Summer | 14

Manual for iDealCam v2.0

DAN LI

Department of Astronomy, University of Florida

Copyright and Disclaimer

iDealCam was developed from 2010 to 2014 at the Department of Astronomy, University of Florida.

The use of this software is free of charge. However, please do not distribute this package without prior consent of D. Li. Please refer any interested user to the web site of this software: http://www.astro.ufl.edu/~dli/web/IDEALCAM.html

I reject all responsibility for the use of this package. The package is provided as-is, and we are not responsible for any damage to hardware or software, nor for incorrect results that may result from the software. The user is fully responsible for any results from this code, and I strongly recommend thorough testing of the code before using its results in any scientific papers.

If you use this software, you may want to notify D. Li (d.li@ufl.edu) so that you are put on an email list. This ensures that you are always up to date with major updates.

Table of Contents

I. Introduction **1**
II. Installation and Quick-Start **3**
III. Aperture Photometry **9**
IV. Sky Background **11**
V. FWHM Measurement **12**
VI. Frame Register **13**
VII. Imaging Polarimetry **15**
VIII. Aperture Polarimetry **19**
Appendix I. Use DL_POLOPLOT to Plot Polarization Maps **21**
Appendix II. Known Restrictions and Bugs **25**

I. Introduction

iDealCam is an IDL GUI application developed for processing raw FITS data produced by CanariCam (CC), one of the first-light instruments of GTC. It provides astronomers with an efficient tool to view, reduce, and analyze multi-extension FITS files. Although iDealCam is optimized for CC imaging and polarimetry data, it can also work with other types of data produced by CanariCam, T-ReCS (facility mid-IR camera of Gemini South), and Michelle (facility mid-IR camera of Gemini North).

Some background knowledge about mid-IR observations and data structure

In order to get rid of the strong (and fast varying) thermal background coming from the atmosphere and the telescope itself (a good way to think about this is to imagine that you are observing a star with naked eyes during daytime), nearly all state-of-the-art mid-IR cameras operate with a special technique called "chop and nod" (*Gemini* website has a very nice and detailed explanation of it). Meanwhile, due to the strong background and finite well depth of the detector, the mid-IR arrays need to be read out with a very short frame time (typically around 10 ms). Consequently, a raw mid-IR FITS files is in fact more like a short movie rather than a single image, consisting of a large number of chop and nod frames. To see the final product of the observation (i.e., the science image of your selected object), those frames need to be properly combined. The FITS structure might be complex, and different cameras use different formats to save the chop/nod frames in their raw FITS files.

In this context, iDealCam is a very handy app to check those mid-IR "movies", and to stack frames. Compared with other existing IRAF or IDL packages and procedures, using iDealCam requires little programming skill. You do not have to be an expert of the mid-IR data structure either. Once the program is installed and compiled, everything can be done interactively within the graphical user interface.

iDealCam is not merely a FITS viewer. It also provides a variety of analyzing tools, such as aperture photometry, signal-to-noise

measurement, etc. As a polarimetry reduction tool, iDealCam is able to calculate the Stokes parameters and visualize the polarizations.

Current release of iDealCam (v2.0) does NOT have the capability to reduce CC spectroscopic data. However, one can still use it to check and stack the mid-IR spectroscopic FITS files.

Features and Capabilities (v2.0)

- Graphical user interface
- Compatible with multiple instruments and data types:
  - CanariCam – imaging, spectroscopy, and polarimetry
  - T-ReCS – imaging and spectroscopy
  - Michelle – imaging and spectroscopy
- Frame registration
- Quick and easy access to each chop or nod frame
- FITS header keywords
- Bad frame(s) removal
- Save a single frame into a new FITS file with a proper header
- Display data with different color scheme and scale (similar to DS9)
- Aperture photometry and profile fitting
- Sky background, noise level, and Signal-to-noise measurements
- Image centroid, FWHM, and jitter
- Built-in database of Cohen standards and in-band fluxes
- Polarimetry reduction and instrumental polarization correction

To Be Added in the Future Release

- Smoothing and filtering
- Sensitivity and Strehl ratio measurements, and more

Referencing the use of iDealCam

- If you choose iDealCam to reduce CC data, please reference the paper:
  Li et al. 2012 (BiBTex entry at ADS)

II. Installation and Quick-Start

Download

In order to run iDealCam on a local machine, one needs to download three packages. The first package is iDealCam itself:

http://www.astro.ufl.edu/~d.li/web/IDEALCAM_files/iDealCam_v2.0.zip

It contains three files:
`iDealCam.pro` – iDealCam source code (ASCII file, can be opened in IDLDE, or by any other text editor).
`CanariCam_zeropoint_Jy.dat` – an auxiliary data file for calculating in-band fluxes (called by iDealCam.pro).
`dl_poloplot.pro` – an optional IDL routine for plotting polarization maps.

The other two packages are IDL libraries called by iDealCam:

IDL Astronomy Users Library

http://idlastro.gsfc.nasa.gov/ftp/astron.tar.gz

Frank Varosi's IDL Library

http://www.astro.ufl.edu/~varosi/mosaic/vlib.tar.gz
or
http://www.astro.ufl.edu/~d.li/web/IDEALCAM_files/vlib.tar.gz

System Requirements

iDealCam is developed under IDL 8.0 and Mac OS X Version 10.6, and has been tested under IDL 8.2 (Mac OS/Linux). However, I found that iDealCam is not compatible with IDL 8.1 (maybe due to a bug in the IDL built-in routine CW_BGROUP in v8.1). So, please have your IDL upgraded to 8.2 or a higher version to run iDealCam if you are still using IDL 8.1.

During development, some attention has been paid to the compatibility with MS Windows OS. But due to the lack of a proper testing environment, this compatibility is not guaranteed at this moment.

Installation

Unzip all three packages. You can either put everything in a folder that has already in the IDL_PATH, or a new folder that will be added into the IDL_PATH.

