

The NICMOS CALNICA and CALNICB Pipelines

Howard Bushouse

Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218

Abstract. This paper describes the STScI NICMOS calibration pipeline tasks CALNICA and CALNICB. Section 1 describes the input and output files for each task. Sections 2 and 3 describe features that are common to both tasks, and how to run them. Sections 4 and 5 describe in detail the processing steps applied by CALNICA and CALNICB, respectively. Section 6 gives information on the version histories of both tasks.

1. Introduction

The STScI NICMOS calibration pipeline is composed of two major tasks, CALNICA and CALNICB. CALNICA applies instrumental calibration to all images and includes basic corrections such as dark current subtraction, detector non-linearity correction, and flat-fielding. CALNICA operates on one image at a time. It takes as input the raw science data files (designated by the “_raw” filename suffix) produced by the Generic Conversion process in the STScI OPUS environment. Its outputs are calibrated files (“_cal” filename suffix) and, for MULTIACCUM observations only, an intermediate calibrated file (“_ima” filename suffix). CALNICA also adds processing history information to the OPUS trailer files (“_trl” filename suffix).

CALNICB is only applied to associated NICMOS observations and is executed after all images in an association have first been calibrated with CALNICA. CALNICB combines images into mosaics and also removes background signal from the images. Its inputs are an association table (“_asn” filename suffix), the cal files produced by CALNICA, and the science support data files (“_spt” filename suffix) accompanying each cal file. Its outputs are an updated version of the association table (“_asc” filename suffix), mosaic images (“_mos” filename suffix), and an spt file to accompany each mos file. CALNICB also adds processing history information to the trl files in the association.

2. CALNICA/CALNICB Common Features

Both CALNICA and CALNICB are built as host-level tasks and can therefore be executed directly from the host (e.g. Unix or VMS) system level, or they can be executed from within the IRAF environment in the STSDAS `hst_calib.nicmos` package. The tasks are completely data driven in that they derive all information that is necessary to guide the processing from values of the input image header keywords and, in the case of CALNICB, the asn table. The run-time file format for all data I/O is FITS files with IMAGE and BINTABLE extensions. All input data, as well as calibration reference data, is held in memory during processing. Each image set (or “imset”), which is comprised of the five science (SCI), error (ERR), data quality (DQ), samples (SAMP), and exposure time (TIME) arrays associated with each detector readout, occupies ~1 Mbyte of memory. Thus a full 26-readout MULTIACCUM observation will require on the order of 30 Mbytes of memory for the science and reference data. CALNICB will require ~1 Mbyte of memory per input image.

All input and output files conform to the same file structure, which allows output files to be reused as input if desired. All steps involving reference data (e.g. dark and flat-field images) propagate statistical uncertainties from the ERR arrays and data quality flags from the DQ arrays of the reference data into the science data being processed. Furthermore, all steps involving image combination propagate the number of samples (SAMP) and their total exposure time (TIME) used to compute each resulting science (SCI) image value.

3. Running the Tasks

The current version of CALNICA takes two command-line arguments, namely the input and output file names of the data to be processed. The output file name is optional and, if not specified by the user, will default to the root name of the input file. Both the input and output file names can be given as either just root names or full names including the suffixes (e.g. “_raw” and “_cal”) and extension (e.g. “.fits”). File names using suffixes or extensions other than these defaults may be used, but in this case the entire file name must be specified by the user. The following examples are all valid ways of running CALNICA:

- `cl> calnica n45m04c8m ""`
- `cl> calnica n45m04c8m_raw.fits ""`
- `cl> calnica n45m04c8m_raw.fits n45m04c8m_cal.fits`
- `cl> calnica n45m04c8m_cal.fits n45m04c8m_special`

CALNICB takes one command-line argument, the name of the input association table. The association table file name can be specified using either only the root name or the full file name. For example:

- `cl> calnicb n3v701060`
- `cl> calnicb n3v701060_asn.fits`

Before running either task, the user may want to modify some of the input image header keywords containing reference file names or calibration “switches”. It is very important to remember that all such keywords are contained in the *primary* header of NICMOS data files, therefore it is the primary header (extension number zero) that must be edited. For example, the IRAF `hedit` or STSDAS `chcalpar` tasks must be used as follows:

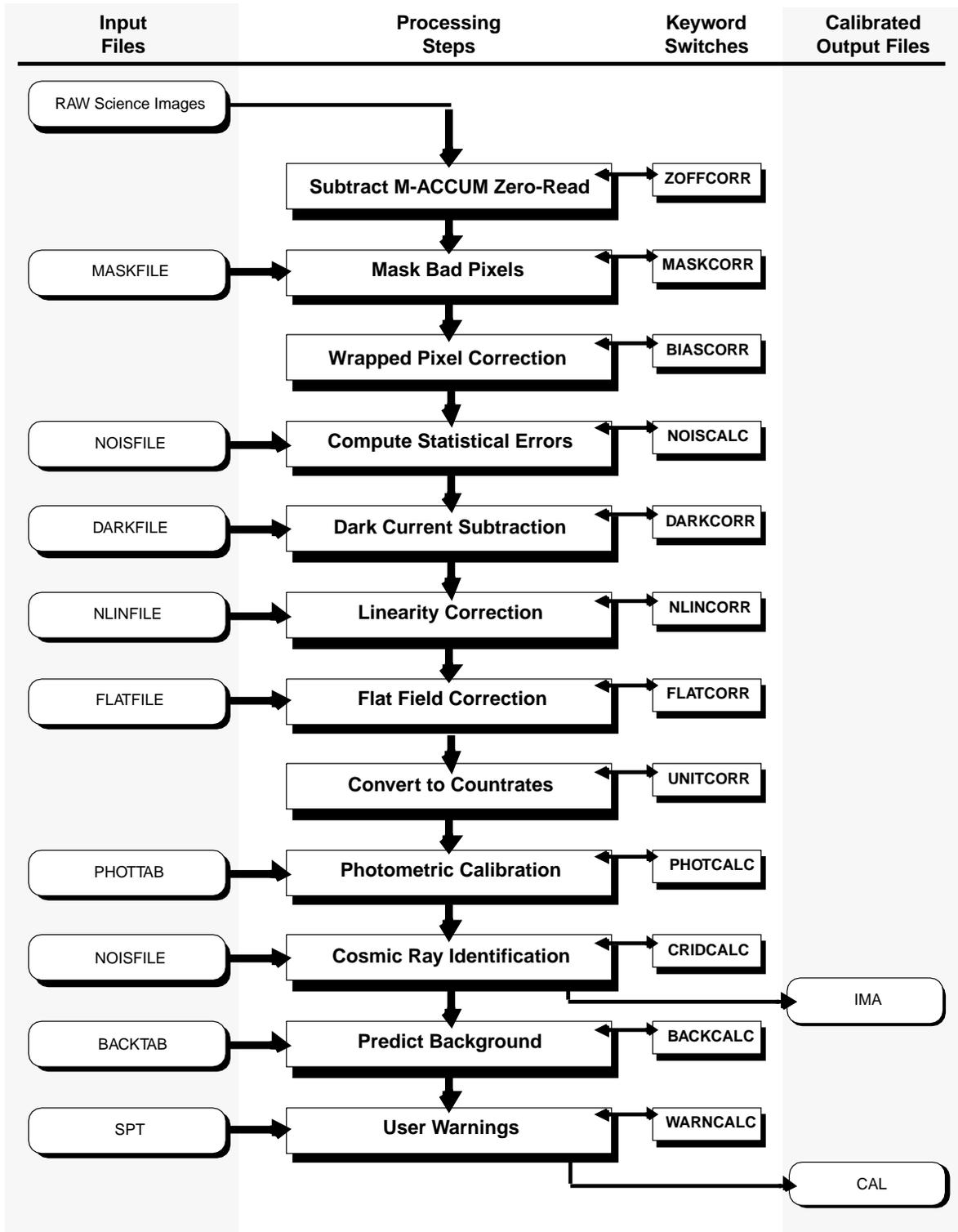
- `cl> hedit myfile_raw.fits[0] <keyword> <value>`
- `cl> chcalpar myfile_raw.fits[0]`

where the “[0]” appended to the file name indicates that it is the primary header which is to be modified.

4. CALNICA Processing

The overall flow of CALNICA processing is shown in Figure 1. Included in this figure are the names of the image header keywords that are read by CALNICA to determine the names of reference files (left column) and the “switches” used to turn each step on or off. The details of each step are described below.

Figure 1. CALNICA Processing Flow.



