AUSTRIA In-Kind Contribution: Data Reduction Software Project

Progress Report

28th July 2010

1 Introduction

This is the fourth progress report on the Austria In-kind Contribution in the context of the *Data Reduction Software Project*, covering the period from April to July 2010.

Currently, work is concentrating on finalising the deliverables DR1 and DR2, which concern the astrometric and photometric correction (DR1) and the stacking and mosaicing of dithered images (DR2). CPL recipes have been implemented in for sky subtraction for near-IR and optical imaging data and were submitted as a prototype release to ESO on 30 June 2010. The determination of astrometric solutions is done through the Terapix programmes `SExtractor`, `Scamp` and `Swarp`.

The main events since the previous status report were:

- **Oliver Czoske** and **Saptarshi Das** visited ESO from 19 to 21 May 2010. During the visit, code structure and the Reflex workflow were discussed.
- **Saptarshi Das** and **Oliver Czoske** developed and coded an instrument-independent sky-subtraction recipe for near-infrared data. Details on the implementation can be found in Sect. 2 and in a separate report.
- **Tomasz Hrycak** implemented a number of mathematical tools, concerning in particular linear algebra and orthogonal polynomials. He used these tools to develop functions for the subtraction of a smoothly varying background from optical imaging data (Sec. 3).
- **Darian Onchis** visited ESO from 3 to 5 May 2010. During the visit, he made a presentation about a prospective new procedure for stacking and improving the resolution of fits images using projections in multi-windows spline-type spaces, developed together with **Hans Feichtinger**. Also, he had discussions about the workflow and code parallelization.
- **Darian Onchis** commenced working on DR03. First goal is a recipe for the stacking of one-dimensional spectra (Sect. 5).
- **Werner Zeilinger** tested the sky subtraction routines in `ESOMVM`. He also assembled spectroscopic test data sets from UVES, FORS and X-Shooter for DR03.

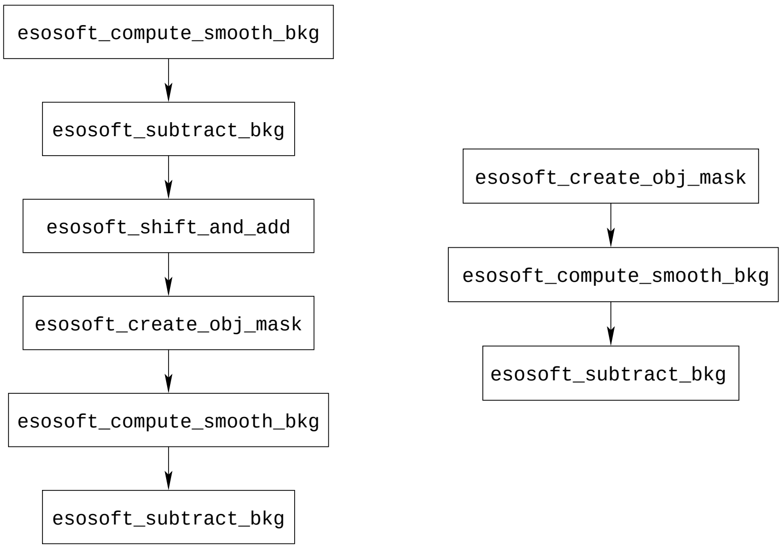


Figure 1: Sequence of recipes to be used for (left) the two-pass correction for NIR images and (right) the smooth correction for optical images.

2 NIR sky subtraction (DR01)

A prototype version (`esosoft-0.0.1`) of the instrument-independent sky subtraction recipes was submitted to ESO on 30 June 2010. The recipes were included in the Reflex workflow by Armin Gabasch. One requirement for the packaging of the software modules that was identified in this attempt is the need for the CPL library to be linked against `wcslib`, which is used in the recipe `esosoft_shift_and_add`.

Since release of the prototype, work on the code has been done to improve the use of static bad-pixel masks.

The NIR background correction is performed in a two-step process which is shown the left panel of Fig. 1 and described briefly in the following. The process works on the set of pre-reduced exposures from a single observing block:

1. For each exposure, stack in the image frame a number of normalized exposures taken immediately before and after using a robust estimator of the mean (`esosoft_compute_bkg`). This produces a first-pass background correction image.

2. Subtract the background correction images from the exposures (`esosoft_subtract_bkg`). This produces the first-pass corrected images.

3. Stack the first-pass corrected images in the world frame, using integer pixel shifts based on WCS information from the fits headers (`esosoft_shift_and_add`).

4. Create an object mask by running an object detector on the combined image. The object detector masks pixels with values exceeding a certain threshold above the background level. The mask is then morphologically opened and grown to include faint object wings (`esosoft_create_obj_mask`).