Example

Assuming that you have unzipped and saved everything in a folder, called idealcam, under your home directory, /astro/homes/xxx, then you should edit your ~/.cshrc (for C shell users) file and add the following line:

```
setenv IDL_PATH +/astro/homes/xxx/idealcam:$IDL_PATH
```

If there is no IDL_PATH defined in the .cshrc file, then include the line:

```
setenv IDL_PATH +/astro/homes/xxx/idealcam
```

The plus sign (+) that immediately precedes /astro/homes/xxx tells IDL that it should also search sub-directories of the folder named idealcam for additional .pro files.

~~You should also edit you IDL startup file and add a line to define a system variable called !DEBUG:~~

```
defsysv, "!DEBUG", 0
```

This system variable is required by CENTROID, a routine in Frank Varosi's library and frequently called by iDealCam. If you do not have an IDL startup file, you can download an example from my website:

http://www.astro.ufl.edu/~d.li/web/IDEALCAM_files/idl_constants_startup.pro

To tell IDL where your startup file is, add a line to your .cshrc

```
setenv IDL_STARTUP /astro/homes/xxx/idealcam/idl_constants_startup.pro
```

Note that an alternative way to setup IDL_PATH and IDL_STARTUP is to use the menu in IDLDE (the graphical interface of IDL). Just select:

Menu → Preferences → IDL → Startup file

And

Menu → Preferences → IDL → Paths → Insert…

Do not forget the checkmark to include sub-directories.

Quick-Start

In IDL (or IDLDE), type in the following command to compile iDealCam before your first use:

```
IDL> .compile idealcam
```

If everything was done right during the installation, IDL should be able to find the source code of iDealCam and to get a bunch of modules compiled successfully without any error or warning messages. Next, type in:

```
IDL> idealcam
```

And the main interface of iDealCam should appear.

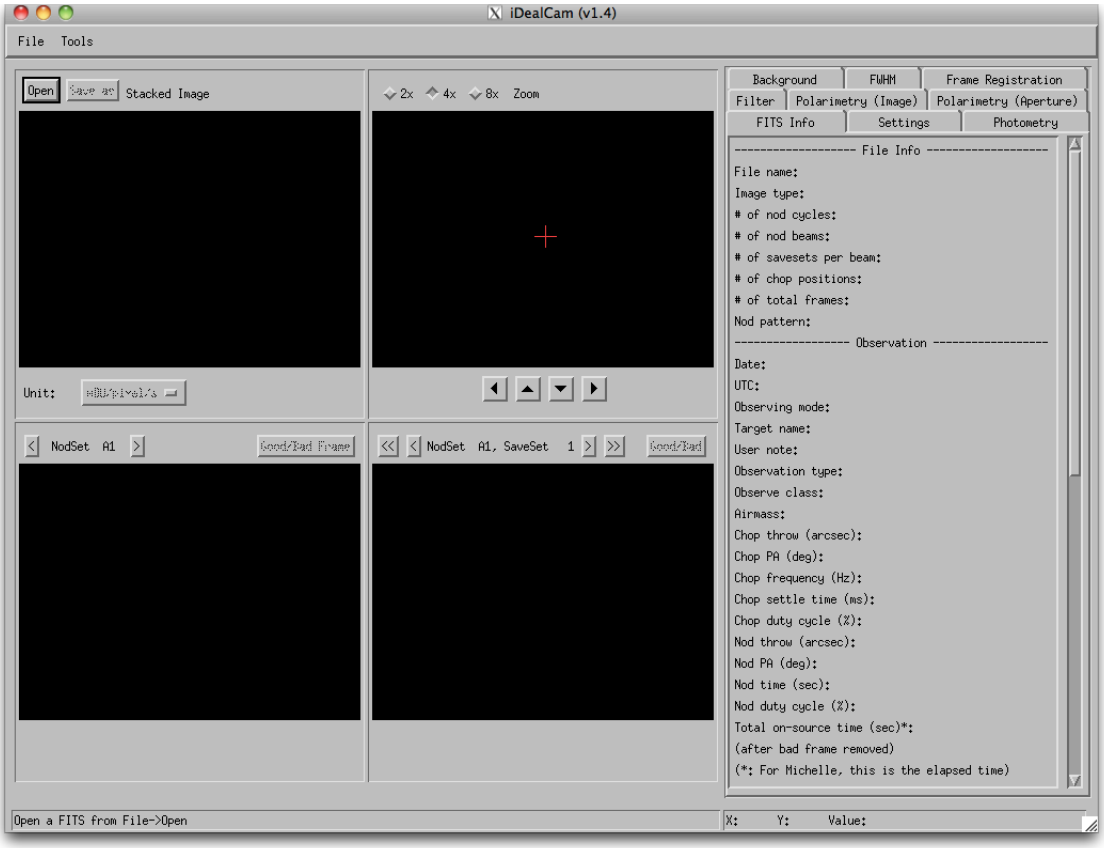
Compiling Errors

If you see the following error message during compilation:

```
% Error opening file. File: idealcam
```

Please check your IDL_PATH and make sure that the idealcam directory is in it. For any other compilation error/warning, please report the entire error/waning message, along with a description of your operation system and IDL version, to me and I will try to find out a solution for you.

The Main Interface of iDealCam



The GUI of iDealCam has two major parts: the left part contains four screens for data display; the right part provides access to the reduction and analysis tools.

Once a raw FITS file is opened in iDealCam, the stacked (without frame registration), chop-corrected image will be displayed in the upper-left screen. The first ***nod frame***[1], and the first ***save set*** of the first nod frame, will be displayed in the bottom-left and bottom-right screens, respectively. Meanwhile, a collection of FITS Keywords and their values will be displayed in the right side of the GUI. The upper-right screen is used for zoom-in display.

[1] In a CanariCam FITS file, each extension is corresponding to a nod position (A or B).

Menus and Buttons

[Open] Open a FITS file. The program will do the chop-correction accordingly based on the data type (imaging or polarimetry) and instrument (CC, T-ReCS, or Michelle).

