Instrument Science Report NICMOS ISR-99-004

User’s Guide to Polarimetric Imaging Tools

Lisa Mazzuca, Dean Hines
August 2, 1999

ABSTRACT

We present informal documentation for a group of IDL programs: POLARIZE.PRO, POLAR.PRO, and POLARPLOT.PRO This set of programs produce polarization coefficient images from observed data images taken with NIC1 or NIC2. These images are derived based on various input parameters from the user. From the derived coefficient images, the intensity image is displayed and options to superimpose polarization vectors or contour lines is given.

1. Introduction

POLARIZE.PRO is an interactive IDL program that derives relevant parameters from NICMOS polarization images and plots the resulting intensity image and polarization vectors. The program produces the Stokes parameters, which are then used to calculate the degree of polarization as well as the polarization angle. The program contains the most recent characteristics for each polarizer on NIC1 and NIC2 as of the date of this ISR[1]. This includes the latest values for the polarizer transmission coefficients.

The IDL program reads three images taken with three polarizers from camera 1 or camera 2 on NICMOS and produces five images as output. The output images are’*q.fit*’ and ’*u.fit*’ (these two images represent the Stokes parameters), ’*i.fit*’(the total intensity), ’*p.fit*’ (the degree of polarization), and, ’*theta.fit*’(the polarization angle). The user has the option of using the single polarizer convention (as is the case for NICMOS) or the multiple polarizer convention (identical polarizers in series)[2]. Regardless of which convention is used, the program will display the image *i.fit*. Polarization vectors or contour maps can be superimposed over the intensity image.

The IDL programs *POLAR.PRO,* and *POLARPLOT.PRO* together provide the same capabilities as *POLARIZE.PRO.* That is, the user can use *POLAR.PRO* if the end result is only the production of the image files with no automatic plotting of the intensity image.

Likewise, if the user already has the image files and simply wants to utilize the plotting capabilities of *POLARIZE.PRO*, then only the program *POLARPLOT.PRO* needs to be run.

Currently, the programs can be found under the NICMOS homepage[3]. Please be aware that this program is not a formally configured piece of software, and that no other support mechanisms are in place beyond this manual.

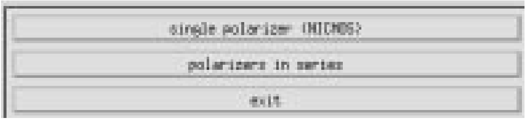
2. Documentation for Polarize.pro

While in an IDL session, the user runs *POLARIZE.PRO* using the following command:

```
IDL> .run polarize

IDL> polarize
```

A widget window is then displayed which shows three buttons:



single polarizer (NICMOS) - uses NICMOS coefficients derived by Dean Hines, et al.[4], for a single polarizer passing light straight through onto the detector

polarizers in series- takes the NICMOS coefficients derived by Hines, et al., and transforms them to be compatible with a pair of identical polarizers in series

exit - exits the main program

The program runs the same for both polarizer conventions with the exception of using different transmission coefficients, which are converted internally during the run. Once clicking either the first or second buttons, the program prompts the user to enter which set of polarizers to use:

1 - POL-S polarizers (polarizers from NIC 1)

2 - POL-L polarizers (polarizers from NIC 2)

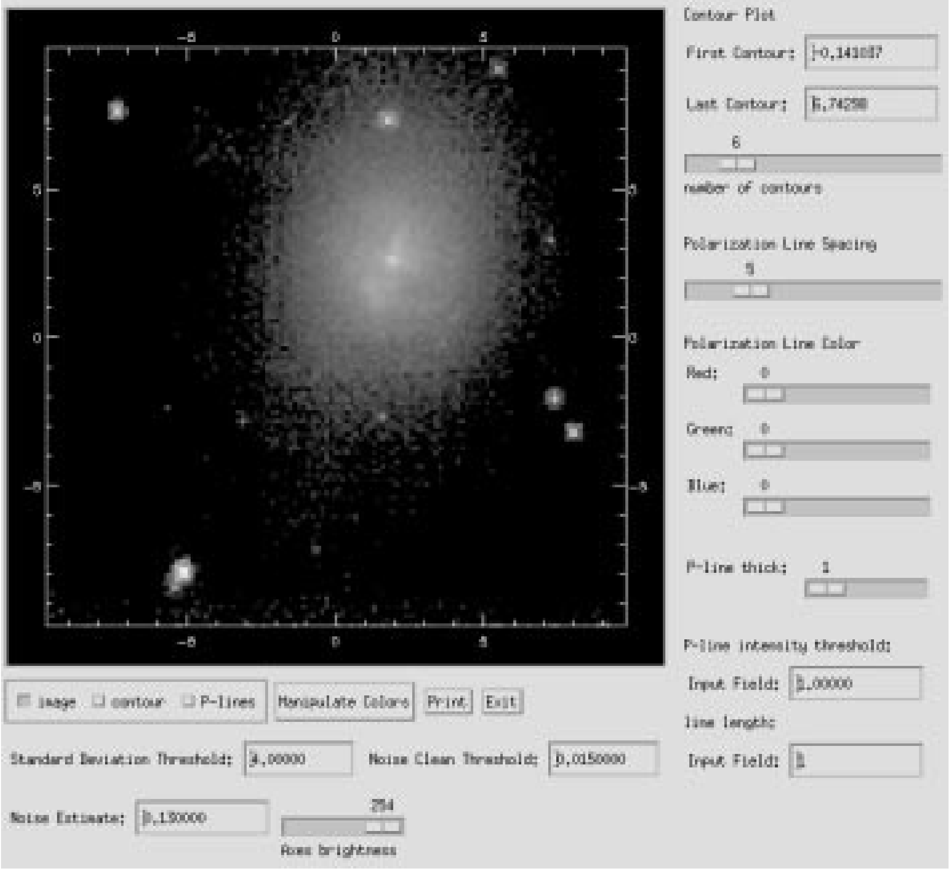
Some user feedback is then given that indicates the following:

