ALSVID
Algorithms for Visualization and Processing of Image Data

John Kielkopf & Karen Collins

May 3, 2017

1 Introduction

The programs described here permit processing image data with high precision, in a flexible command line system useful for astronomical and laboratory image processing. This is a work in progress, with its roots in basic code for handling CCD images acquired under Linux, and its branches in precision photometry, spectroscopy, and 3-D visualization. We have adopted the acronym **Alsvid**, an old Norse name meaning "Very Quick" for one of the two horses who pull the Chariot of Sol across the sky.

The software is available from

`http://www.astro.louisville.edu/software/alsvid`

2 Dependencies

The programs under development are all written in Python, a programming language that is a widely used tool for basic research and engineering. Its rapid rise in popularity is supported by comprehensive, largely open-source, contributions from scientists who use it for their own work. Astronomers and physicists have found that it is powerful alternative to restrictively licensed software, or legacy systems developed before modern computing environments became available on every desktop.

The versions in the current distribution have been edited for use in Python 3 and are tested with Python 3.4. They depend on Python modules that are readily available and well-maintained:

Numpy Numerical processing of arrays

Scipy Additional components for scientific data

Astropy Support for Flexible Image Transport (FITS) and World Coordinate System components of FITS files

Scikit-image For advanced processing including Lucy-Richardson deconvolution

Pyastronomy Provides utilities for spectroscopy

Other features useful for astronomy and astrophysics are self-contained in the code and do not require additional libraries.

Alsvid programs are intended for use alongside other Open Source code, especially AstroImageJ (AIJ) for real-time precision photometry and analysis with a sophisticated graphical user interface. SAOImage ds9 may be used for FITS file display. Alsvid contains routines to export and import region files with ds9, and aperture files with AIJ. Additionally, SWARP is excellent for combining large image sets, astrometry.net will add WCS coordinates, and GRACE is an interactive plotting and data analysis program that produces publication quality graphics.

3 Licensing

4 File Storage

Images are held in FITS files which contain a header and binary data. Data are kept with the headers in single files, and the header is updated with the history of the processing. Typical header information at a minimum identifies the target, the acquisition instrument, the exposure duration, filters, and the time at which the exposure was taken. Exposure time in the header may be used in scaling dark files for dark subtraction, and in calibration for photometry. In the case of astronomical data, additional *World Coordinate System* (WCS) data may be included to relate each pixel to a particular direction on the sky. WCS positions are used to identify known objects in photometric reduction scripts.

Camera data are typically 16-bit unsigned integers with a bias to insure that no pixels have negative data. A pixel will therefore range from 0 to 65535. With processing it is possible that values will, on average, go below zero. When several images are added, the information content may exceed 16-bits as well. Therefore, all processed images are stored

is 32-bit floating point data by default. However, the ease of modifying Python code would allow a user to adapt the programs to output other data types. In most cases there are no restrictions in type of input data.

5 Legacy C

Previous programs written in C are available in our on-line archive for compilation when linked with the cfitsio library. The are no longer maintained. In most cases they have been replaced in functionality by the Python programs described here. The C versions are no longer maintained.

6 Alsvid Python routines

Alsvid is a collection of Python routines for command line execution. They are not combined into a library, and each one can easily be edited or modified for other special cases. These programs provide for

- Dark subtraction either by a file with the same exposure time as the image, or by another exposure time that will be scaled automatically when a bias frame is available.
- Flat division to correct for pixel-to-pixel sensitivity and throughput variations.
- Bias subtraction to remove the remove the signal for no light
- Clipping and management of out of range values
- Scaling all values by a polynomial useful for non-linear response correction
- Mean and median averaging
- Removal of a background gradient
- Removal of sky background
- Finding stars in an image
- Removal of field stars from an image
- Summation of a stack of frames
- Centering and summing a stack of frames
- Flipping and rotating images
- Extracting sums of rows or columns for spectroscopy

- Generating an average radial profile of a circularly symmetric object
- Conversion of FITS images to png images with linear or logarithm scaling
- Conversion of png images to FITS images
- Conversion of text image (often from data analysis programs such as LabVIEW) to fits
- Generation of pixel coordinates from equatorial sky coordinates for ds9 and AIJ
- Generation of sky coordinates from pixel coordinates
- Summarizing image statistics
- Change the data type for a FITS image
- Clear an old FITS header
- List and edit FITS headers by keyword
- Aperture photometry
- Temporal Fourier transforms on a uniformly candenced stack
- Lucy-Richardson deconvolution of images

Utilities for generating local sidereal time and Julian Day are provided.

When a routine is executed without arguments or when the wrong arguments are detected it will return usage information (if no argument is required). Routines which write files are written to overwrite existing files by default, but that behavior is easily modified by changing the access flags in the source code.