[Save as] Save the stacked image as a new FITS file.

[Unit] Switch between ADU or ADU/pixel/second (i.e., ADU divided by the on-source integration time) when display the pixel value in the status bar.

[<] [>] Go to the previous (or next) nod frame or save set.

[<<] [>>] Similar to the buttons above, but jump to the same save set of the previous (or next) nod frame.

[Good/Bad] Set the current nod frame or save set as a good (or bad) frame. The stacked frame will be updated immediately. Note that if the current stacked image is a result of frame registration, then after setting a new good or bad frame, you need to perform the frame registration again to update the stacked image.

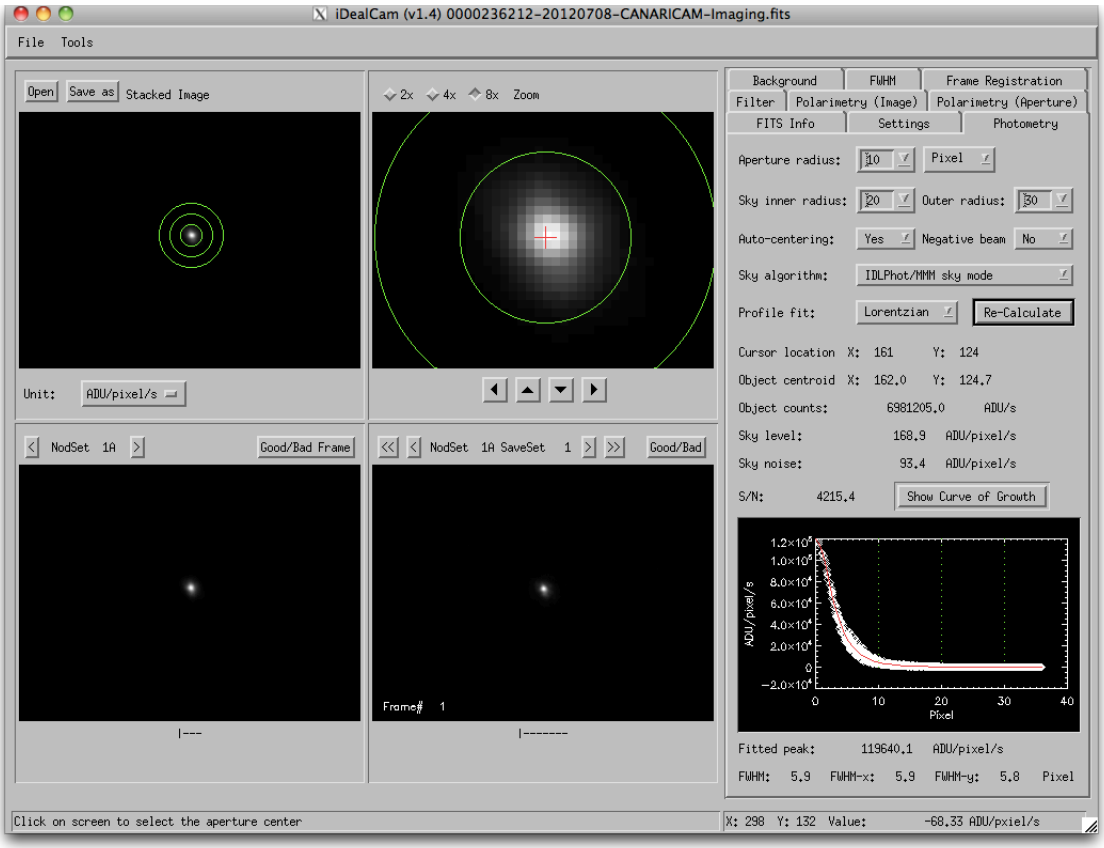
[2x/4x/8x zoom] Magnification for the zoom-in display.

File – Save All Save each save set into an individual FITS file. Each new file will have its own FITS header. This is useful if someone want to reduce a CC-Polarimetry file with POLPACK.

Tools – Standard Stars
Search in a build-in database for in-band fluxes of a chosen Cohen standards.

III. Aperture Photometry

To perform aperture photometry on the stacked image, click on the tab *Photometry*, and then click on a point source on the screen.



You can define the radius of the aperture, and the inner/outer radii of the sky (in unit of pixel or arcsec). By default, the program will center the aperture on the centroid of the source, but you can turn this feature off by selecting "No" in **Autocentering** (if the source is very faint, then autocentering may fail). If the source is in a negative beam, then you must select "Yes" in **Negative beam** (default value is "No"). Likewise, you can choose whether to fit the stellar profile by a Lorentzian function (**Profile Fit**) to measure the FWHM and peal value of the profile.

