "scanner" and its primary purpose is for quickly scanning through a list of images to look for irregularities in the data sets. It has one image display and forward and reverse scan buttons. Along the bottom of the window, the median and standard deviation of each image is shown.

4. Arithmetic

Standard arithmetic operators can be used in FLC expressions. Operators like plus, minus, multiply, divide, and exponentiation are generally overloaded where overloading makes sense. For example, an image can be added to a list resulting in a new list where each member of the new list is the sum of an image from the original list and the added image. In other situations, where overloading would be ambiguous, those constructs cause errors; for example, trying to add a 1-D array to a list of images.

5. Functions

Several functions for operating on lists, images, and arrays are predefined for the convenience of the user. They include makelist, getlist, savlist, and dellist for creating, saving, and deleting lists. Similarly, getimage, getmask, savimage, and delimage are predefined for images and extract, getrow, getcol, savarray, and delarray for arrays. Other functions defined specifically for lists include averaging images in a given list, computing a median image from the images in a given list, summing the images in a given list, computing image statistics (mean, standard deviation, median, skew, and kurtosis) images across the images in a given list, and computing the same statistics for each image in a given list.

6. Procedures

Several routines do not require a target for storing a result. The user may plot a specified array, display a specified image or element in a list, scan all images in a specified list, fit a linear expression by linear regression to each pixel of the images in a specified list, or execute a script.

7. Scripts

Often many datasets are processed in an identical manner which can require several repetitions of a set of commands. Scripts were implemented to avoid repetitious typing and minimizing the opportunity for mistakes. Any command allowed at the command line may also be used in scripts and comments are supported when prefaced by a semicolon (;). Scripts may call other scripts.

8. Parser

Most commands, excluding scripts and IDL procedures, may be invoked by selecting them from the appropriate widget menu or by typing the commands in the command window. The FLC parser processes commands as they are entered in the command window. All variables, functions, and procedures are identified as well as arithmetic operators. Variables are then replaced with internal variables that are associated with the variable names and the IDL EXECUTE command is used to execute the command. No command may extend beyond one line (80 characters in length) and no conditionals or do loops are allowed.

9. Future Plans

The current version of FLC has only been tested on computers running IDL version 4.0 or later under various flavors of the UNIX operating system. With a little work, it could be made to run on the Apple Macintosh and Intel processor based systems running Windows.

FLC will continue to be used within the NICMOS project during the coming years. If there is a demand, the program could be generalized somewhat to handle a wider variety of data.

Acknowledgments. We are grateful to Tony Ferro for helping test the program.