Astronomical Data Analysis Software and Systems X
ASP Conference Series, Vol. 238, 2001
F. R. Harnden Jr., F. A. Primini, and H. E. Payne, eds.

Infrared Imaging Data Reduction Software and Techniques

C. N. Sabbey, R. G. McMahon, J. R. Lewis, M. J. Irwin

Institute of Astronomy, Madingley Road, Cambridge, CB3 0HA, UK

Abstract. Developed to satisfy certain design requirements not met in existing packages (e.g., full weight map handling) and to optimize the software for large data sets (non-interactive tasks that are CPU and disk efficient), the InfraRed Data Reduction[1] software package is a small ANSI C library of fast image processing routines for automated pipeline reduction of infrared (dithered) observations. The software includes standalone C programs for tasks such as running sky frame subtraction with object masking, image registration and co-addition with weight maps, dither offset measurement using cross-correlation, and object mask dilation. Although currently used for near-IR mosaic images, the modular software is concise and readily adaptable for reuse in other work.

1. Introduction

The Cambridge Infrared Survey Instrument (CIRSI), a near-IR mosaic imager containing a 2 x 2 array of Rockwell Hawaii I 1024 x 1024 detectors (Beckett et al. 1996; Mackay et al. 2000), has been in operation for about two years, obtaining almost 1 TB of imaging data. The uniquely wide field accessible by CIRSI on a 2–4 m class telescope makes it ideal for moderate depth large-area surveys. Preliminary results from two such current surveys include the measurement of galaxy clustering at intermediate redshift (McCarthy et al. 2001), and the demonstration of a reddening-independent quasar selection technique based on combined deep optical and near-IR color diagrams (Sabbey et al. 2001).

However, the CIRSI data reduction poses several challenges. With the large data rate (5–10 GB of data taken per night, ∼100 nights per year, and a significant data backlog currently) the software has to be very efficient and completely automated. Also, the software should handle diverse data sets, from Galactic center observations to very sparse fields at high Galactic latitude. From thousands of individual images taken over many nights, wide-field, deep mosaic images are generated. Since the gaps between detectors comparable to detector size, filled mosaic images are made using the co-added dither sets from different chips and telescope pointings. Thus weight maps and accurate astrometry are crucial and the common simplification of clipping the dither sets to their intersection region is not appropriate.

[1]IRDR, available via anonymous ftp at ftp.ast.cam.ac.uk in pub/sabbey

© Copyright 2001 Astronomical Society of the Pacific. All rights reserved.

Although existing software packages are used when possible (see below), the decision was made to write core image processing routines to satisfy certain design requirements. For example, a two pass reduction provides: object masks derived from first pass co-added dither sets, subpixel image registration/co-addition using full-weight maps without image clipping, optimizations for CPU and disk efficiency, customized artifact cleaning (destriping and defringing), and reusable tools (from having stand-alone tasks to be glued together with a high level scripting language, to making C library calls, or even extracting portions of source code). The basic processing steps, described below, are: flatfield correction, running sky frame subtraction, dither offsets measurement, dither set co-addition, and mosaic image creation.

2. Data Reduction

2.1. Flatfield Correction

Flatfield images are produced by subtracting stack medians of lamp-off from lamp-on dome flats. The flatfield images are divided by the mode of the chip 1 flatfield to produce a gain map per chip. Bad pixels are automatically identified in the gain maps (and set to 0.0) by looking for outliers ($> 5\sigma$ from the median in 15x15-pixel blocks) or pixels with extremely low or high sensitivity ($> 30\%$ from the median gain). Because bad pixels often occur in clumps, these are eliminated during image co-addition rather than by interpolation in an initial cleaning pass. The data frames are multiplied by the inverse of the gain map.

The image stacking is done using `cubemean.c`, which calculates the median, robust standard deviation, or robust mean plane with a choice of weights (none, scalar, or maps) and image scaling or zero offsets using the image modes. This is not a general purpose tool like IRAF's `imcombine`, but for the specific task of calculating the median plane was found to be 2.5 times faster (for a stack of 50 data frames). The flatfield image is converted to a gain map and bad pixels identified using `gainmap.c`. The data are flatfield corrected using `flat.c`.

2.2. Running Sky Frame Subtraction

With `skyfilter.c`, a sky image is constructed from the robust mean of the eight nearest frames in the observation sequence and subtracted from each data frame. Objects detected in the co-added dither sets from the first pass reduction are masked out during sky frame creation in the second pass. The object masks are produced using the check image `OBJECTS` option to SExtractor (Bertin and Arnouts 1996), which produces a FITS image with non-object pixels set to 0. This is simpler and more effective than building masks from a catalog of object coordinates and shape parameters. The object regions (detection isophotes) generated by SExtractor are then expanded by a multiplicative factor of 1.5 (using `dilate.c`), thereby growing the mask regions for large objects more than small objects. The object masks for individual frames are obtained on the fly using pixel offsets (i.e., the dither offsets) into the master dither set masks.