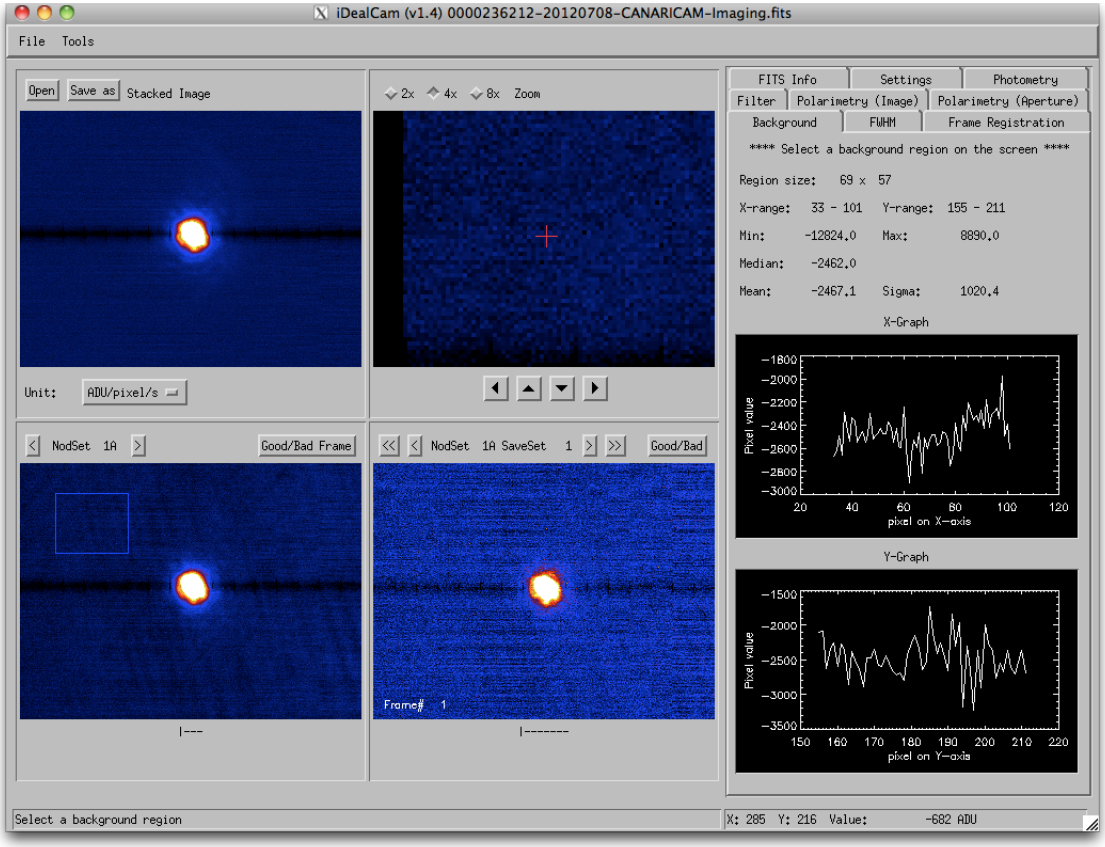
After clicking on the source of your choice, the aperture and sky area will be indicated by green circles on the screen. Meanwhile, a plot of pixel

value vs. radius will be drawn. If you select to do the profile fit, then the best-fit Lorentzian profile will be plotted in red. The plot can be switched between the stellar profile and the curve of growth (**Show Curve of Growth**).

A known restriction in aperture photometry

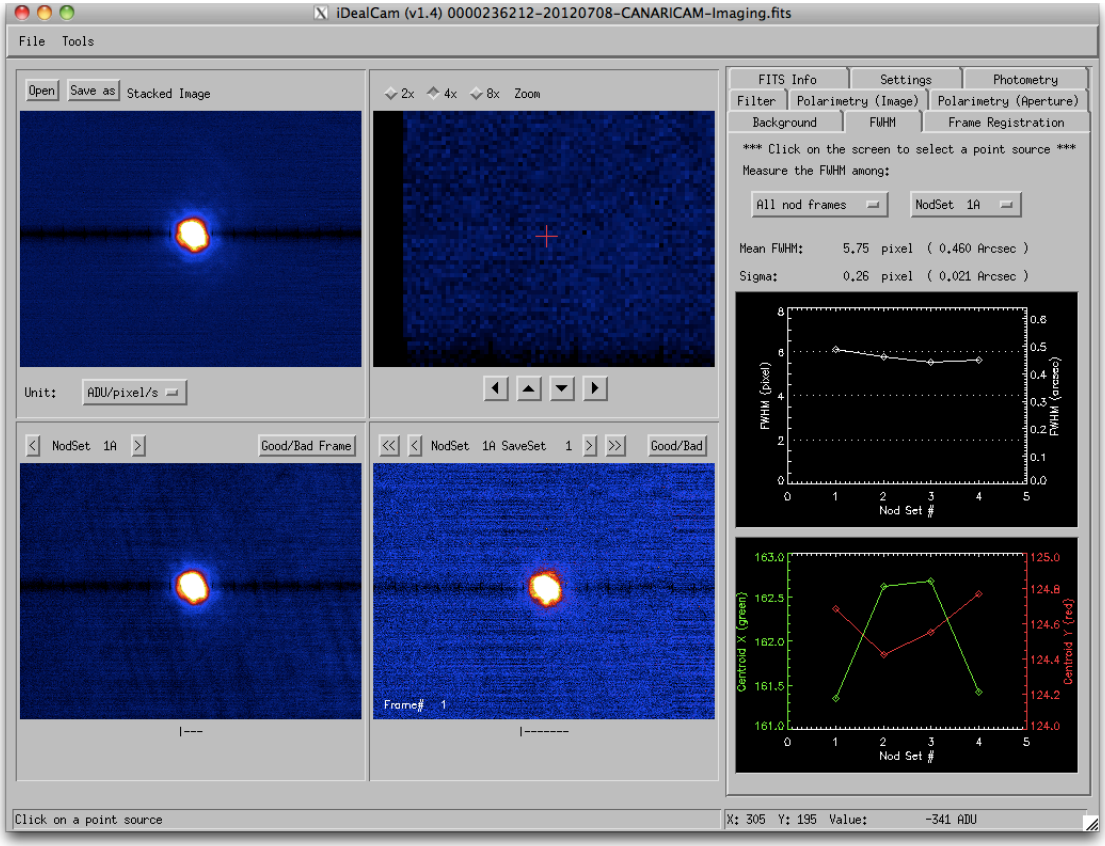
When calculating the in-aperture fluxes, a routine called APER from the IDL Astrolib is used. We note that the maximum sky noise that APER can accept is 999.9 (in whatever unit the data matrix has). In most cases, this is safe. However, if one chooses to display the data in unit of ADU (instead of ADU/s) in iDealCam, then it is possible that the pixel values in the stacked image are extremely high, and the true sky noise is several times greater than 999.99 ADU per pixel. In this case, the sky noise will be underestimated. To avoid this problem, switch the display unit to ADU/pixel/s.