A current complete list of functions and current command line arguments is available on-line. Please consult the Alsvid website for a link to the latest versions and to support. The following routines are in the latest version.

```
ALSVID File Processing
======================

 file_renumber.py              Renumber files in a directory
 fits_list_exposure.py         List all exposures for FITS files in a directory
 fits_list_head.py             List the FITS header
 fits_list_head_to_csv.py      Export FITS headers to a csv file


ALSVID FITS Image Processing
```

==========

fits_1d_to_dat.py	Extract a 1-dimensional FITS array as data
fits_absolute_value.py	Absolute value of an image
fits_background_remove.py	Fit and subtract a background
fits_bias.py	Subtract a bias frame
fits_bin.py	Bin a stack of images one by one
fits_border.py	Zero values outside borders
fits_clean_head.py	Clear all but essential items from the header
fits_clip.py	Clip an image at minimum and maximum values
fits_convert.py	Convert an image from one type to another
fits_dark.py	Subtract a dark frame from an image
fits_derivative.py	Create a derivative stack from a temporal stack of FITS images
fits_divide.py	Divide one FITS file by another
fits_edit_head.py	Edit the FITS header
fits_fft.py	Create a frequency stack from a temporal stack of FITS images
fits_fft_test.py	Template to generate test stack for fits_fft.py
fits_find_stars.py	Find stars in an image
fits_fix_col.py	Repair a bad column
fits_flat.py	Divide an image by a flat frame
fits_flip_lr.py	Flip an image left-right
fits_flip_ud.py	Flip an image up-down
fits_from_png.py	Convert a PNG file to a FITS file
fits_from_raster.py	Built a FITS file from a stack of 1d raster data files
fits_from_raw_dslr.py	Exctract R, B, and B fits images from a RAW Canon, Nikon, or Sony image
fits_from_text.py	Create a FITS image from a text file
fits_from_tifs.py	Generate FITS images from TIF files
fits_level.py	Fit and remove a plane gradient
fits_lucy_richardson.py	Peform interative Lucy-Richardson deconvolution using a Gaussian PSF on a FITS image file
fits_make_threshold_mask.py	Create a mask by setting threshold levels
fits_mask.py	Mask regions from a FITS image
fits_mean.py	Take the mean of several images
fits_median.py	Take the median of several images
fits_mef_to_fits_images.py	Extract individual FITS images from a Multi-Extension file
fits_multiply.py	Multiply two FITS images of the same size
fits_nan_to_num.py	Change "NAN" elements to numbers

```
fits_norm.py                    Normalize an image
fits_nstats.py                  Statistics on a stack of images
fits_phoenix_hires_to_dat.py    Extract spectra from a PHOENIX model
fits_pixel_photometry.py        Aperture photometry on an image
fits_pixel_to_wcs_photometry.py  Aperture photometry outputing RA and Dec
fits_pix_to_ds9.py              Convert an x,y list to ds9 regions
fits_pix_to_sky.py              Convert an x,y list to a WCS ra,dec list
fits_radial_average.py          Take an average assuming circular symmetry
fits_rd.py                      Create a random decrement autocorrelation
                                  stack from a temporal stack of FITS images
fits_remove_stars.py            Use a pixel x,y list to remove stars from an
                                  image
fits_remove_stars_with_psf.py   Remove stars and replace based on a model
                                  point spread function
fits_roll.py                    Rolls and wraps by dx and dy within the
                                  same image size
fits_rotate_90.py               Rotate an image in 90 degree increments
fits_rotate.py                  Rotate an image an arbitrary angle
fits_scaled_dark.py             Dark subtraction scaling exposure time
fits_scale.py                   Quadratically scale image data
fits_sky_to_aij.py              Create AIJ x,y apertures from a sky ra,dec list
fits_sky_to_ds9.py              Create ds9 x,y regions from a sky ra,dec list
fits_sky_to_pix.py              Create plain x,y text from ra,dec
fits_sqrt.py                    Create a new image that is a square root of
                                  the input image
fits_stats.py                   Statistics on a single image
fits_subtract.py                Subtract two FITS images of the same size
fits_sum_centered.py            Center and sum an image stack
fits_sum_cols.py                Sum selected columns (for spectra)
fits_sum.py                     Sum an image stack
fits_sum_region.py              Sum over a region bounded by rows and columns
fits_sum_rows.py                Sum selected rows (for spectra)
fits_to_float32.py              Convert an integer (or other) FITS image to
                                  32-bit float image
fits_to_lin_png.py              Create a linear 16-bit gray-scale png
fits_to_log_png.py              Create a logarithmic 16-bit gray-scale png
fits_viewer.py                  Simple image viewer for a single FITS image


ALSVID Utilities for Date and Time
==================================
```

jd.py	Provide the Julian day now
lst.py	Provide the local sidereal time now
utc.py	Provide the universal time now

ALSVID Utilities for Spectroscopic Data
==========

spectrum_airtovac.py	Convert spectral 2-column wavelength, flux file from air to vacuum wavelengths
spectrum_crosscorr.py	Cross correlate target and template spectra to find the radial velocity of the target
spectrum_heliovel.py	Return the barycentric velocity correction accounting for the topocentric motion of the observer
spectrum_rotbroad.py	Broaden a stellar template spectrum with a model of vsini and linear limb darkening
spectrum_vactoair.py	Convert spectral 2-column wavelength, flux file from vacuum to air wavelengths