5. Create the second-pass background correction images as in step 1, this time using the object mask to reject pixels affected by object flux (`esosoft_compute_bkg`).

6. Subtract the second-pass background correction images from the pre-reduced exposures to create the second-pass corrected images.

In cases where the target field contains extended or diffuse emission of astronomical interest (large galaxies, gas nebulae), observers should schedule a number of exposures taken on a patch of empty sky interspersed within the science exposures. In this case, the background correction image should be computed from the sky exposures, not the object exposures. This case can be handled by the recipes if the SOF files are written in the appropriate way.

Currently, validation tests for the performance of the recipes on a variety of fields with different object density. Details on the results of these tests will be given in a separate report.

3 Smooth sky subtraction (DR01)

Optical images are much less affected by sky background emission than NIR images. The variation of the sky background across the field of view is smooth which means that, unlike the NIR background which has to be estimated from a set of exposures not containing the exposure to be corrected, the optical background can be estimated from the exposure itself. To this end, a highly smoothed version of the exposure is produced, e.g. by median filtering or fitting of a parametric model.

Here, a fit using orthogonal polynomials in x and y is used. The method is implemented in the recipe `esosoft_compute_smooth_bkg`, which uses support functions from `esosoft_linalg.c`, `esosoft_matrix.c` and `esosoft_orth_poly.c`. The latter extend the functionality of CPL for general linear algebra, matrix computations and orthogonal polynomials.

The procedure for the background correction for optical images is as follows (cf. the right hand panel in Fig. 1):

1. detect objects and create an object mask (`esosoft_create_obj_mask`)

2. fit a smooth background model (`esosoft_compute_smooth_bkg`)

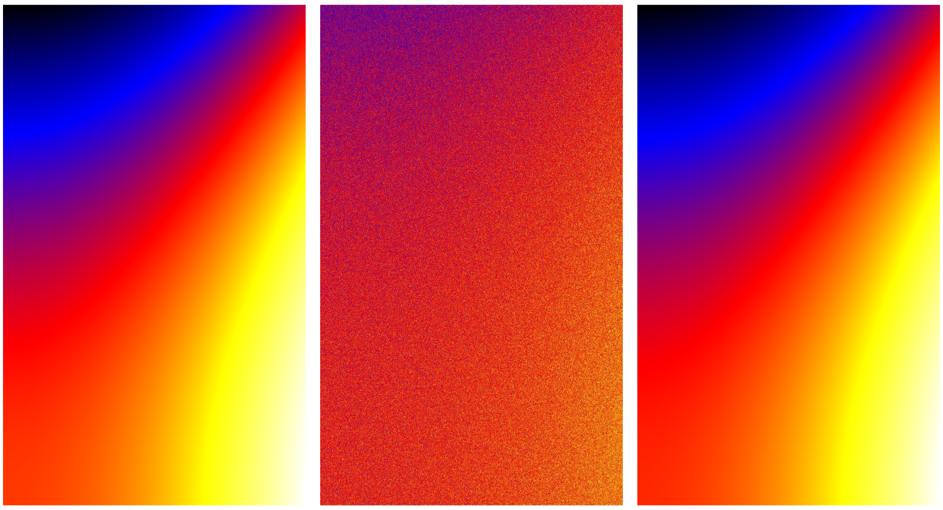
3. subtract the background model (`esosoft_subtract_bkg`)

The preparatory step attempts to clean the set of pixels over which the polynomial is fitted from object contamination.

Next, the noisy sky background is regularised, and partially denoised, by means of a least squares fit with the polynomials of two variables.

The least squares fit is computed using the Wiener filter, the method is also known as the Tikhonov regularisation or the ridge regression. The method requires the solution of the normal equations with the diagonal incremented by a constant reflecting the conditioning of the problem and the amount of noise (e.g. the variance of the Gaussian noise).

The resulting polynomial approximation is a linear combination of the basis polynomials. For reasons of numerical stability, the basis consists of the tensor products of the Legendre polynomials. Specifically, each basis polynomial has the form $P_i(x)P_j(y)$, where P_i is the Legendre



(a) The original image. (b) The original image with added noise. (c) The regularised noisy image.

Figure 2: Least squares regularisation of the sky background.

polynomial of degree i rescaled according to the size of the image. When used with rectangular images, the Legendre polynomials in the variable x have a different scaling than those in the variable y. The most practical basis consists of all tensor products $P_i(x)P_j(y)$ with $i + j \leqslant k$, where k is a fixed integer.

Figure 2 presents regularisation results obtained with 6 tensor products at 15% noise level.

