The Commissioning of the Arcetri Near-Infrared Camera ARNICA: III. Suggestions for Broadband Data Acquisition and Reduction.

L. Hunt[1], L. Testi[2], S. Borelli[2],
R. Maiolino[2], and G. Moriondo[2]

Arcetri Technical Report, N° 4/1994

Firenze, June 1994

[1] C.A.I.S.M.I.–C.N.R.
[2] Università degli Studi di Firenze

Contents

1 Introduction

The availability of large-format near-infrared (NIR) array detectors has made possibile a host of scientific objectives, too time-consuming or too ambitious to be realized with the previous generation of detectors. While qualitative morphological information about star-forming regions, galactic disks, distant protogalaxies and the like can be readily obtained with such devices, achieving reliable quantitative results can be somewhat more difficult. Obtaining such results with one of the currently available arrays, the NICMOS-3, is the subject of this report.

This report is the third of a series that describes the data analysis and the quantitative results obtained with ARNICA at the TIRGO during the commissioning of the camera (November 1992 – June 1993). We describe here in the third report a series of broadband data acquisition schemes, suitable for point-like and extended sources, and the IRAF procedures available in the new ARNICA data reduction package. We do not pretend to provide an exhaustive reference for NIR image acquisition and reduction, nor to provide procedures for every possibile observing situation. The intention is rather to offer suggestions for efficient image acquisition and reduction, and provide a foundation for future development.

The first part of the report considers how best to compensate for spatial variation in system response at high, intermediate, and low spatial frequencies, in other words, how best to apply the "flat-field" correction. Then, we describe data acquisition methods for point-like and extended sources, followed by a task-by-task outline of the data reduction procedures to be used for each of the specific data acquistion schemes. Finally, we give a typical data reduction sequence, together with some `awk` procedures useful for standard star photometry.

Specialized tasks such as the creation of a bad pixel list or a photometric corrective surface are summarized in the Appendices. Also described in Appendices is how to access, copy, and modify the `arnica` reduction package under IRAF.

2 Flat field correction: sky frame subtraction or division

The term "flat field correction" is intended to describe the procedure by which systematic spatial variation in instrumental throughput is eliminated. Such systematic variation comprises low and intermediate spatial frequencies arising from camera vignetting and variation in the quantum efficiency of the detector, as well as the pixel-to-pixel (high spatial frequency) variation in detector response. After application of the flat field correction, the *flatness* of the corrected image is measured, not by the noise in a single pixel or small region, but rather by the large-scale (low spatial frequency) residual trends that remain even after application of the flat field correction. In the best case, such trends should be comparable to the pixel-to-pixel noise in a small area, as this noise also depends on other factors such as on-chip integration time and number of coadds.

In the NIR, the large amplitude of the background emission necessitates extremely flat images. For example, to reach at a signal-to-noise (S/N) of 1σ a limiting magnitude in the K-band of 21 mag arcsec^{-2} where the sky is typically 12–13 mag arcsec^{-2}, images must be flat to one part in a thousand or 0.1%. Such a constraint is a necessary condition for detection of faint extended emission, and, thus determines the sensitivity of a camera. Of course, point-like source detection relies only on images being at least locally flat, but residual non-flatness makes threshold noise estimates and automatic detection difficult, and complicates sky subtraction.

The question of how best to apply the flat-field correction in the NIR is a delicate one (e.g., McCaughrean 1989). A source of illumination is appropriate to use for a flat field if: 1) there exists a multiplicative relationship with instrument response; and 2) it is spatially uniform across the detector. In a typical source observation, the measured flux in analog-to-digital units (ADUs) can be written as:

$$\begin{aligned} A(x,y) &= [f_1(x,y) + f_2(x,y) + f_3(x,y) + f_4(x,y)]t\eta(x,y)\psi(x,y)G + \\ &\quad +[d(x,y)]tG + O(x,y) \end{aligned} \tag{1}$$

where:

$A(x,y)$	=	measured flux in ADU
$f_1(x,y)$	=	source emission (photon s^{-1})
$f_2(x,y)$	=	background sky emission (photon s^{-1})
$f_3(x,y)$	=	background from warm telescope and optical components (photon s^{-1})
$f_4(x,y)$	=	stray light in camera optics (cool) (photon s^{-1})
$\eta(x,y)$	=	detector quantum efficiency (e$^-$/photon)
$\psi(x,y)$	=	telescope + camera transmission efficiency
t	=	on-chip integration time
G	=	conversion factor in units of ADU/e$^-$
$d(x,y)$	=	detector dark current (e$^-$ s^{-1})
$O(x,y)$	=	instrumental offsets

Background sources of emission multiplied by the instrumental throughput, that is by the product of η and ψ, generally include three components[1]. The atmospheric air glow is given by f_2. The NICMOS-3 detector is sensitive to radiation from 0.9 to 2.5μm, and the emissivity of the atmosphere in this spectral region is due primarily to the transitions between vibrational states of the hydroxyl radical (e.g., Oliva & Origlia 1992). This emission can be orders of magnitude brighter than the objects one wishes to measure, and is the main cause of the differences between optical and NIR observing and reduction techniques.

[1]In principle, f_3 and f_4 should be multiplied by different transmission efficiencies ψ than are f_1 and f_2.

Thermal emission from the telescope and telescope structure is given by f_3, noticeable only for λ longward of 2μm. Since thermal background in the Wien regime increases rapidly with temperature, structures at 300 K ambient temperatures can contribute to the background measured in the standard broadband K filter (2 to 2.4μm). Reduction of ambient thermal emission is the main advantage afforded by the long wavelength-truncated K-filter, K′. Telescopes optimized for infrared observations also reduce the thermal background emission, because the contribution of emissive structures such as Cassegrain holes in the primary mirror, baffles, and mirror cell edges is minimized.

Scattered light inside the camera cryostat is given by f_4, and can also contribute to the observed background. Inadequate internal baffling or highly reflective optical mountings or both can result in internal stray light that shows spatial variations comparable to those in the detector itself (20–30%).