IV. Sky Background



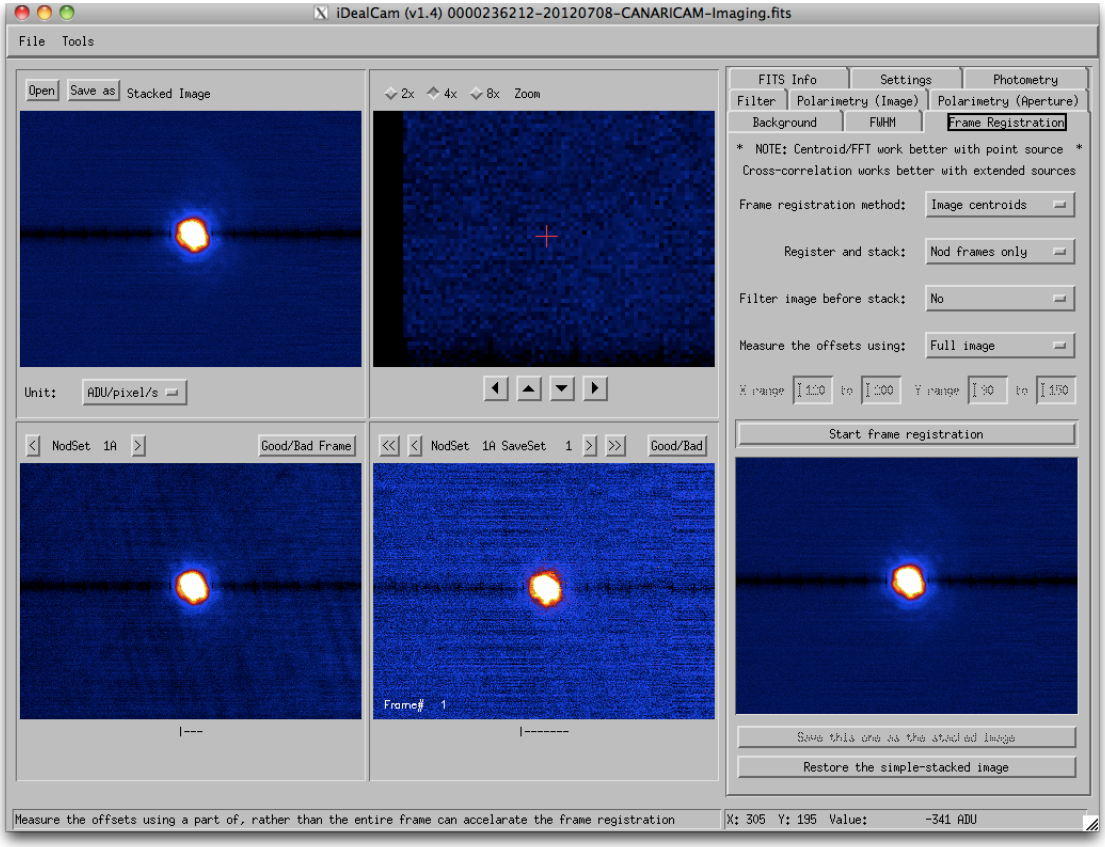
To Measure the sky background, switch to the tab of **Background**, and then select a region in either a stacked image, a nod frame, or a save set.

V. FWHM Measurement



In this tab, you can measure the FWHM of a source, not only in a single frame or a stacked image, but also in many frames at the same time. Just select the desired data set (all nod frames, all save sets, or save sets in one nod frame), and then point at a source on the screen. The plots inside the tab will display the FWHM and centroid, as functions of the frame number. This is particularly useful for identifying bad frame(s).

VI. Frame Register



Frames (nod frames or save sets) can be stacked through frame registration (FR), i.e., the frames are shifted before being combined. This can be used to compensate the jittering between frames due to the seeing effect or imperfect chopping/tracking. In some cases, the final image quality (in terms of FWHM and Strehl ratio) can be significantly improved by doing so.

To use FR, ideally there should be at least one bright source in the field of view.

There are several options or FR. You can choose among three algorithms, image centroid, cross-correlation, or cross-correlation with FFT (Fast Fourier Transform), to measure the frame offset. The first one seems to be fast and reliable for point-like sources. Note that any of these

algorithms may fail (i.e., the calculated frame offset is incorrect) for frames with very low signal-to-noise ratios. So, always be cautious with it. Remove bad frames first, and do not trust the FR result blindly. For your convenience, the measured centroid for each frame will be printed out in the IDL command line during frame registration. You can use these numbers to tell if anything goes wrong.