4 Fringe correction using the two-pass method (DR08)

The two-pass method (Sect. 2) can also be used to subtract fringes from optical images taken through red filter bands (e.g. I band). Fig. 3 shows a section of a VIMOS I-band image of the Chandra Deep Field South before and after the application of the standard two-pass method. The very strong fringe pattern is cleanly removed.

5 Progress on DR03

We have begun the work for the DR03 package by checking the bibliographical references and the workflow for VIPGI, the VIMOS Interactive Pipeline Graphical Interface[1] (MOS mode). Darian Onchis made a presentation to the team where he presented the detailed specifications for DR03. Also, fringe removal algorithms where discussed based on wavelets and on an algorithm using computational algebraical topology presented by Pedro Real (University of Sevilla). Further Matlab experiments on fringe removal where done by Hans Fechtinger, using the Gabor

[1] `http://cosmos.iasf-milano.inaf.it/pandora/vipgi.html`

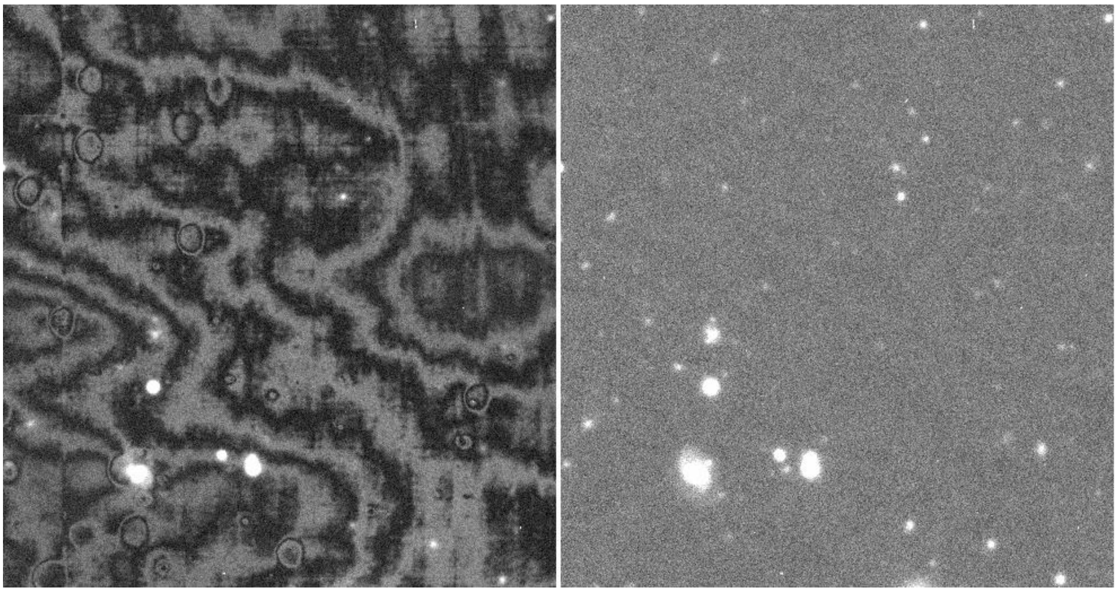


Figure 3: VIMOS I-band image of Chandra Deep Field South before and after fringe correction using the two-pass method. Field size is 1.67 arcmin (488 pixels) in the x direction.

transform and some assumption on the shape. The results were very encouraging on complicated fringes. These advanced algorithm will be studied further with the possible perspective of implementing them as CPL recipes.

Werner Zeilinger queried the ESO archive and assembled suitable test data sets for the stacking of one-dimensional spectra. Using these test data as input, we created a first Matlab prototype for stacking the 1D spectra. The algorithm is now in the process to be transfered into CPL implementation where further functionalities will be added. The actual algorithm presently tested using data from UVES (obtained from the Advanced Data Products archive), but the efforts are in the direction of making it instrument independent.

6 Notes on CPL

Working intensively with the Common Pipeline Library, we occasionally encounter features that we find problematic. The severest are mentioned in the following.

6.1 CPL definition of the median

At several points in our recipes, the median of a sample of data points has to be computed. The Common Pipeline Library[2] has functions to obtain the median from arrays, vectors, matrices and images. Unfortunately, the implementation of the median in CPL is flawed and we had to decide against using it in our modules.

[2] We test against versions 5.0.1 and 5.2.0

We apply the median to samples of (quasi-)continuous data and wish to obtain an unbiased estimate of the population median. When stacking a series of images, the median is a useful robust estimate of the mean of the distribution which can be assumed to be symmetric, so that the population mean equals the population median.

For samples whose size N is odd, the sample median is given by the element number $\frac{N+1}{2}$ of the ordered sample; this provides the required unbiased estimator of the population median.

