ALSVID
Algorithms for Visualization and Processing of Image Data

John Kielkopf & Karen Collins

July 1, 2021

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 3.9. 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 version 5.9 released on June 30, 2021.

```
ALSVID Python utilities for working with FITS images and data
=============================================================

  fits_1d_to_dat.py
    Extract a 1-dimensional FITS array as data
  fits_absolute_value.py
    Absolute value of an image
  fits_add_datetime.py
```

```
  Add a date and time to a FITS file header
fits_add_filter_to_filename.py
  Add the filter id to the file name
fits_add_instrument.py
  Add an instrument to the FITS file header
fits_autocorrelate.py
  Autocorrelated images from stack of  fits images
fits_background_remove.py
  Fit and subtract a background
fits_bias.py
  Subtract a bias frame
fits_bin_1d.py
  Bin a stack of images along the time or z-axis
fits_bin_2d.py
  Bin nxn each image in a stack
fits_border.py
  Zero values outside borders
fits_clean_head.py
  Clean 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_convolve_gaussian.py
  Convolve an image with a Gaussian blur
fits_copy_header.py
  Copy header from one fits file to another
fits_correlate.py
  Correlation  from a temporal stack of  fits images
fits_crop_all.py
  Crops fits all files in a directory
fits_crop.py
  Crops a single fits file
fits_dark.py
  Subtract a dark frame from an image
fits_derivative.py
  Create a derivative stack from FITS images
fits_divide.py
  Divide one FITS file by another
fits_edit_head.py
  Edit the FITS header
fits_fft_2d.py
```

```
  Create a stack of 2D Fourier Transformed images
fits_fft.py
  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_exr.py
  Create RGB FITS files from a 16-bit color EXR file
fits_from_png.py
  Convert a PNG file to a FITS file
fits_from_pngs.py
  Convert PNG files 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
  Extract R, B, and B fits images from a RAW image
fits_from_text.py
  Create a FITS image from a text file
fits_from_tifs.py
  Generate FITS images from TIF files
fits_histogram.py
  Export a histogram for an image
fits_level.py
  Fit and remove a plane gradient
fits_list_date-obs.py
  List the dates of observation for files in a directory
fits_list_date.py
  List the file dates from the FITS headers of files in a directory
fits_list_exposure.py
  List all exposures for FITS files in a directory
fits_list_head_entry.py
  List all the values for a header entry searching files in a directory
fits_list_head.py
```

```
  List the FITS header for a file
fits_list_head_to_csv.py
  Make a CSV file of  the FITS headers for all files in a directory
fits_lucy_richardson.py
  Peform interative Lucy-Richardson deconvolution
fits_make_threshold_mask.py
  Create a mask by setting threshold levels
fits_mask.py
  Mask regions from a FITS image
fits_mast_to_dat.py
  Extract data from MAST spectral fits table file
fits_mean.py
  Take the mean of several images
fits_median_1d.py
  Take the median of several images
fits_median_2d.py
  Median filter 3x3 all images in a stack
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 from pixel coordinates
fits_pixel_to_wcs_photometry.py
  Aperture photometry from pixel coordinates outputing sky coordinates
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
fits_relative_transients.py
```

```
  Identify relative transient events in an otherwise static image stack
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_sigma.py
  Create a standard deviation (sigma) image from a stack of fits images
fits_signal_autocorrelate.py
   Autocorrelated images from stack of  fits images using scipy signal
fits_sky_radec_to_aij.py
  Convert an RA Dec AstroImageJ file to a an AIJ x,y apertures file
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_sliding_median.py
  Perform a sliding median smoothing to an image stack
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
  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_to_tiles.py
    Generate a stack of tiled png files for web display of large images
  fits_unmask.py
    Unmask regions from a FITS image
  fits_viewer.py
    View a FITS image in Python GUI