- states which polarizer convention is being used
- states which set of polarizers is being used
- states the filters being used
- displays the coefficients for epsilon and phi
- states the fraction of light transmitted in the parallel and perpendicular directions for each polarizer.

- from these values it produces the coefficient matrix that will be used to compute the Stokes vector.

The program then prompts the user for the filenames (enter one at a time) of the three images to be used (e.g., pol0.fits, pol120.fits, pol240.fits). It then produces the inverted coefficients matrix and the output images - *i.fit*, *q.fit*, and *u.fit*, which are written to the current directory. It also produces the images *p.fit* and *theta.fit* in the current directory as well. The *i*, *q*, and *u* image files are the polarization components, *p* is the polarization intensity image, and *theta* is the polarization angle image.

A widget window is then automatically displayed which plots the polarization vectors over an intensity image:



☒ image ☐ contour ☐ P-lines

Standard Deviation Threshold: 3.00000

Noise Clean Threshold: 0.0150000

Noise Estimate: 0.130000

Axes brightness: 254

First Contour: -0.141037

Last Contour: 6.74298

number of contours: 6

Polarization Line Spacing: 5

Red: 0

Green: 0

Blue: 0

P-line thick: 1

P-line intensity threshold:

Input Field: 1.00000

line length:

Input Field: 1

Details of the various buttons and what they do:

- Below the image display there are three buttons for selecting the contents of the that display. The intensity image can be shown as greyscale and/or contour plot and the polarization vectors can be turn on or off.

- The ’*manipulate colors*’ button brings up a widget for selecting and adjusting the color tables. The print button produces a postscript image of the display and sends it to the selected printer.

- The ‘*exit*’ button exits the window widget.
- The ‘*Standard Deviation Threshold*’ field applies any user-defined standard deviation
- The ‘*Noise Clean Threshold*’ field value is used when displaying the intensity image. The user can specify a threshold below which data will be floored.
- The ‘*Noise Estimate*’ field applies any user-defined noise estimate
- The ‘*Axes Brightness*’ slider adjusts the brightness of the axis from black to white

Contour Plot Controls
- ‘*first contour*’ and ‘*last contour*’ fields: set the upper and lower boundaries for the the color used to plot the polarization vectors.
- *‘p-line thickness’* slider adjust the thickness of the polarization vectors from 1 to 4
- ‘*p-line intensity threshold*’ field: This is the threshold in the intensity image below which polarization vectors will not be plotted.
- ‘*p-line intensity threshold line length*’ field that defines the length of the polarization vector

Once the window widget is exited, the user returns to the Reduction Transformation widget window and then can either proceed with another run or exiting the program.

Because images often contain areas where the signal is low and the image is mostly noise, the polarization intensity for these areas of the image are often also noisy and have arbitrary values. When plotting the polarization vectors, the intent is to avoid plotting vectors in these noisy image areas. Therefore the software only plots polarization vectors in regions where:

($pfi/ne > sdt$) and ($ii > it$) and ($pfi < 0.9$), and define:

pfi = polarization fraction image (varies from 0.0 to 1.0) (an input parameter)
ii = intensity image (an input parameter)
ne = noise estimate (read from ‘*Noise Estimate*’ field)
sdt = standard deviation threshold (from ‘*Standard Deviation Threshold*’ field)
it = intensity threshold (from ‘*p-line intensity threshold*’ field)