As stated before, the flat-field correction is applied to eliminate the spatial dependence on instrumental throughput. It is evident that this cannot be achieved unless the additive components in Eq. 1, $d(x, y)$ and $O(x, y)$, have first been subtracted off. Nevertheless, even after subtraction we are left with the sum of four terms, only one of which is interesting, namely the source emission.

The tacit assumption made when using an image as a flat field is that the source of illumination is uniform. Although f_3 and f_4 in Eq. 1 are multiplicatively related to the throughput, they almost certainly vary spatially, and must therefore be subtracted off before the image can be used for a flat field correction. This is the justification for the use of "differential flats", or the difference between pairs of images taken at the same integration time of a uniformly illuminated scene and the same scene without illumination (typically the dome). As mentioned above, thermal emission f_3 becomes important for $\lambda \geq 2\mu$m, so, unless a term like f_4 is present, differential flats should be necessary only for the K-band and longer wavelengths. f_3 is small for telescopes like the TIRGO or the UKIRT that are optimized for infrared work. Therefore, at infrared facilities such as these, differential flats in principle are superfluous, although this supposition can and should be checked on an individual basis.

All the above suggests that unless there are terms in the background that vary spatially (i.e., if f_2 does not depend on x and y), after dark current and offset subtraction, sky frames can be used for the flat-field correction. Otherwise, sky frames should first be subtracted so as to eliminate terms like f_3 and f_4 in Eq.1; then differential flats can be used to properly correct for spatial variation of instrumental throughput. Evidently, the key assumption is that the atmospheric airglow in the infrared is spatially uniform over the field-of-view (FOV) of the detector. This may not be always be true for narrow-band work, but seems to generally be the case for broad-band imaging, at least on time scales of a few minutes.

Another consideration that should be mentioned for completeness is that of matching the color of the flat field with that of the source of interest (see McCaughrean 1989). The spatial variation of the detector quantum efficiency may depend on wavelength. Therefore, for photometric work, it can be important to use a source for the flat field that is similar in color to that of the source under investigation.

We have found, however, that for good quality flat fielding (0.1% or better), sky frames are more than adequate. Moreover, as described in the previous report ("II. Astronomical performance"), the difference between sky and source colors does not seem to be the primary source of discrepancies in the photometry.

3 Data acquisition

The observing techniques described here are variations of the method developed by Tyson and collaborators for deep CCD imaging (Tyson 1986 and references therein). In this method, the total exposure time needed for a given signal-to-noise (S/N, for a single pixel) is broken up into a number of background-limited shorter exposures. After each exposure, the telescope position is shifted by a small amount that depends on the typical image dimension, the camera field-of-view, and the size of the seeing disk. The images are reduced with a "master" flat-field image, constructed by combining the images in the stack; the final image is formed by registering the reduced images, and combining them with adaptive modal filters (median filters + clipping).

- **Integration time.** The most important consideration when acquiring science frames is that of ensuring that the images are background limited, and that no source in the field is saturated. For weak sources, background-limited performance is achieved when the sky counts $\gg$ 125 ADU, since the square of the read noise is 2500 e^- with a conversion factor of 20 e^-/ADU (see Hunt, Maiolino, & Moriondo 1994a). On-detector integration times of 3 s or longer should guarantee background-limited performance in H and K at the TIRGO. In the J-band, integration times of at least 10–12 s may be necessary, as, in our experience, shorter times may not be long enough to guarantee background-limited performance [2].

- **Number of coadds.** The next consideration is how many fundamental integrations should be combined together before moving the telescope, that is to say how many "coadds" are required for each telescope position. On the basis of observing runs with different NIR cameras at different sites (TIRGO, WHT, ESO, UKIRT), the shorter the time spent with the telescope in one position the better. On the other hand, one needs to strike a compromise between observing efficiency and spatial changes in the NIR sky background, so that the sky is adequately sampled and the dead time spent in positioning the telescope and starting a group of measurements is small relative to the time spent integrating. On a very good night, the telescope can remain as long as 4–5 m in one position without compromising final image quality; however, the quality of the night is often assessed a posteriori and, conservatively speaking, the telescope should remain in one position no longer than 100–120 s. Furthermore, for a given total integration time, as long as images are background

[2] This constraint is clearly important in narrow-band imaging.

limited it is more advantageous to combine together large numbers of repetitions rather than small numbers of high signal-to-noise coadds. This point is described in more detail in the following.

- **Where to move the telescope.** After performing a given of number of coadds at a fixed on-chip integration time, the telescope must be moved. Where to move the telescope for efficient acquisition of science frames depends on the nature of the source. If extended emission or nebulosity is present in the FOV of the camera, for sky frame acquisition (either for the flat-field or to subtract) it is necessary to move the telescope away from the source position to a region where the sky background is spatially uniform (see previous section). If, instead, only point-like sources are present (and the field is not overcrowded), then the telescope can be moved only slightly, keeping the source of interest in the camera field-of-view.

- **Number of repetitions.** The final requested signal-to-noise (S/N, for a single pixel) dictates how many times to move the telescope. For each individual or coadded flat-fielded frame, camera performance scales with S/N and total time spent on-source, and, if the individual images (and coadded images) are background limited, then conventional formulae can be used to calculate final S/N for a given background. However, the noise introduced by the flat-field correction should also be taken into account (see Hunt et al. 1994b); typically, when the number of source frames and the number of frames in the flat field are the same, the flat-field correction introduces an increase in the noise per pixel of a factor of $\sqrt{2}$.

  When the frames are "mosaiced" together to produce the final source frame, camera performance would scale with S/N and integration time if the frames were coadded (combined with a simple arithmetic average). Instead, because of inevitable stars in the sky frames and possible other spurious artifacts, frames are better combined with a median filter algorithm coupled with rigorous clipping (see next section). Experience has shown that such a method of combination results in scaling with an *effective* integration time about 2/3 of the elapsed integration time (see also Carter 1993), although the scaling changes with number of frames. The combination of as many frames as possible is advantageous, as a large number of frames (≥ 10) reduces the noise associated with the filtering.

3.1 Point-like sources

Point-like sources are more efficient to observe than are extended sources as sky frames can be acquired while keeping the sources of interest in the field-of-view of the camera. Representative sky frames can then be obtained by combining some subset of displaced source frames with appropriate filtering and clipping algorithms.