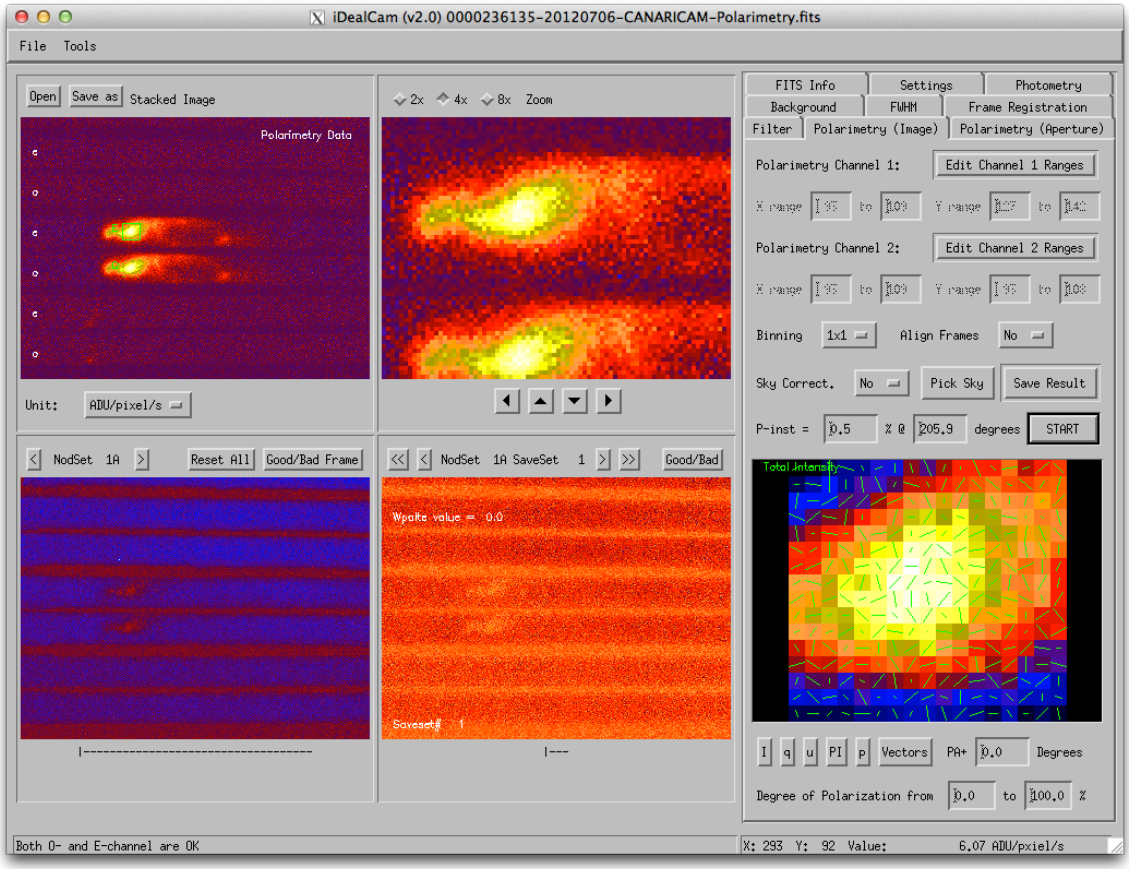
The offset can be measured using the whole frame (slow and maybe problematic in some cases), or s sub-image around a bright source (recommended), you need to enter the position of which manually, though.

You can also choose to filter each frame (via Gaussian Smoothing or Median Smoothing) before measuring the offset. However, this treatment is only used for measuring the offset; the actual frames being stacked are NOT smoothed.

If you are satisfied with the FR-generated image, you can save it as the stacked image, and display it in the upper-left screen (so that you can perform other analyses, like aperture photometry, upon it).

Note that once you change the status (good or bad) of a frame after FR, you need to come back to this tab and do FR again to see the updated result.

VII. Imaging Polarimetry



The entire field of view of a polarimetry image is divided into six slots (i.e., three *O*-and-*E*-ray pairs) by the polo-mask. The positions of the slots are labeled by letters of "e" or "o" along the left side of the stacked image. Note that the structure of a polarimetry FITS file is different from a normal imaging FITS file. Specifically, each nod frame of polo image contains a fixed number (four) of save sets, corresponding to four positions of the half-wave plate (0.0, 22.5, 45.0, or 62.5 degrees, as displayed in the upper-left corner of each save set).

To reduce a polarimetry image with iDealCam, all you need to do is to select a region of interest using the mouse, and then click the button [**Start**]. The region could be within an O-ray slot, or an E-ray slot, but not both. Once a region is selected, iDealCam will automatically find the corresponding region in the other slot, and align them. The boundaries

of the region can be edited manually, if necessary. To avoid overlapping between the o-ray and e-ray images, the size of the selected region should be smaller than 35 pixels (ideally 32 pixels or less) in the Y-direction. There is no limitation on the size in the X-direction.

Results will be displayed once the reduction is done without any error. You can select among images of total intensity (I), polarized intensity (PI), Stokes U/Q component, or degree of polarization (p) for display. You can also choose to overlap the image with polarization vectors or not:

Note 1: the polarization (degree and orientation) of the pixels along the four boundaries of the selected region may be problematic in some cases (due to the frame alignment).

Note 2: the position angle (PA) of the p-vector is not calibrated. It is a value with respect to an internal reference. For the purpose of plotting, we assign 0 degree to the up, 90 degrees to the left, and 270 degrees to the right. If there is a known calibrated offset between the true PA and the calculated PA (through, for example, a comparison between observations of a science object and a polarization standard star), you can enter that value in iDealCam to mimic a calibrated vector map:

Note 3: The telescope produces significant instrumental polarization (P-inst) in the mid-IR. P-inst is a function of the instrument configuration and wavelength. During reduction, iDealCam calculates the P-inst (amplitude and orientation) and does the correction automatically. At this moment, the amplitude is assumed to be 0.5% for all wavelengths, which is reasonably good. Users can overwrite these values in iDealCam:

Note 4: By default, the p-vector of zero polarization has a zero length on the screen, and that of 100% polarization has a length of 1-pixel. You can overwrite this setting by entering new values in the bottom line of the tab:

Other Options of Polarization Reduction

[Binning] The frames can be binned by 2x2, 3x3, or 4x4 pixels before the calculation of I/Q/U intensity to enhance the signal-to-noise ratio.

[Sky Correction] Sky subtraction is not necessary, and by default, there is no sky subtraction during the polarimetry reduction. This is because 1) when you subtract an O-ray image from an E-ray image (or reversely) to get the Q/U intensities, the sky residue, if any, can be removed, and 2) the sky level is indeed very low in most cases, and the sky subtraction has little effect on the final result. Anyway, if one prefers to subtract the sky, it is doable in iDealCam.

[Align Frames] The O-ray and E-ray regions are aligned using the similar algorithm for frame registration. This function can be turned off (not recommended though). However, for very faint object, the SNR in each save set may be too low to perform frame alignment. In this case, it is suggested to turn off this function.

Save the Reduced Images (for plot or other uses)

Using the [**Save**] button, you can save the reduced data into new FITS files that can then be analyzed by other software. These FITS files can also be used for producing high-quality plots, either with your own

plotting routines/tools, or the one comes along with iDealCam (see Appendix I for more instructions on how to use that routine).