Sample Output

Below is sample output to the screen from a run with the given known parameters:

- images taken with NIC2
- filters were POL0-L, POL120-L, and POL240-L

```
IDL> .r polarize.pro
IDL> polarize
(user prompted to enter which convention to use - for this example, the button for
the single polarizer convention is pressed)
...using single pol (NICMOS) convention
> type 1 for POL-S or 2 for POL-L: 2
...applying appropriate transmission coefficients for POL-L
epsilon:
     0.731300     0.628800     0.873800
phi:
      8.84000      131.420      248.180
fraction of transmitted light in parallel direction:
     0.898100     0.855100     0.966700
fraction of transmitted light in perpendicular direction:
     0.155200     0.227900     0.0673000
 **resulting coefficient matrix is:
     0.518743     0.361439     0.115211
     0.524989   -0.0411453    -0.327539
     0.515879    -0.326222     0.311091
 >Input filename 1: pol0_mos.fits
 >Input filename 2: pol120_mos.fits
 >Input filename 3: pol240_mos.fits
 ...Using coefficients from camera 2
 Filters are: POL0L            POL120L           POL240L
 **the inverted coefficient matrix is:
      0.599883     0.752170     0.569774
       1.66598    -0.511098     -1.15511
      0.752231     -1.78327      1.05835
 ...producing the files i.fit, q.fit, and u.fit
 ...producing the file p.fit
 ...producing the file theta.fit
WARNING: Calculated values of theta contained values less then 0
These less than zero values were corrected by adding PI.
(the program then opens up a separate window and plots the intensity image; from
this window the user is able to adjust several display parameters as well as super-
impose either a contour or the polarization vectors)
(once the user exits from the plotting window, he decides to exit the main program
by pressing the EXIT button)
until next time...
```

3. Documentation for Polar.pro and Polarplot.pro

Polar.pro and *Polarplot.pro* are basically the components of *Polarize.pro*. That is, *Polarize.pro* contains the functions from both *Polar.pro* and *Polarplot.pro.* Since it is foreseen that the user may only want to create the output images (i.e., Stokes, intensity, and polarization images) without plotting them or simply plot the intensity image and polarization vectors for a given set of input images that were already created, the authors decided to supply independent IDL programs to produce such results.

Polar.pro calculates polarization coefficient images from observed data images taken with either NIC1 or NIC2. The program reads three images taken with the three polarizers and produces five images as output. Unlike *Polarize.pro*, *Polar.pro* does not contain any special user interface controls. The user must input the three filenames on the command line when executing the program:

```
IDL> .run polar

IDL> polar,’image_one.fits’, ‘image_two.fits’, ’image_three.fits’
```

The program will then indicate when the output images are being produced. The output images are ’*i.fit*’, ’*q.fit*’, ’*u.fit*’, ’*p.fit*’, ’*theta.fit*’. These images are written to the current directory. The *i, q*, and *u* files are the polarization components, *p* is the polarization intensity image and *theta* is the polarization angle image. The *i, p,* and *theta* images are used by *polarplot.pro*.

Polarplot.pro is used to plot polarization vectors or contours over a given intensity image. The arguments are the intensity image, *i.fit*’, the polarization direction image, ’*theta.fit*’, and the polarization fraction image, ’*p.fit*’, in that order. Therefore the command would be

```
IDL> .run polarplot

IDL> polarplot,’i.fit’, ‘theta.fit’, ‘p.fit’
```

The program will then produce the same plot display as documented in Section 2. When exiting the plotting image, the user is also exiting the program itself and will have to rerun it for subsequent image plots.

4. Run Requirements

The IDL programs have been tested using IDL versions 5.0 and 5.1. In order for the software to execute, the user must also download the following fits IDL routines (supplied by the University of Arizona):

primary_update.pro *ua_fits_close.pro*

ua_fits_open.pro *ua_fits_read.pro*

ua_fits_write.pro

These routines have also been made available under the NICMOS homepage. They must be placed in the same directory as the polarization programs.

5. References

1.Hines, D.C., Schmidt, G.D., & Lytle, D., “The Polarimetric Capabilities of NICMOS”, in The 1997 HST Calibration Workshop with a New Generation of Instruments, ed. Casertano et al, 1997

2. Mazzuca, Sparks, Axon, NICMOS ISR 98-017, “Methodologies to Calibrating NICMOS Polarimetry Characteristics, 1998.

3. NICMOS Homepage is http://www.stsci.edu/instruments/nicmos/NICMOS_tools/nicmos_tools.html

4. Mazzuca, Hines, *HST Data Handbook Version 3.0* (Baltimore: STScI), chapter 18.4: Polarimetry, 1999.