Running sky frame subtraction is normally a significant bottleneck in processing infrared imaging, so optimizations are important. To do running sky frame subtraction for a stack of N images, the program `skyfilter.c` uses a

sliding window (circular buffer of image pointers) to require only N image reads and N image mode calculations. In contrast, putting this logic into a script normally involves N × M image reads and mode calculations, where M is the width of the sky filter in frames. Also, the disk I/O (and storage) for non-co-added data uses short integers (2 bytes deep), even though most calculations are done in floating point (4 bytes). Some calculations work with short integer data however to allow optimizations. For example, the almost trivial distribution sort can be used to obtain an image histogram, sorted image array, and median value in $O(n)$ time (∼5 times faster than running an optimized median routine on typical data images).

2.3. Dither Offsets Measurement

A typical dither sequence consists of nine observations in a 3 × 3 grid with offsets in each direction of $\sim 10''$. The approximate dither offsets stored in the FITS header WCS information are refined using cross-correlation analysis (`offsets.c`). The non-zero (object) pixels of the reference frame object mask (SExtractor OBJECTS image) are stored in a pixel list (x, y, brightness), and this list is cross-correlated against the object mask images of the following frames in the dither set. The SExtractor OBJECTS image conveniently removes the background (important for cross-correlation methods) and identifies the object pixels more reliably than a simple thresholding algorithm (e.g., especially in images with a non-flat background, large noise, and cosmic rays). Using an object list in the cross-correlation focuses on the pixels that contribute to the cross-correlation signal and is faster than cross-correlating two images.

The cross-correlation technique uses coordinate, magnitude, and shape information and is more reliable than matching object coordinate lists (improvement was noted in extreme cases, e.g., Galactic center images and nearly empty fields with an extended galaxy). Subpixel offset measurement accuracy of ∼ 0.1 pixel is obtained by fitting a parabola to the peak of the cross-correlation image. This cross-correlation method was found to be ∼ 10 times faster (for typical survey data and a relatively large search box of 100 pixels) than IRAF STSDAS `crosscor`. Although the success rate is ∼100%, offset measurements corresponding exactly to the search area border, or a small fraction of object pixels overlapping in the aligned data images, would indicate failure.

2.4. Dither Set Co-addition

A weight map is generated on the fly for each data frame with the weight for pixel p_i given by: $w_i = g_i \cdot t/V$, where g_i is the gain for pixel p_i (0.0 for bad pixels), t is the exposure time, and V is the image variance. The data frames and corresponding weight maps in the dither set are registered using bi-linear interpolation modified to account for bad pixels and image weights. Each output (interpolated) pixel value P is calculated from the weighted average of the four overlapping input pixels p_i of the input image:

$$P = \frac{1}{W} \sum_{i=1}^{4} a_i w_i p_i, \quad W = \sum_{i=1}^{4} a_i w_i$$

where a_i are pre-calculated fractional areas of overlap of P with p_i, and w_i are image weights for p_i. The weight maps are registered similarly, but weight W

for pixel P is calculated via a weighted sum. An alternate registration method sometimes recommended is to replicate each image pixel into N x N pixels and do an integer shift in units of these new pixels. Although this will approximate the above bi-linear interpolation scheme as N becomes large, it requires more work for less precision. Since the default approach is fast and reasonable, especially given the low signal-to-noise ratio of the individual data frames, higher order interpolators (see e.g., Devillard 2000) have not been tested.

With `dithercubemean.c`, dither frames are combined by calculating the weighted mean pixel value at each (x, y) position of the dither stack, with pixel values $> 5\,\sigma$ from the median at each position rejected. Images borders are added during registration to avoid clipping the data to the intersection of the dither frames. The standard deviation (σ) at each position is calculated from $\sigma = \mathrm{MAD}\ /\ 0.6745$, where MAD is the median absolute deviation from the median (simpler methods such as minmax rejection are inappropriate when taking averages of small numbers of values, e.g., 5–9 frames per dither set). The weight maps are combined by calculating the sum at each (x, y) position of the stack of weight maps (for pixels not clipped during co-addition).

2.5. Mosaic Creation

The current astrometry pipeline, a small SExtractor Perl script that produces an object catalog for each co-added dither set, runs APMCAT (a stand-alone C program)[2] to download over the network the APM sky coordinates of objects in each field of view and then runs IMWCS from WCSTools (Mink 1999) to calculate the astrometry fit and update FITS header WCS information. Co-added dither sets and weight maps from different chips, telescope pointings, and nights are then drizzled onto a wide-field mosaic image using EIS Drizzle[3]. Astrometry residuals between the mosaic image and the APM catalogue show a random error of $\sigma \approx 0.3''$ without significant systematic effects.

References

Beckett, M. G., et al. 1996, SPIE, 2871, 1152

Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393

Devillard, N. 2000, The Messenger, 100, 48

Mackay, C. D., et al. 2000, SPIE, 4008, 1317

McCarthy, P., et al. 2001, in ESO "Deep Field" Workshop, held in Garching, Germany, October 2000, astro-ph/0011499

Mink, D., 1999, in ASP Conf. Ser., Vol. 172, Astronomical Data Analysis Software and Systems VIII, ed. David M. Mehringer, Raymond L. Plante, & Douglas A. Roberts (San Francisco: ASP), 498

Sabbey, C. N., et al. 2001, in New Era of Wide Field Astronomy, held in Preston, England, August 2000, astro-ph/0012294

[2] `http://www.ast.cam.ac.uk/~apmcat`

[3] `http://www.eso.org/eis`