For samples with even size, the unbiased estimator is given by the average of the elements $N/2$ and $N/2+1$. This definition of the sample median is implemented in standard software packages like `matlab` and `R`. CPL, on the other hand, implements the sample median for even-sized samples as the *lower* of the two central values of the sample, i.e. the value of element $N/2$. Fig. 4 demonstrates that this definition leads to jumps in the background of an image stack in places where the number of exposures changes; this is a result of the bias in the CPL definition of the median.

We argue that the implementation of the median in CPL should be changed to provide an unbiased estimator of the population median.

6.2 Return values of some functions

Several functions that compute summary statistics on matrices and arrays return 0.0 on error and do not set an error code. This makes catching errors difficult, in particular because 0.0 is a valid result of these functions. Examples are `cpl_array_get_mean` or `cpl_matrix_get_median`. In the vector and image versions of these functions, error handling is done properly.

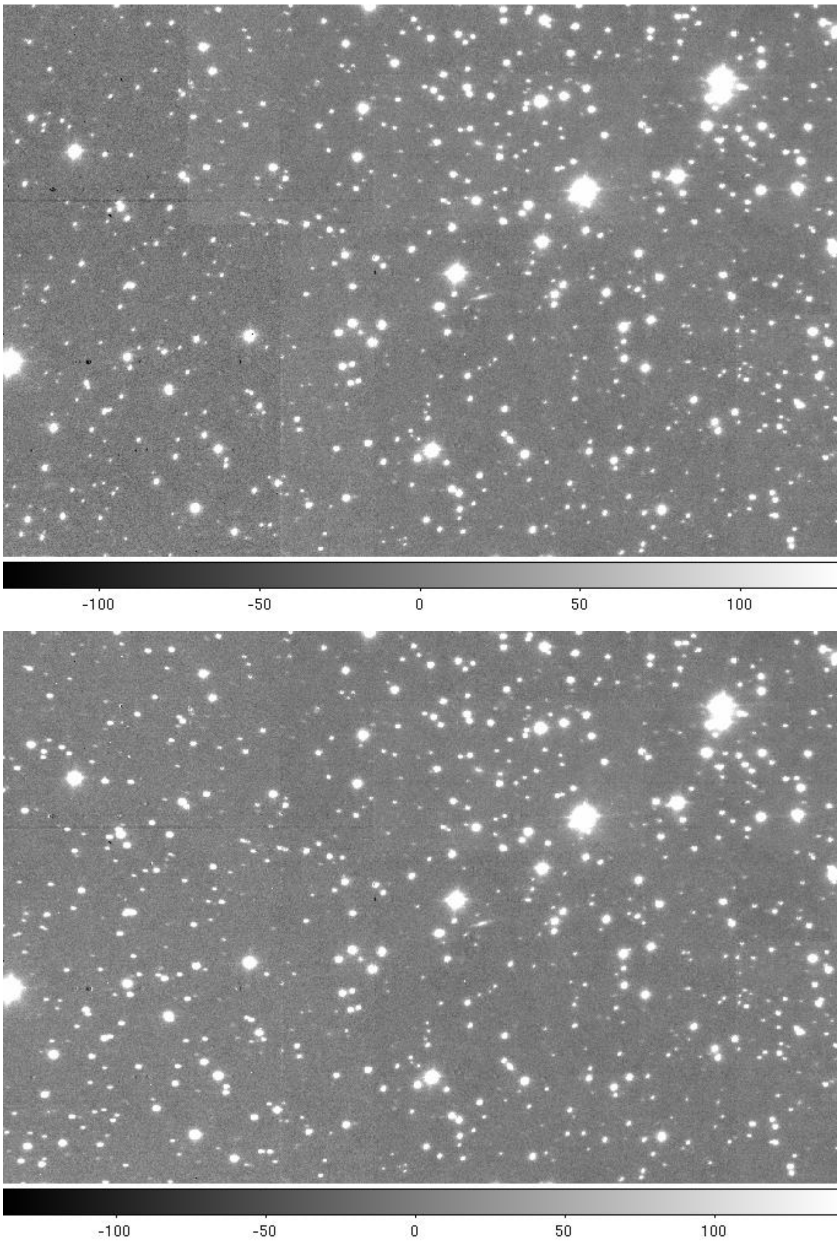


Figure 4: The panels show the median stacks of a set of HAWK-I exposures on Messier 4 created using `esosoft_shift_and_add`. The left third of the field is covered by two exposures, the remainder by three or more exposures. The stack in the top panel was created using the CPL implementation of the median; the left third is noticeably darker, a clear illustration that the CPL median is biased for even-sized samples. For the stack in the bottom panel, the median was replaced by the average of the two values in the left third of the field. The background is now continuous as it should be.