The telescope should first be pointed so that the nominal source position is somewhere near the center of the array. The precise center (i.e., pixels 128,129) should be avoided as the quadrant read-out scheme (see Hunt, Maiolino, & Moriondo

1994a) sometimes makes the central row or column or both unusable, even after the flat-field correction. Then after acquisition of one group of measurements, the telescope can be pointed to a nearby position that displaces the sources of interest, but keeps them on the array. The amount to move the telescope depends on the dimensions of the seeing disk. To avoid overlap of the broad low-intensity wings of the point-spread function (PSF), it is best to move the telescope something like at least 10 times the width of the PSF.

Successive movements of the telescope should be irregular, so that the flat-fielded images do not contain artifacts created by regular repetition. Moving the telescope in such an irregular way is sometimes called "dithering", and we will refer to this term in the following sections. If the science images are to be reduced by the procedures described in §4, the only patterns to avoid when acquiring point-source images are regular grids. For each position of the telescope, the data reduction procedures combine the remaining images to form the sky frame for flat-fielding. If the positions of the telescope are evenly spaced, then the resulting flat field will contain regularly-spaced features that will not properly be eliminated when the science frames are "mosaiced" together. Superposition of sources with residual "holes"and improperly flattened images or both may cause subsequent problems with photometry. On the other hand, if one chooses to perform photometry on each flat-fielded image separately without combining them into a single mosaic, then regular grid patterns can be used to position the telescope.

3.2 Extended sources or crowded fields

As mentioned above, the main difference between point-like and crowded-field or extended-source acquisition is the telescope position for the acquisition of sky frames. When the source is extended or the field crowded, it is necessary to move the telescope to an empty area of the sky to ensure that the illumination is as spatially uniform as possible. How much to move the telescope depends on the size of the source; typically the field-of-view of the array (~ 4.3 arcmin) is sufficient.

As for point-sources, the telescope should not be moved to the same exact sky positions, but rather "dithered" to one or more sky positions. An ideal movement sequence approximates that performed by single-detector beam-switching, that is sky to one side, source, sky to the other side, source, and so on. A sequence like this theoretically eliminates linear trends in sky gradients. If a given region of sky is chosen to be empty (as in the case of a very large extended source), then the same considerations apply as described above; namely, the telescope should be dithered about a nominal empty sky region with an amplitude of at least 10 times the width of the PSF.

To further eliminate spurious artifacts in the flat-fielded images, the telescope should preferably not be moved back to the same exact source position, but rather dithered around some nominal placement. The dithering amplitude may be smaller than that of the sky (a few arcsecs) as typical extended sources occupy a large fraction of the camera FOV; here, the idea is to make as irregular as possible the relative source–sky placement and to eliminate small-scale structures like bad pixels that may not be compensated for by a bad pixel mask.

3.3 Standard stars

Standard star photometry is facilitated by using the same pattern for each set of standard star measurements. In this way, the virtual aperture used for photometry can be centered semi-automatically, thus avoiding the laborious and time-consuming procedure of centering the apertures by hand for each standard star.

We recommend at least three positions in different areas of the array to at least coarsely sample photometric spatial variations; better still is four positions, or one in each quadrant. During commissioning, we used a five-position sequence called `pt_std3` that starts with the star in a nominal center position (displaced from the array center by 25 arcsec or so), then places the star in the center of each quadrant.

4 The ARNICA reduction package: task description

In the course of ARNICA's commissioning and performance assessment, we have developed a set of IRAF procedures to reduce the calibration and science frames acquired in the generic modes described above. These procedures are organized into an external IRAF package called `arnica`. (See Appendix C for system requirements and how to retrieve and install the package outside Arcetri). Here we review the package organization and explain the use of the individual tasks. An example of a complete reduction session will be given in the following section. We stress again that these IRAF procedures are provided for the convenience of ARNICA observers, and are not intended to be the end-all of NIR image reduction.

Once the package has been installed as an external IRAF package, the tasks can be accessed as for all the external packages. Upon entering IRAF, the list of the installed packages should appear; the following list appears at Arcetri:

```
apropos      hst_pipeline.  personal.    tables.
arcetri.     images.        plot.        utilities.
arnica.      language.      proto.       vol.
color.       lists.         softools.    xray.
dataio.      local.         spptools.
dbms.        noao.          stsdas.
euv.         obsolete.      system.
```

To load the package type:

```
cl> arnica
```

and the parameters of the standard observatory and the list of the available tasks will appear[3]:

```
# Observatory parameters for Gornergrat Infrared Telescope
        observatory = tirgo
        timezone = -1
        altitude = 3135
        latitude = 45 : 59 : 04.0
        longitude = -7 : 47 : 30.0
        name = 'Gornergrat Infrared Telescope'
```

```
airmass_  gal2    mk_mean   photfit2  std    surface_
clean_    gal3    mosaic    photfit3  stda   tia
desincr   gal3d   phot_c    phots     stdad  txa
gal1      infits  photfit1  sta       stdas
```

The package is distributed with a short on–line help database. Help for individual tasks, if available, can be accessed by typing `help` *taskname*; as usual, to view the source file of each task simply type `help` *taskname* `option=source`. Typing `help arnica` at the prompt will provide a short reference on each task (this should work even if the package is not loaded):

```
airmass_  --  Corrects ARNICA image header (for dates < Oct 1993)
  clean_  --  Cleans bad pixels using a mask and spatial median
    gal1  --  Reduces ON-OFF...  observations, flat from all OFFs
    gal2  --  Reduces ON-OFF...  observations, flat from nearest OFFs
    gal3  --  Reduces OFF-ON-OFF...  observations, flat from nearest OFFs
   gal3d  --  Same as gal3, but with sky subtraction and differential flat
  infits  --  Converts FITS files into IRAF format, optionally deleting original
 mk_mean  --  Combines images with arithmetic mean
  mosaic  --  Aligns and combines offset images (median with ccdclip)
  phot_c  --  Performs photometry of standard star image sequences
     sta  --  Performs statistics for a list of images
     std  --  Reduces "dithered" observations, flat from all images
    stda  --  Reduces "dithered" observations, flat from all images but one
   stdad  --  Same as stda, but with sky subtraction and differential flat
   stdas  --  Same as stda, but for long scans (plus-or-minus 5 only)
surface_  --  Subtracts image median and divides by photometric correction frame
     tia  --  Corrects integration time in ARNICA header (for dates < Oct 1993)
     txa  --  Extracts relevant information from a list of photometry files
```

As usual the parameters of the tasks can be given directly when typing the command at the prompt, or can be set using the IRAF parameter editor (`epar` *taskname*).