4.1. ZOFFCORR

This step is only executed for MULTIACCUM observations and simply subtracts the SCI image of the zeroth readout from the SCI images of all other readouts. The zeroth-read SCI image is also subtracted from itself, so the zeroth-read SCI image in the output ima file will have a constant zero value. The zeroth-read DQ image is also logically “or-ed” with the DQ images of all other readouts.

4.2. MASKCORR

This step propagates (logical “or” operation) the DQ image from the MASKFILE reference file into the DQ images of the file being processed. The MASKFILE DQ image contains flags for the known hot and cold pixels (flag value=32) in the NICMOS detectors.

4.3. BIASCORR

This step is only necessary for non-MULTIACCUM observations in which the difference of the final and initial detector readouts is computed on-board. This computation is performed in 16-bit arithmetic and therefore it is possible, when observing bright targets, for the result of the subtraction to exceed the dynamic range of the calculation, in which case the values for pixels with very large signal will “wrap around” the maximum available value of +32767 to negative values in the range -23500 to -32768 DN. This step searches the SCI images for values in this negative range and, upon finding any, adds 65536 to them to restore them to their proper values.

4.4. NOISCALC

This step initializes the ERR images by summing (in quadrature) the detector readnoise and the Poisson noise in the recorded signal for each pixel, i.e.

$$ERR = \sqrt{readnoise^2 + SCI * adcgain} / adcgain, \quad (1)$$

where *adcgain* is the electron-to-DN conversion factor. The NOISFILE contains the pixel-by-pixel readnoise, in units of electrons, in its SCI image. The NOISFILE DQ image data is also propagated into the DQ images being processed.

4.5. DARKCORR

This step subtracts the detector dark current signal from the science data. It is necessary to use a DARKFILE containing dark images with exposure times matching those of the science data readouts so that the correct levels of dark current, amplifier glow, and “shading” are removed from the data. CALNICA selects a DARKFILE imset that has a matching exposure time for each science imset exposure time and subtracts the DARKFILE SCI image from the SCI images being processed. The DARKFILE ERR images are summed (in quadrature) with the science file ERR images, and the DARKFILE DQ image data are also propagated into the DQ images of the data being processed.

4.6. NLINCORR

This step corrects for the non-linear response of the detectors. The response can be conveniently divided into three regimes. In the low signal regime, the detector response is linear, hence no correction is applied. In the mid-level regime, the response deviates from true linearity in a way that is correctable using a first-order polynomial of the form

$$SCI(corrected) = (c_1 + c_2 * SCI) * SCI \quad (2)$$

$$ERR(corrected) = \sqrt{ERR^2 + e_1 + e_2 * SCI^2} \quad (3)$$

where c_1 and c_2 are the polynomial coefficients and e_1 and e_2 are the uncertainties in the coefficients. These quantities are stored, on a pixel-by-pixel basis, in the “coef” and “err” images in the NLINFILE. In the high signal regime, where pixels begin to saturate, no correction is applied to the data, but a saturation flag value is set in the DQ image. The DN levels defining the boundaries between the regimes are also stored on a pixel-by-pixel basis in two “node” images in the NLINFILE.

4.7. FLATCORR

This step corrects for pixel-to-pixel gain variations by multiplying the SCI images by the inverse flat field image contained in the FLATFILE SCI image. The FLATFILE ERR and DQ image data are propagated into the science data ERR and DQ images. The ERR data are updated using

$$ERR = \sqrt{(SCI * ERR_{FLATFILE})^2 + (ERR * SCI_{FLATFILE})^2} \quad (4)$$

4.8. UNITCORR

The step converts the data from units of counts to countrates (DN per second) by dividing both the SCI and ERR image data by the value of the SAMPTIME keyword for each imset.