Once the [**Save**] button is clicked, the following files are created:

I.fits	Image of total intensity
PI.fits	Polarized intensity
P.fits	Degree of polarization
Q.fits	Stokes Q component
U.fits	Stokes U componets
Theta.fits	PA of polarization

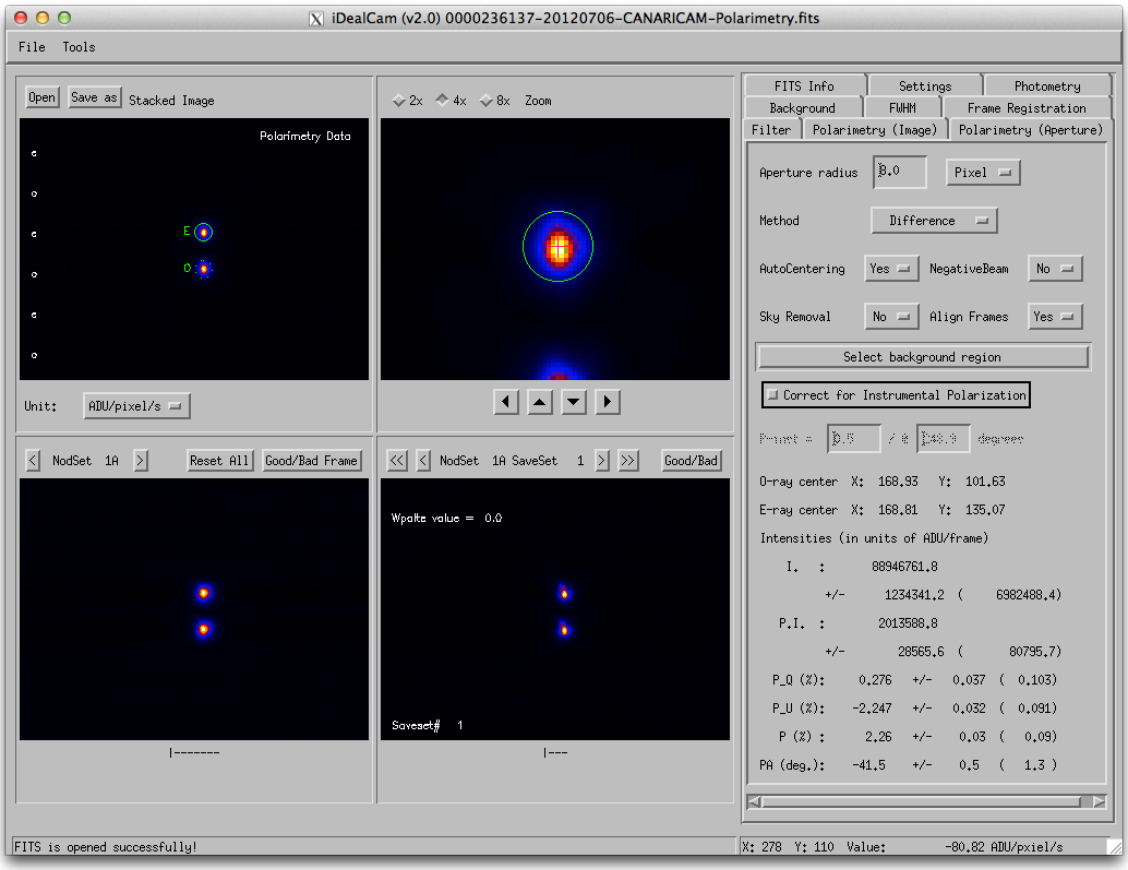
in a new folder, the name of which is as same as the original FITS file. All of the files are ordinary FITS files that can be opened or edited by tools such as DS9.

Note that I, Q, and U files saved in this way are in units of ADU/save set. To get AUD/sec, one needs to know the integration time (t_saveset) for each save set, which is given by:

t_saveset = total on-source time / number of nod cycles / number of nod beams / number of save set per beam

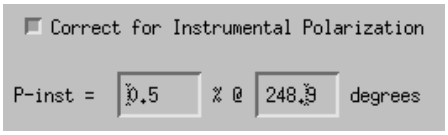
All these values can be found in the panel of FITS Info in iDealCam.

VIII. Aperture Polarimetry



For observations of point-like sources, polarization can be calculated within a circular aperture (similar to aperture photometry). The radius of the aperture is given by the user. There are two algorithms for aperture polarimetry, so called **ratio method** and the **difference method** (Bagnulo et al., 2009. PASP, 121, 993.)