[3]The file `obsdb.dat` in the `arnica$` directory contains the settings for various observatories; to change the default type: `observatory set "`*obsname*`"`.

A general philosophy of the `arnica` reduction package is to name the reduced images (before combining and registering) by adding a prefix to the original name. The prefix changes according to the task and affords an implicit record of which task was used for reduction. The only assumption made by the tasks in the package is that the images are named with a prefix followed by a sequential numbering scheme, and that all the images with sequence numbers within the range of sequence numbers in the task arguments are present. In addition, the extended-source tasks assume that source and sky positions alternate.

All the source reduction tasks correct for bad pixels with pre-existing bad pixel lists using linear interpolation (`proto.fixpix`). The bad pixel lists can be found in `arnica$badpix/bpf.fix.92dec`, `arnica$badpix/bpf.fix.93mar`, and `arnica$badpix/bpf.fix.93jun` for December 1992, March, and June 1993, respectively.

The available tasks can be divided into five categories depending on their purpose: general utilities, point–source reduction, extended–source reduction, application of photometric corrective surface, and standard star photometry. In the following, we will describe and explain the use of each task; the taskname is in `tt` and the parameters in *italics*. Hidden (optional) parameters are given by the syntax *parameter=value*, with the default shown as *value*. We apologize for the tasknames, as their names represent only historical development.

4.1 Utilities

The tasks that can be grouped in this category are: `airmass_`, `clean_`, `infits`, `mk_mean`, `mosaic`, `sta` and `tia`.

- `airmass_` *sfrm*

  `tia`

  The tasks `airmass_` and `tia` correct the headers of ARNICA images acquired before October 1993. Before this date, the `AIRMASS` and `TIME-INT` keyword in the header are incorrect and are better corrected before reduction and analysis.

  `airmass_` requires as an argument the name of the image to be corrected; for example to correct the header of *im_name* just type:

  `airmass_` *im_name*
  The task `tia` does not have parameters, as the names of the images to be corrected are read from a file called `tint.lis` which contains an image name per line:

  im_name_1

  im_name_2

  .

.

im_name_N

The task is run by simply typing the taskname at the prompt. Note that the image names in the file must contain the complete path to the real files location if they are not located in the current directory.

- `clean_` *sfrm*

  `clean_` *sfrm* corrects for bad pixels by applying a bad pixel mask together with a spatial median filter. Unfortunately, the NICMOS 3 arrays tend to present clusters of bad pixels as well as isolated ones, so that correction by spatial median substitution works less well than interpolation techniques (see IRAF command `fixpix`).

- `infits` *list sprefix in* `erase-`

  `infits` is logically the first task to be used when starting reduction. Once the FITS files reside on disk, start IRAF and change to the directory where the reduction will be made. Suppose that the FITS files are in the directory /*fits_dir*/, and create a file (for example *nfits*) with all the FITS names, by typing:

  `files` /*fits_dir*/`*.fts` > *nfits*
  To run `infits` to convert FITS files into IRAF images, type:

  `infits` *nfits prefix start_seq* > `infits.out`
  where *prefix* is the prefix of the IRAF image names and *start_seq* is the first number of the sequence that will be written. The log of the task will be written in the file `infits.out`. For example, if the file containing the FITS names is `fts.lis`, the prefix is `dat`, and the first sequence number is 12, the command should be:

  `infits fts.lis dat 12 > infits.out`
  The FITS images will be converted into IRAF format images `dat0012, dat0013` ... and so on until the last FITS file. By default the task will not erase the FITS files; if you want to delete them as they are converted, just add `erase+` to the parameters (before the pipe, of course).

- `mk_mean` *outfrm sprefix n1 n2*

  The task `mk_mean` can be used to coadd with a simple arithmetic average a set of (registered) frames. The parameters are the output image name, the prefix of the input images, and the first and last sequence number of the input image set. To coadd the frames `dat0003`, `dat0004`, `dat0005` and `dat0006` to produce the image `mean` just type:

`mk_mean mean dat 3 6`

This task was originally designed to be used with the data acquired before February 1993 because the software installed before that did not support on–line coadding during integration. With the new software, this task is probably not very useful, because on–line coadding is highly encouraged for the reduction of dead times, for disk space constraints, and for data reduction fatigue.

- `mosaic` *imlist outima* `box= zs= zr= z1= z2= comb= erase= lthresh= hthresh=`

  This task registers and combines a set of offset flat-fielded images, using the iraf task `noao.imred.ccdred.combine` with a median filter, ccdclipping (with appropriate ARNICA parameters), and a relative (dc) offset equal to the image median. The images are registered using shifts calculated with a set of fiducial objects (typically stars) present in all the images; the stars are selected with the mouse on the display, hence the task requires the use of a pictorial terminal running `suntools` or `X11/openwindows` window manager interfaces. *imlist* is the name of a file containing the names of the images to be combined and *outima* is the output image. Image sections can be specified; this is very useful when the use of the entire field-of-view is undesirable as the outermost parts of each image can be excluded. The `box` parameter is used to set the width (in pixels) of the centering box used for the calculation of the centroid of selected stars. `zs, zr, z1,` and `z2` are the usual `zscale, zrange, z1` and `z2` parameters of the task `display` used by `mosaic` to display the images on the tv device. `comb` and `erase` are boolean parameters; the first is used to force the task to actually combine the images in the output image instead of to simply calculate the offsets between the images, and the second is used to force the task to delete the file (called `offsets.`*imlist*) which contains the calculated offsets between the images. `lthresh` and `hthresh` are the minimum and maximum good data values to be considered.

  For example to make a final image called `image` by combining the sections $[6:251, 6:251]$ of a set of images `bfad_0025 ... bfad_0035` without erasing the offset (relative shifts) file, edit a file (for example `img.lis`) which an image section per line:

  `bfad_0025[6:251,6:251]`

  .