4.9. PHOTCALC

This step does not alter the data, but simply sets the values of certain photometry-related header keywords. The values of the PHOTFLAM and PHOTFNU keywords can be used to convert data to an absolute flux scale using

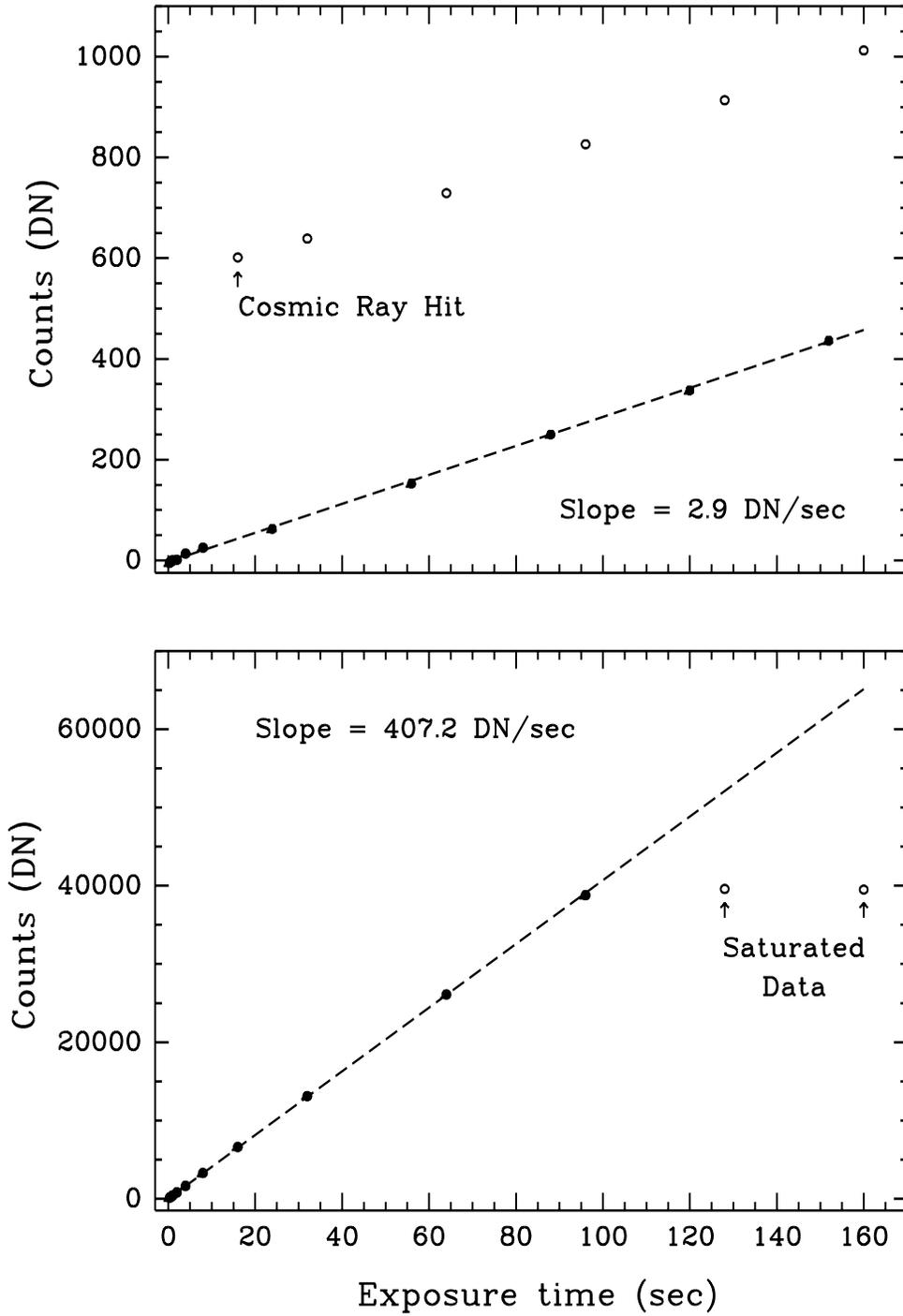
$$f_{\lambda}(\text{ergs/sec/cm}^2/\text{\AA}) = SCI(\text{DN/sec}) * PHOTFLAM \quad (5)$$