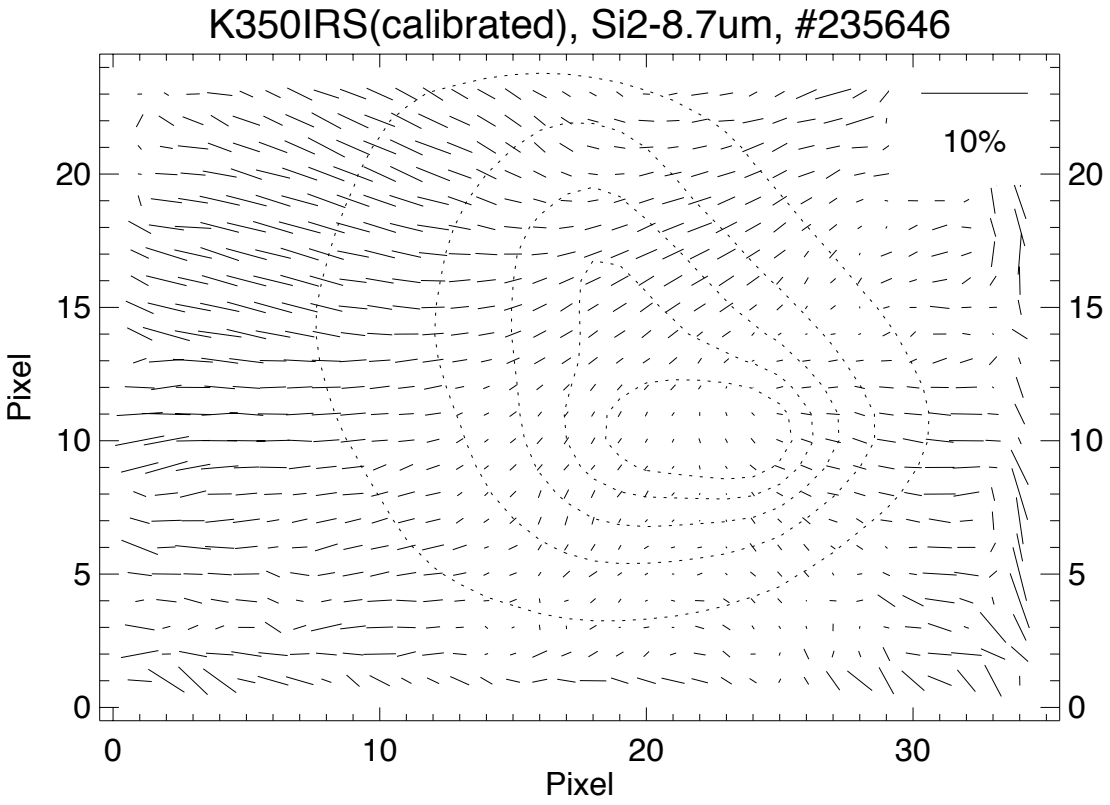
Like in the imaging polarimetry, sky removal is not necessary (and not recommended), but user still can do it in iDealcam. By default, P-inst correction is **NOT** performed in aperture photometry. To do so, one needs to turn this function on manually:



(blank page)

Appendix I. Use DL_POLOPLOT to Plot Polarization Maps

The routine dl_poloplot (included in the zip package) can be used to plot a polarization map like this:



Basically, this routine can plot an image overlapped by p-vectors, the lengths of which are proportional to the degree of polarization. The image could be an image of total intensity, polarized intensity, or Stokes Q/U parameters. The result is saved in EPS format.

To use this routine, the following files should be given:

I.fits (or PI.fits, Q.fits, U.fits)

P.fits (degree of polarization)

Theta.fits (PA of polarization, in units of radian; 0.0 means up; positive directions is counter-clockwise)

All FITS files should be put in the same folder and with same size. For CanariCam data, one can use the [Save] button introduced in Section VII to prepare all those files conveniently.

Examples

The calling of dl_poloplot in IDL command line is like this:

```
IDL> DL_POLOPLOT [, keyword options]
```

A calling with no keyword, i.e.,

```
IDL> DL_POLOPLOT
```

looks for I.fits, P.fits, and Theta.fits in the current folder, displays I.fits in gray scale (with white background) overlapped by p-vectors, and save the plot in a new file named result.eps.

Some other examples:

Looks for input files in a directory named /astro/homes/xxx:

```
IDL> dl_poloplot, dir='/astro/homes/xxx/'
```

Draw image with coutours instead of gray scale:

```
IDL> dl_poloplot, /contour
```

Rotate all p-vectors counterclockwise by 1/2 radian (used for PA calibration):

```
IDL> dl_poloplot, cali_offset=0.5
```

Plot p-vectors only in odd rows and columns:

```
IDL> dl_poloplot, /odd
```

An (incomplete) list of keywords

`dir`	Directory that contains input files.
`portrait`	If the image is longer in the Y-axis, then setting this keyword (/portrait) to give a better appearance.
`width`	Width of the plot region, in inch (default value is 5). The height of the plot region is chosen automatically to get a correct aspect ratio.
`margin_x`	By default, the paper width is 1.2 times the width of the plot region, but user can change this setting by entering a new margin_x value here. For example, letting margin_x=1.1 will leave less space in the right and left sides of the paper.
`margin_y`	Similar to margin_x but controls the margin size in the Y-direction. Default value is 1.25.
`offset_x`	Can be used to move the plot within the paper. For example, letting offset_x=0.01 will move the plot a little bit to the right.
`offset_y`	Similar to offset_y but for Y-direction. These two keywords, plus margin_x and margin_y, are used to finely position the plot in the paper.
`/contour`	Display the image with contours, instead of gray scale.
`title, xtitle, ytitle`	Title of the plot, x-axis, and y-axis, respectively.
`ground`	The pixel value corresponding the lowest contour
`step`	Step between two adjacent contours.
`n_levels`	Total number of contours.
`levels`	Contour levels can also be given explicitly (`ground`, `step`, and `n_levels` will be ignored if `levels` are given).
`c_linestyle`	Line style of contours.
`p_max`	The maximum degree of polarization to display. Pixels with p greater than this threshold will be skipped.
`p_length`	The length of the p-vector for a 100% polarization, in units of pixel width. Default value is 1.0.

`I_min`	The minimum pixel value, for which a p-vector will be displayed. Can be used as an signal-to-noise threshold.
`/odd`	Only plot p-vectors for odd rows and columns.
`cali_offset`	Rotate all p-vectors by a certain angle, in radian. A positive value means to rotate counterclockwise.
`c_vectors`	Color of p-vectors. By default the color table #16 in IDL will be loaded for color plotting, so black=0, green=32, blue=96, red=176, and white=254.
`adutojy`	A scale factor to be multiplied with the pixel values. It could be, for example, the photometric calibration coefficient (to translate ADU/pixel to Jy).
`_extra`	Other keywords that are accepted by PLOT.

For more keywords, see the header of dl_poloplot.pro

Appendix II. Known Restrictions and Bugs

If one tries to open a FITS file that is damaged or not finished properly, iDealCam may halt (not crash). In this case, you can either restart the program, or type in the IDL command line:

```
IDL> retall
```

This will allow you to quit the module of file opening, and hopefully the other functions are still running and you can then select another file to open.

If you find any other bugs or actions that may cause a crash, please send me a message by email (d.li@ufl.edu). Thank you very much!

D.L. Last update on 06/20/2014