  .

  `bfad_0035[6:251,6:251]`

  Then start the task execution typing:

  `mosaic img.lis image comb+ erase-`

  The task will load the first image on the image display, and will switch to the image cursor mode waiting for input (assuming that the display tool has been opened and the display is correctly set to the terminal in use). Move the cursor to the position of the first of the selected stars and press `p`; the coordinates of the star will appear on the terminal. Move to the next star and repeat the operation. After the last star, press `q` to go to the next image. When the second image is displayed, repeat the star selection (using the same stars in the same order), and proceed until stars are selected from the last image. The task will then display the calculated shifts, and the images are combined to produce the output image. The value assigned to pixel not covered by the mosaic is −10000. The calculated offsets are conserved in the file `offsets.img.lis`.

- `sta`

  `sta` can be used to extract statistics for a list of images. First edit a file called `stat.lis` which contains the list of images for the statistics; then run `sta`. In this way, relevant information on the image will be printed on the terminal.
  For each image a line is printed which contains:
  FITS number, `IMTITLE` header parameter, filter, integration time, IRAF image name, mean, median, standard deviation, minimum value and maximum value.
  Of course, it is much more useful to redirect the output into a text file, `stat.out`, by, for example, typing:

  `sta > stat.out`.

4.2 Point–like sources

As explained before, in order to maximize the on–source integration time point–like sources can be observed using a "dithering" technique in which the source(s) are always kept in the FOV of the camera. The tasks used to reduce this kind of acquisition are called: `std, stda` and `stdad`. Each task reduces each image in a complete measurement sequence by subtracting a sky image, if appropriate, then dividing by the flat field image; the tasks differ in the method used to calculate the sky image and the flat field image. In particular, `std` and `stda` construct the sky frame from a set of source frames and use it for the flat field image, while `stdad` requires an independently-obtained differential flat-field image, and subtracts the sky frame. In all three tasks, the sky frame is constructed using the iraf task `noao.imred.ccdred.combine` with a median filter, percentile clipping, and scaling to the image median.

- `std` *sprefix n1 n2* `bplist=`

  `stda` *sprefix n1 n2* `bplist= imtype=`

The difference between these two tasks is that `std` uses all the images to construct the flat field image by which all images are then divided; instead, for each image `stda` calculates the flat field from the remaining images in the sequence. *sprefix* is the image name prefix; *n1* and *n2* are the sequence numbers of the first and last images of the mosaic. `bplist` is the name of the bad pixel file (with the complete path), and `imtype` is the image format code `hhh` or `imh`. (Set the correct one otherwise the task will stop with an error). The prefix of the name of the reduced images is `bf` if using `std`, and and `bfa` if using `stda`. For example, to reduce with `stda` the sequence starting with the frame `dat0006`, ending with the frame `dat00014`, using the bad pixel list of the December 1992 run, type (if you are using the `hhh` standard):

```
stda dat 6 14 bplist=arnica$badpix/bpf.fix.92dec \
>>> imtype=hhh > stda.out &
```

Since the task will require some time and will produce a lot of output on the terminal, the best thing to do is to redirect the output in a file and run the task in background (in the example the log file is called `stda.out`). The task will produce a set of reduced images called `bfadat0006 ... bfadat0014`, that can then be registered and combined using the `mosaic` task.

- `stdad` *sprefix n1 n2 dflat* `bplist= imtype=`

  As mentioned before, this task is used when differential flats are necessary. All the parameters have the same meaning as those explained above. The additional parameter, *dflat*, is the complete path name of the flat field frame to be used. The prefix of the name of the reduced images is `bfsa`. For each image, as in `stda`, `stdad` calculates the sky frame from the remaining images in the sequence; the sky frame is subtracted before division by the differential flat field.

4.3 Extended sources

As discussed in the previous section, extended sources are better observed using a "beam–switching" technique, that is going "OFF" to a blank area of the sky for sky measurement. We have used basically two observing sequences: an ON–OFF–ON–...–OFF cycle, and an OFF–ON–...–OFF cycle. The advantage of the second is that each "source" frame can be flat fielded using the two sky frames taken sequentially before and after, hence short-term background variation both in structure and in amplitude is better compensated for. "Beam–switching" or OFF-ON observations can be reduced using one of the following tasks: `gal1, gal2, gal3` and `gal3d`.

- `gal1` *sprefix n1 n2* `bplist=`

  `gal2` *sprefix n1 n2* `bplist= imtype=`

These tasks are designed for ON–OFF sequences that begin with an ON (ON–OFF–ON–...–OFF cycles); the parameters have the same meaning as for the `std`–like tasks. The difference between these two tasks is that `gal1` uses *all* the off–source frames to construct the flat-field image that is divided into all on–source frames; `gal2` calculates the flat-field image using only the two off–source frames closest to each on–source frame. For the first ON (as for the second ON), the first and second OFF are used. The prefix of the name of the reduced images is `bf1` if using `gal1`, and `bf2` if using `gal2`. Note that since only the on–source frames are reduced, the `gal2` reduced frames will have the names: `bf2`*sprefix(n1)*,`bf2`*sprefix(n1+2)* ... `bf2`*sprefix(n2−1)*.

- `gal3` *sprefix n1 n2* `bplist= imtype=`

  `gal3d` *sprefix n1 n2 dflat* `bplist= imtype=`

  These tasks are designed for ON–OFF sequences that begin with an OFF (OFF–ON–OFF–...–OFF cycles). As for `gal2`, `gal3` uses only the two off–source frames closest to each on–source frame for the sky frame, and the sky frame is used for the flat field correction. The prefix of the name of the reduced images is `bf3`, and only the source frames are reduced: `bf3`*sprefix(n1+1)* ... `bf3`*sprefix(n2−1)*.

  `gal3d`, like `stdad`, uses a differential flat field image specified with the parameter `dflat`, and subtracts the sky frame before division by the flat field. The prefix of the name of the reduced images is `bfs3`, and only the source frames are reduced: `bfs3`*sprefix(n1+1)* ... `bfs3`*sprefix(n2−1)*.