$$f_{\nu}(\text{Jy}) = SCI(\text{DN/sec}) * PHOTFNU \quad (6)$$

4.10. CRIDCALC

For MULTIACCUM observations, this step combines the calibrated data from all readouts into a single image and, in the process, rejects bad data due to, for example, cosmic ray hits and saturation. The rejection of data and computation of the final image are accomplished in two steps. First, accumulated signals from one readout to the next are differenced so that the signal that arrived between individual readouts can be examined. The error-weighted mean of the difference signals is computed, not including data samples already flagged with a non-zero DQ value. Outliers are identified as those samples lying more than 5 times their error away from the mean. This process is iterated by rejecting the largest outlier and recomputing the mean until no new outliers are found. When all outliers have been rejected, the accumulated signal as a function of time is recomputed using only the non-rejected samples. These data are then fit with a first-order polynomial using standard linear regression. The slope of the polynomial is used as the final countrate (SCI) for each output (“_cal” file) pixel and the uncertainty in the slope is assigned to the output ERR value. The number of non-rejected data samples and their total exposure time are recorded in the output SAMP and TIME images, respectively. If at least one good sample was found for a pixel, its output DQ value is set to zero. The ima file SCI, ERR, SAMP, and TIME image data are not modified by this step. The ima file DQ images will have cosmic ray flag values set for those pixels where samples were rejected. Examples of the fits produced for pixels having a cosmic ray hit and saturated samples are shown in Figure 2.

Figure 2. CRIDCALC Examples.



4.11. BACKCALC

This step does not alter the data. It computes an estimate of the sky plus telescope background signal for an observation, based on its position in the sky relative to the sun and zodiacal plane. The BACKTAB file contains a table of background model parameters. This step has not yet been implemented in the current version of CALNICA.

4.12. WARNCALC

This step also does not alter the data. It examines engineering data contained in the `_spt` file and issues warnings to the user if certain parameters, such as temperatures or voltages, had suspect values during the observation. This step has not yet been implemented.

5. CALNICB Processing

The overall flow of CALNICB processing is shown in Figure 3. Each step is described in the following sections.

5.1. Read ASN Table

The association (`asn`) file is a FITS file with a BINTABLE extension containing the list of (root) file names to be processed, as well as the (root) file names of all output mosaic (`mos`) image files to be produced. CALNICB determines which names in the list are input file and which are output files based on the MEMTYPE column in the ASN table. Input files have a MEMTYPE prefix of “EXP” (exposure), while output files have a prefix of “PROD” (products). Furthermore, association members that are observations of the target of interest have a MEMTYPE suffix of “TARG” (target), while observations of sky background locations (which are only produced by using chop patterns) have a suffix of “BCKn”, where `n` is the background region number in the pattern (see Table 1). One `mos` image will always be produced for the target, and chop patterns will result in one `mos` image being produced for each background region observed.

5.2. Read CAL Images

This step simply reads the input `cal` images into memory.

5.3. Determine Processing Parameters

This step reads the values of various keywords from the input image headers to determine what processing needs to be performed. Certain keywords, such as PATTERN, NUMPOS, and NUMITER should have the same value in all images and are therefore only read from the first image in the association. Others are image specific, such as PATT.POS and World Coordinate System (WCS) keywords, have unique values in each image and are therefore read from each image. From this information CALNICB determines what type of pattern was used, how many positions are in the pattern, how many images there are at each pattern position, the relative positions of each image in the pattern, and which input images go into which output mosaic.

5.4. Combine NUMITER Images

If there is more than one image at each pattern position, this step will combine the images at a given position into a single image. The offsets between each image at a given position are computed from their WCS information, using the first image at each pattern position as a reference. These offsets are then refined using a cross-correlation technique. The images are then aligned with their reference image, using bilinear interpolation, and are then combined. The combining process computes the ERR-weighted mean for each SCI

Figure 3. CALNICB Processing Flow.