  fits_wcs_photometry.py
    Aperture photometry from sky coordinates



ALSVID Python utilities
=======================

  aij_table_reader.py
    Read an AIJ data table
  bls_astropy_bokeh.py
    BLS search of data with bokeh plot output
  decimal_deg_to_dms.py
    Convert decimal degrees to dd:mm:ss.sss
  decimal_radeg_to_hms.py
    Convert decimal RA in degrees to hh:mm:ss.sss
  dms_to_decimal.py
    Convert ddd:mm:ss to decimal
  file_renumber.py
    Renumber files sequentially
  jd.py
    Provide the Julian day now
  lomb_scargle_astropy.py
    Lomb Scargle search using astropy code
  lomb_scargle_scipy.py
    Lomb Scagle search using scipy code
  lst.py
```

```
  Provide the local sidereal time now
moon.py
  Position and phase of the Moon now and a list for other JD's
plotly_data.py
  Plot an x,y data file interactively with Plotly
process_fits.py
  Batch process raw data in a directory based on a configuration file.
query_mast_gaia_to_csv.py
  Query MAST for Gaia stars around a target
query_simbad_to_aij.py
  Query Simbad for to produce AstroImageJ apertures
sliding_median_normalize.py
  Normalize a data file with a sliding median
spectrum_airtovac.py
  Convert flux file from air to vacuum wavelengths
spectrum_crosscorr.py
  Cross correlate target and template spectra to find the radial velocity
spectrum_heliovel.py
  Barycentric velocity correction for the topocentric motion
spectrum_rotbroad.py
  Broaden a stellar template spectrum
spectrum_thermal.py
  Compute a black body spectrum at requested temperature
spectrum_vactoair.py
  Convert spectral flux file from vacuum to air wavelengths
sun.py
  Position of the Sun now and a list for other JD's
tiles_from_png.py
  A companion to fits_to_tiles.py taking a png file as input
tk_plot_3d.py
  Plot x,y,x data interactively with a Tk interface
tk_plot.py
  Plot x,y data interactively with a Tk interface
tk_query_mast_edr3_to_aij_radec.py
  Query MAST for Gaia EDR3 stars around a target with Tk interface
unix_time_from_date.py
  Return Unix time from a data and time
unix_time.py
  Current Unix time
utc.py
  Current universal time
```

```
ALSVID for TESS data processing
===============================

  ffi_for_fft.py
    Prepare a time series image stack for temporal FFT
  fits_clean_up_ffi.py
    Remove all low DATAQUALITY and empty TESS FFI's
  fits_detrend_sliding_median.py
    Detrend an image stack with a sliding median
  fits_detrend_sliding_minimum.py
    Detrend an image stack with a sliding minimum
  fits_extract_tess_background.py
    Extract background from  a stack of TESS fits images
  fits_extract_tess_cutout_images.py
    Simple stack of FITS images from a TESS cutout BINTABLE file
  fits_extract_tp_images.py
    Simple stack of FITS images from a TESS fast cadence TIC pixel file
  fits_ffi_to_simple_images.py
    Simply full frame TESS images by removing all but the image
  fits_find_tess_annulus_background.py
    Find the background of a TESS FFI images
  fits_level_by_column.py
    Remove median bias for column noise
  fits_list_xhead_entry.py
    Values of  entry in the first extended FITS header for a directory
  fits_list_xhead.py
    Entire first extended header for a FITS file
  fits_relative_transients.py
    Identify relative transient events in an otherwise static image stack
  fits_remove_tess_background.py
    Remove background from  a stack of TESS fits images
  query_mast_gaia_to_csv.py
    Search a field for Gaia stars and generate a database
  query_mast_tic_to_csv.py
    Search a field for TIC stars and generate a database
  query_simbad_to_aij.py
    Object named on the command line to an AIJ aperture format file
  query_tic_to_aij.py
    Search field for TIC stars and generate an AIJ aperture file
  tk_query_mast_edr3_to_aij_radec.py
```

```
    Search Gaia EDR3 for nearby stars and generate an AIJ aperture file
  tk_query_mast_tic_to_aij_radec.py
    Search TIC for nearby stars and generate an AIJ aperture file
  tls.py
    Transit least squares model from  time-series data
  transitmodel.py
    Model for transit photometry and compare to an observation


ALSVID with Julia Languate
==========================


  transitmodel.jl
    Planet, star and orbit parameters generate  exoplanet transit model
```