4.4 Photometric corrective surfaces

As described in the previous report ("II. Astronomical performance"), for images acquired at the TIRGO before 1994, spatial variation in the photometry had to be corrected by a corrective surface. This corrective surface is applied as a second flat-field correction, dividing reduced images by it before performing photometry. Application of such a correction results in more accurate point- and extended- source photometry, but the flatness of the image is degraded.

We have developed specific observing sequences and reduction procedures that have been used to calculate the corrective surface. The IRAF tasks for this have been included in the `arnica` package to ensure that every observer can build his/her own surface, if necessary. The distributed package contains the surfaces for the December 1992 and February 1993 runs, in the directories `arnica$surfaces/92dec` and `arnica$surfaces/93feb`. The tasks needed to construct the surface and apply the correction are: `photfit1, photfit2, photfit3, phots, stdas` and `surface_`. The first five are used to calculate the surface and are described in Appendix B; here we describe only the last one that is used to apply the correction. This photometric correction is the last step of data reduction *before* combining individual reduced images into a final "mosaic", applied *after* the tasks described in the previous two sections.

- `surface_` *sfrm ssurf*

  sfrm is the name of the frame to be corrected and *ssurf* is the complete path name of the corrective surface to be used. The task first subtracts the constant defined by the image median, then divides by the corrective surface. The prefix added by the task is `cs`, so that the output image is called `cs`*sfrm*. For example, to correct an image called `bfadat0012` with the K–band surface of the February 1993 run, type:

  `surface_ bfadat0012 arnica$surfaces/93feb/k.d.fit3_1.imh`
  The corrected frame will be called `csbfadat0012`.

NOTE: The corrective surface depends strongly on the waveband, and on the coupling between camera and telescope (as well as on the camera optics as described elsewhere). Each time the camera is dismounted and mounted again at the telescope, the surface must be recalculated and its effect assessed. Do not use the wrong surface; otherwise you will obtain meaningless data.

4.5 Standard star photometry

We have found that the simplest way to "standardize" observations and simplify the photometric calibration was to always use the same telescope sequence for standard star observations. If all the standard stars have been obtained using the same sequence with approximately the same relative offsets and starting point, then the virtual aperture can be centered semi–automatically, and manual intervention is minimized. Even if the offsets are the same but the starting point is different, different coordinate files for the first guess of the virtual aperture position can still be easily created. The photometry task `phot_c` and the `awk` procedures described in the following section assume that the standard stars have been observed using a five–position sequence, as only five positions are processed.
NOTE: The following will work only if you are using the `imh` image format, as, due to some uncorrected IRAF bug, the `hhh` format does not conserve the `IMTITLE` header parameter that is used by the following tasks to discriminate between the sequence positions.

- `phot_c` *sprefix n1 n2 smag* `coo=`

  As before, *sprefix* is the image name prefix, and *n1* and *n2* are the first and last sequence numbers. *smag* is the magnitude of the standard star (in the same band as the images, of course), and *coo* is the path of the coordinate files. For each frame, the task will produce a standard `apphot/daophot` output file containing the zero point calculated in a set of six apertures (6,7,8,9,10 and 12 pixel radii). The coordinate files are five files called `coo0, coo1, coo2, coo3` and `coo4`. Each of them contains the approximate position of the star in the corresponding sequential order: `coo0` is the star position in the first frame, `coo1` in the second, and so on.

- `txa`

  This task extracts relevant information from the redundant files produced by `phot_c`. First, edit a list called `phot.lis` which contains the files output by all the calls to `phot_c` (for example by typing `files *.mag.1 > phot.lis`). Then, run `txa`, redirecting the output into a file. In this file, six lines will be written for each image, one per aperture. Each line contains frame name, FITS number, `IMTITLE` header parameter, filter, x and y position of the virtual aperture, aperture radius, airmass, zero point, and error code. (The zero points will appear negative.) This file can then be processed by the `awk` procedures described in the following section.

5 Typical reduction sequence

An image reduction session for a single observing night's data can be conveniently organized into various steps. Since each of the image reduction tasks reduces one object sequence, the flat-fielding (and sky frame subtraction, if necessary) phase can be grouped into one step; an IRAF procedure consisting of a single `.cl` file with calls to `stda`, `gal2`, `gal3`, etc. can be created for this. (See Shames & Tody 1986 for a description of the IRAF command language and the use of `.cl` files.) If necessary, the photometric corrective surface can be then applied to each flat-fielded image in a single `.cl` file with calls to `surface_`. Finally, the reduced images can be mosaiced together with `mosaic`, and standard star photometry can be performed with a single `.cl` file that contains repeated calls to `phot_c`. Each of these steps is outlined below, together with a description of when the utilities tasks may be useful. The steps enclosed in square brackets should not be necessary for ARNICA data acquired after 1993.

1. **Convert FITS images to IRAF format.** If disk space is a problem or if an arbitrary sequential numbering scheme is needed for one reason or another (because of previous aborts, for example), then `infits` can be used to convert FITS images to IRAF format. Otherwise, `rfits` suffices.

2. **Perform statistics.** Make `stat.lis`, then run `sta` and pipe output into a file. Output is more aesthetically pleasing when the file is filtered with an `awk` procedure called `stat.awk` (in the `awk` subdirectory of the `arnica` package). This information can be used merely to organize observations, or determine (at least for data taken between December 1993 and February 1994) which images are unusable.

3. [**Make means.**] For images acquired before 13/14 Feb 93, the individual coadds must first be averaged together. `mk_mean` is useful for this, and the repeated calls to this task can be grouped together into one command file. For images acquired after this, this step can be eliminated if the individual coadds have been averaged by the data acquisition software.

4. **Reduce images.** Make and execute a `.cl` file which contains, for each object sequence, calls to tasks in the `std` category and in the `gal` category. Understanding which IRAF image contains which object observation is facilitated by the `sta` output as outlined above, or by the `infits` output.

5. [**Apply photometric corrective surface.**] Make and execute a `.cl` file which contains, for each frame, calls to `surface_`.

6. **Perform photometric calibration.** First define the files which contain the approximate positions of the standard stars for which zero points are needed (the `coo` files). Then make and execute a `.cl` file which contains, for each standard star sequence, calls to `phot_c`.