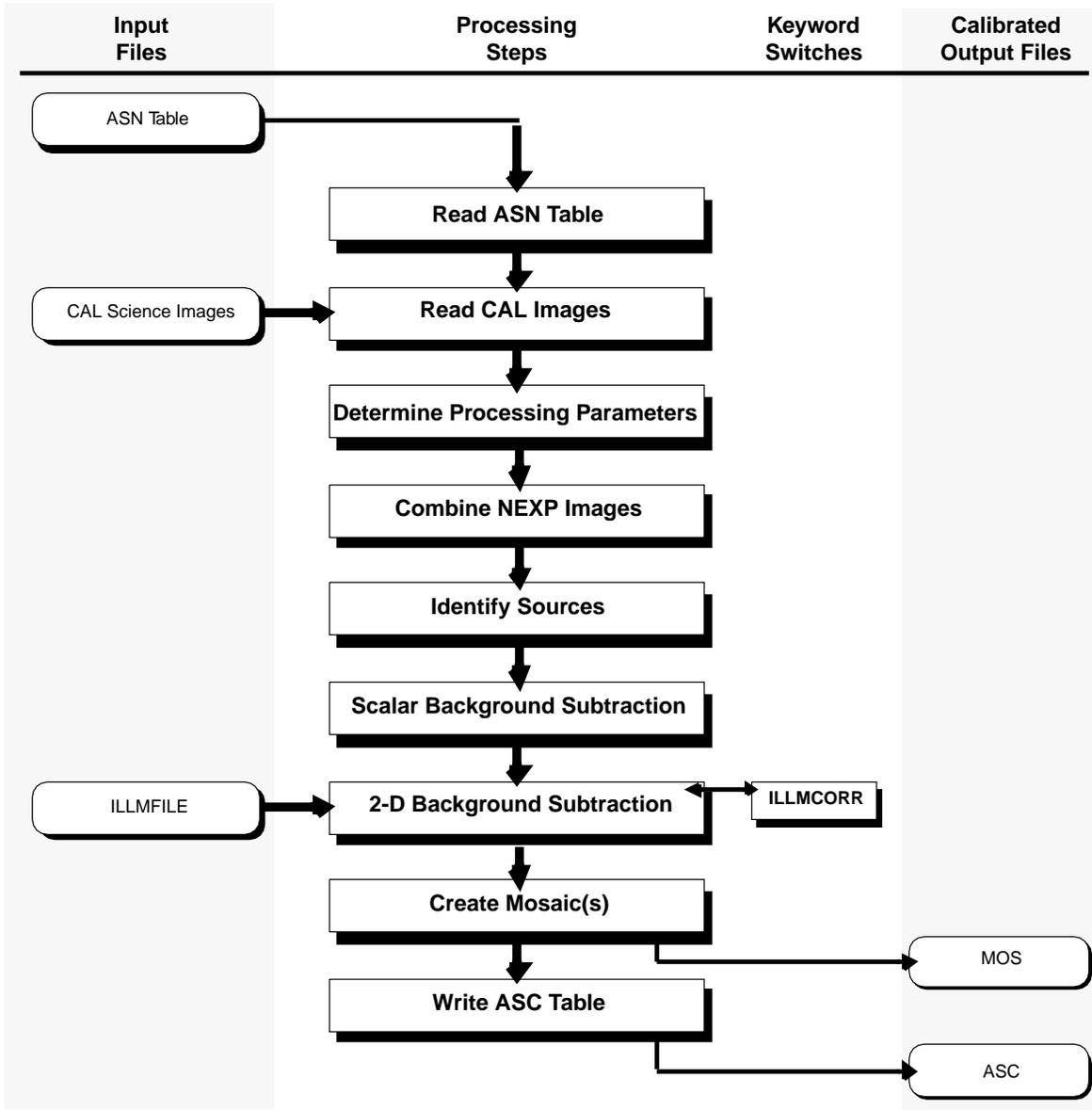


image value, with rejection of bad (non-zero DQ) pixels and iterative sigma clipping. The number of non-rejected samples, and their total exposure time, are saved in the SAMP and TIME images.

5.5. Identify Sources

This step flags (in the DQ images) pixels suspected of containing flux from a source. This is accomplished using a 3-stage process. First, pixels that are greater than 5 sigma above the mean signal level in each image are identified as candidate source pixels. Second, in order to filter out spurious identifications, only those candidates having 2 or more neighboring pixels also identified as containing a source are retained. Finally, the surviving pixels have their source DQ flags “grown” to their 4 immediate neighbors.

5.6. Scalar Background Subtraction

This step computes and subtracts a scalar (or “DC”) background signal from all images in the association. This is accomplished as follows. First, the mean signal level in each BCK image in the association is computed. The computation excludes bad and source-flagged pixels and uses iterative sigma clipping to avoid any possible contamination. If there are not any BCK images in the association (e.g. the observations are from a pure dither pattern), the same computation is performed on the target images. In this case the computed background level is compared with the CALNICA estimate that was stored in the BACKESTn keywords and if the computed results differ substantially from the CALNICA estimates, it is assumed that the computed result is biased by the presence of the target and it is replaced with the CALNICA estimate. The resulting background levels for each image are then averaged to compute the overall mean background level for the whole association. This overall mean is then subtracted from all images. The mean background level is recorded in the MEAN_BKG keyword in the output association (asc) table.

5.7. 2-D Background Subtraction

This step subtracts the ILLMFILE image from the association images. It is intended to remove any spatial variations in the telescope background illumination pattern, but to date, little if any such spatial variations have been seen. Thus the ILLMFILES currently contain dummy data (constant zero values). CALNICB recognizes that the files are dummy and skips the subtraction.

5.8. Create Mosaics

This step performs the final combination of images to produce the output mosaics. The process is very similar to the step that combines the NUMITER images at each pattern position. The relative offsets for each image in a given mosaic are first computed from their WCS information, using the first image in each mosaic as a reference. The offsets are refined using cross-correlation, and then the images are aligned using bilinear interpolation and combined. The combining process computes the ERR-weighted mean of overlapping pixels, rejecting those with non-zero DQ values and applying an iterative sigma clipping. The number of non-rejected samples and their total exposure time are saved in the output SAMP and TIME images of the mos files.

5.9. Write ASC Table

This step creates the output association (asc) table. The asc table is a copy of the input asn table, with four new columns of information appended. An example of an asc table is shown in Table 1. The new columns are BCKIMAGE, MEANBCK, XOFFSET, and YOFFSET. The BCKIMAGE column indicates whether or not a given image was used in the scalar background computation. The example shown in Table 1 is for a TWO-CHOP

pattern, with NUMITER=2, and NUMPOS=5. So there are two images at each of the 5 pattern positions and two of the positions are background (BCK) regions. Thus, in this example, only the images at the BCK positions were used to compute the scalar background level. The MEANBCK column records the background signal computed for each individual image. The XOFFSET and YOFFSET columns record the computed offsets for each image relative to its reference image.

Table 1. Output Association (ASC) Table.

MEMNAME	MEMTYPE	MEMPRSNT	BCKIMAGE	MEANBCK (DN/sec)	XOFFSET (pixels)	YOFFSET (pixels)
n3uw01a1r	EXP-TARG	yes	no	INDEF	0.00	0.00
n3uw01a2r	EXP-TARG	yes	no	INDEF	-0.15	0.00
n3uw01a3r	EXP-BCK1	yes	yes	0.28	0.00	0.00
n3uw01a4r	EXP-BCK1	yes	no	INDEF	0.00	0.20
n3uw01a5r	EXP-TARG	yes	no	INDEF	0.32	0.00
n3uw01a6r	EXP-TARG	yes	no	INDEF	0.00	0.00
n3uw01a7r	EXP-BCK2	yes	yes	0.26	0.00	0.00
n3uw01a8r	EXP-BCK2	yes	no	INDEF	0.00	0.00
n3uw01a9r	EXP-TARG	yes	no	INDEF	0.00	0.00
n3uw01a0r	EXP-TARG	yes	no	INDEF	0.00	-0.15
n3uw01010	PROD-TARG	yes	no	INDEF	INDEF	INDEF
n3uw01011	PROD-BCK1	yes	no	INDEF	INDEF	INDEF
n3uw01012	PROD-BCK2	yes	no	INDEF	INDEF	INDEF

6. Version Histories

6.1. CALNICA

- Version 2.1: Version in use in OPUS at the time NICMOS was launched.
- Version 2.2: Installed 19 May 1997, this version corrected minor arithmetic bugs.
- Version 2.3: Installed 19 June 1997, this version contained a major rewrite of the CRIDCALC step, and also populates the CAL_VER keyword in the output file headers to record which version of CALNICA was used to process the data.
- Version 3.0: This version is currently in testing. It includes a modification to the WRAPCORR threshold, fixes some bugs in the CRIDCALC routine, corrects for signal from bright sources in the zeroth-read of MULTIACCUM observations, and allows the user to set the cosmic ray rejection threshold level for the CRIDCALC step.

6.2. CALNICB

- Version 1.4: Version in use in OPUS at the time NICMOS was launched.
- Version 1.5: Installed 11 April 1997, this version fixed minor bugs.
- Version 1.6: Installed 19 May 1997, this version correctly updated the FILENAME and EXTNAME keywords in all output files.
- Version 2.1.1: Installed 14 July 1997, this version was a major rewrite of the entire task and introduced the use of cross-correlation for refining image offsets, the flagging of sources, and the rejection of source-flagged pixels from the scalar background computation.
- Version 2.1.2: Installed 23 July 1997, this version fixed a sporadic arithmetic bug in the cross-correlation routine.