7. **Extract photometry from `.mag` files.** Make `phot.lis`, then run `txa` and pipe output into a file.

8. **Make final source images.** Register and combine each source sequence with `mosaic`.

NOTE: Steps 6+7 and 8 are independent, not sequential.

5.1 AWK utilities for standard star photometry

Two `awk` procedures, included with the `arnica` package, facilitate photometric calibration. (For more about `awk`, see the UNIX manual, or Aho, Kernighan, & Weinberger 1988.) The first procedure, `phot_sep.awk`, applied to the file (piped) output of `txa` selects photometry from a given aperture. For example, to apply `phot_sep.awk` to a (piped) output file of `txa` called `phot.out`, type:

```
nawk -f phot_sep.awk phot.out > phot.out.r10
```

The file `phot.out.r10` contains only the photometry (or zero points) for a given aperture. The procedure can be edited to define which aperture (of those calculated by `phot_c` is to be selected.

The second procedure, `phot_avg.awk`, is applied to the output of the first procedure, `phot.out.r10` and produces, for each of the J, H, and K filters, the nightly global averages, the nightly averages as a function of position, and the position averages. Peak-to-peak values are also provided. For example, to apply `phot_avg.awk` to `phot.out.r10`, type:

```
nawk -f phot_avg.awk phot.out.r10 > phot.out.r10.avg
```

The file `phot.out.r10.avg` contains the nightly (over the file) averages as mentioned above, organized by filter band. As mentioned before, the procedure assumes five positions; however, it is straightforward to alter this by editing the procedure add or subtract positions. The filters for which the averages are calculated can also be changed by simply editing the procedure.

6 Conclusions and disclaimers

As stated in the Introduction, this report was not written to provide an exhaustive reference on NIR image acquisition and reduction. We hope that users will copy the sources of the IRAF scripts and alter them to suit individual needs. Positive feedback or constructive criticism is welcome (!), but we cannot promise to answer to each and every (inevitable) complaint, as the package was constructed only for internal use.

7 Acknowledgements

We would like to thank C. Baffa and R. Stanga for testing the procedures and installation, R. Baglioni for making life easy with IRAF and installation procedures, and R. Bandiera for a careful reading of a first draft of the report and insightful comments.

References

Aho, A.V., Kernighan, B.W., and Weinberger, P.J. 1988, *The AWK programming language*, Addison-Wesley Publ. Co., Reading, Massachusetts

Carter, D. 1993, Gemini, No. 39, 14

Hunt, L.K., Maiolino, R., and Moriondo, G. 1994a, Arcetri Technical Report, 2/1994.

Hunt, L.K., Maiolino, R., Moriondo, G., and Testi, L. 1994b, Arcetri Technical Report, 3/1994.

McCaughrean, M.J. 1989, in Proceedings of the Third Infrared Detector Technology Workshop, ed. C. R. McCreight, NASA Technical Memorandum No. 102209, p.201

Oliva, E. and Origlia, L. 1992, A&A, 254, 466

Shames, P.M. and Tody, D. 1986, *A User's Introduction to the IRAF Command Language*

Tyson, J.A. 1986, J. Opt. Soc. Am. A, 3(12), 2131

A Construction of a bad pixel mask

We outline here one method for the construction of a bad pixel mask or list. Other methods include selection from dark current frames or from fitted coefficients of a series of frames (see Hunt et al. 1994a); however, we found that this method is more reliable and produces more consistent results.

1. **Construct a high signal-to-noise image.** Combine (with `scale=`*median*) a number of frames of a uniformly illuminated scene to make a low-noise image. Check noise of combined image and compare it with the noise in one or a few frames; if noise in combined image is greater, use fewer frames and make sure they have similar count levels. The count levels in the various images should be roughly equal, as otherwise non-linearity will influence the outlying pixels.

2. **Smooth and subtract smoothed image from original.** Smooth with (typical cluster size) 5×5 box (IRAF task `fmedian` is useful for this), and subtract smoothed frame from the original.

3. **Make histogram of difference and perform statistics.** The histogram (with IRAF `imhist`) should be roughly symmetric, centered about zero. The mode and median should be roughly zero as well, but the mean may be slightly negative. Statistics (mean, mode, median, standard deviation) calculated for those pixels inside roughly the interval [-750,550] (e.g., [-800,500], [-900,600]) should give mode and median very close to zero, and a slightly negative mean. The exact interval can be determined from the histogram (see Hunt et al. 1994a), and should correspond approximately to the full-width zero-intensity of the peak.

4. **List the pixels outside the interval.** Use `stsdas.toolbox.imgtools.pixlocate` to list the pixels *outside* the same interval as determined above. The output can be piped into a file, for example called `bpf.dat`.

5. **Convert to a mask or a region list.** The output file from `pixlocate`, `bpf.dat`, can be converted into a bad pixel mask (image) with `imred.ccdred.badpixelimage` to be used with `arnica.clean_`. It probably would be advantageous to convert it with `awk` procedure `mask.awk` to a region list for input to `fixpix`. For optimal correction, contiguous regions for `fixpix` should be derived from the list of individual bad pixels; we have done this manually (by eye) and welcome suggestions for automation.

B Construction of a photometric corrective surface

We outline here the method for the construction of a surface that corrects for spatial gradients in photometric response. Such gradients have been drastically reduced with the enlargement of the pupil plane cold stop (February 1994), so that use of a corrective surface after that date should not be necessary. Corrective surfaces have been calculated for the December 1992–January 1993, and February 1993 commissioning runs, and for the February 1994 observing campaign; in addition, they have been calculated for the December 1993 observing run at the WHT.

The reduction technique outlined below relies on observations consisting of a scan sequence (e.g., with "mosaic" file `scan4.txt`) of an 8×8 grid in which a standard star is placed in 64 different positions on the array, roughly separated (at the TIRGO) by 32 arcsec. After each row (or column) of 8 positions, the star should be returned to a fiducial position to check the sky transparency variations.

1. **If necessary, rearrange sequence numbers of scan frames.** In the early days of commissioning, ∼7% of the frames were desynchonized, or characterized by horizontal striations. These can be eliminated with the special task (in the `arnica` package) `desincr`.

2. **Flat-field the individual images.** Use `stdas` (a variation of `stda` with the same syntax) to flat-field the images (presumably 64 + fiducial positions) in the scan sequence.

3. **Check for bad pixel contamination.** Although the task `stdas` corrects automatically for bad pixels with a given bad pixel list, it may be necessary to correct manually some of the frames for bad pixels. The automatic procedure may not adequately correct for those sporadically bad pixels that fall inconveniently at the same position as the star.

4. **Perform photometry.** Use `phots` to perform photometry; the syntax is the same as that of `phot_c`, but the photometry is performed with an 8-pixel radius, a background annulus with inner and outer radius, respectively, of 12 and 20 pixels, and approximate star centers taken from `coo*` files. The output file `phot.out` contains the x- and y-coordinates of the stars' centers and -2.5 log (*counts*). Delete empty lines and "INDEF" from `phot.out` before proceeding.

5. **Create an 8×8 image of normalized flux ratios.** Use `photfit1` to create an image in which each pixel corresponds to the measured flux of the source in that grid position.

   `photfit1 phot.out` *prefix*

`phot.out` is the name of the (edited) file output from `phots`, and *prefix* is the prefix of the output images. `photfit1` creates two output images: *prefix*`.8` with the measured flux as a function of position, and *prefix*`.8n` with the measured flux as a function of position, normalized to the scan mean. `photfit1` also creates an `ascii` file *prefix*`.8b` which contains the coordinates of the pixels where photometry is missing (either because of desynchronized images, or because the star is at the edge of the array, or because of bad pixels within the photometry radius).

6. **Construct a 256×256 image of normalized flux ratios.** Use `photfit2` to first linearly interpolate over regions where data are missing, then to create a 256×256 magnified image.

   `photfit2` *prefix*`.8b` *prefix #-reduction*

   prefix`.8b` is the *ascii* file output from `photfit1`. Usually, this can more easily be created by hand; the format is that for `fixpix` (i.e., four columns defining rectangular regions, with starting and ending x-coordinates in the first two columns and starting and ending y-coordinates in the last two columns). *prefix* is that of the input (used in `photfit1` and output images, and *#-reduction* is a sequence number that can be incremented every time you change the input file with the missing data. The task first linearly interpolates regions where data are missing, creating an 8×8 output image called *prefix*`.8f_`*#-reduction*. This image is then expanded, via bilinear interpolation performed with IRAF `imlintran`, to a 256×256 image called *prefix*`.exp_`*#-reduction*.

7. **Perform polynomial fit to create corrective surface.** Use `photfit3` to perform a fit to the 256×256 image output by `photfit2`.

   `photfit3` *prefix #-reduction order*

   prefix is (as before) the prefix of the input and output images; *#-reduction* is (as before) the sequential number for reduction iterations; and *order* is the order of the Legendre polynomials to be used for the fit. `photfit3` performs a two-dimensional polynomial fit with Legendre polynomials to the 256×256 image *prefix*`.exp_`*#-reduction*, output by `photfit2`; the output image is called *prefix*`.fit`*order*`_`*#-reduction*.

C System requirements, retrieval, and installation of ARNICA package

Here we review system requirements, and the retrieval and installation procedures necessary to install the `arnica` package on your host computer. Even though the procedures should not be system dependent, the package has been tested only on SUN workstations running SunOS 4.1.1, and all the following will assume that you are using a UNIX system. Refer to your local IRAF manager if you are working with a different operating system. For any questions or problems with installation, refer to Leonardo Testi (Tel. +39–55-2752220, e–mail `testi@arcetri.astro.it`).

C.1 System requirements

The package has been developed under the IRAF 2.10 environment; hence, even if it can probably run under older IRAF versions, we strongly recommend updating your IRAF version (IRAF is distributed by the National Optical Astronomical Observatory). All the "common use" tasks require only the standard IRAF packages and the `noao` external package, but the tasks used to calculate the bad pixel mask and the corrective surface (see Appendix A and B) need also the `xray` and `stsdas_1.3.1` packages. The compiled package requires about 2.5 Mb of disk space.

C.2 Retrieval and installation

The sources of the package can be retrieved via anonymous ftp on the internet node `sisifo.arcetri.astro.it` (150.217.20.1) in the directory `/pub/arnica/` or using `WorldWideWeb` at the Arcetri server. If you are using `ftp` then type:

```
% ftp 150.217.20.1
login: anonymous
password: (your complete e–mail address)
cd /pub/arnica/
get readme.arnica
binary
get arnica.tar.Z
bye
% ls -ls *arnica*
1816 -rw-r--r-- 1 iraf 1845669 May 17 18:09 arnica.tar.Z
4 -rw-r--r-- 1 iraf 3106 May 17 18:12 readme.arnica
% uncompress arnica.tar
```

The file `readme.arnica` contains the same installation instructions outlined here.

1. Create a directory outside the IRAF tree where the package will be compiled. For example, at Arcetri this directory is called `/home/im3/arnica.iraf/`. Ask your system and IRAF manager to create the appropriate directory on your system. This directory must be owned by the IRAF account.

2. Log in as `iraf` and go to the package directory; then extract the files from the archive:

   `cd /home/im3/arnica.iraf`

   `tar -xvf` *path*`/arnica.tar`

   NOTE: If the host operating system is not UNIX and the `tar` command does not exist, the IRAF task `softools.rtar` can be used to extract the `tar` file.

3. Edit the `extern.pkg` file in order to add the `arnica` package. The `extern.pkg` file is in the `hlib$` directory; to find it start the `cl` and type `cd hlib$`. Then, include the following lines in the file :

   `reset arnica = "/home/im3/arnica.iraf/"`

   `task arnica.pkg = "arnica$arnica.cl"`
   Note that the pathname must end with a "/" symbol.

4. Near the end of the `extern.pkg` file add the `arnica` help database to the help database list (copy the syntax of the string) by adding the line:

   `,arnica$lib/helpdb.mip\`

5. Exit from the `cl` and start IRAF again to set the new definitions. Change the directory to the `arnica$src/` directory and type:

   `mkpkg`

   `mkpkg install`

   If no errors are displayed, the package should be correctly installed.