

# NICMOS Calibration

## In This Chapter...

Pipeline Processing / 16-1  
NICMOS Calibration Software / 16-2  
Basic Data Reduction: calnica / 16-3  
Mosaicing: Calnicb / 16-10  
Grism Data Reduction: Calnicc / 16-16  
Recalibration / 16-21

This chapter is designed to help you understand what steps are performed on your data in the routine pipeline process and to help you decide whether you should recalibrate your data with different reference files. In this chapter we:

- Provide flowcharts and descriptions of the NICMOS pipeline calibration steps.
- Explain how to recalibrate your data using the calibration software in STSDAS.

---

## 16.1 Pipeline Processing

Shortly after your data arrive at STScI they pass through the OPUS *pipeline* which processes and calibrates them. All of the steps performed by the pipeline are recorded in the trailer file for your dataset (`*_trl.fits`). The main steps performed by the pipeline are:

1. The data are partitioned (separated into individual files, e.g., engineering and science data are separated).
2. The data are edited, if necessary, to insert fill in place of missing data.
3. The data are evaluated to determine discrepancies between the subset of the planned and executed observational parameters.

4. A list of calibration reference files to be used in the calibration of the data is created based on the executed observational parameters. This step does not generate comments in the NICMOS trailer file.
5. The data are converted to a generic (FITS) format and the header keyword values are populated (known as *generic conversion*).
6. The data are calibrated using a standard set of NICMOS calibration algorithms (**calnica** and **calnicb**).

The calibration software used by the pipeline (step 6 above) is exactly the same as that provided within STSDAS (see below). The calibration files and tables used are taken from the Calibration Data Base System (CDBS) at STScI and are the most up-to-date calibration files appropriate for the instrument configuration used in the observation.

---

## 16.2 NICMOS Calibration Software

### 16.2.1 The Calibration Pipeline

The science data that an observer receives are calibrated in the pipeline by at least one, and possibly two, STSDAS calibration routines: **calnica** and **calnicb**. The two routines perform different operations:

1. **calnica**: This routine removes the instrumental signature from the science data. It is the first calibration step, and is applied to *all* NICMOS datasets individually. **Calnica** operates on the raw science data files.
2. **calnicb**: This routine operates on *associations*: it co-adds datasets obtained from multiple iterations of the same exposure, mosaics images obtained from dither patterns, and background-subtracts images obtained from chop patterns. **Calnicb** is applied to the calibrated science data files (output from **calnica**), and requires association tables and the `*_spt.fits` files.

Both tasks determine which calibration steps are to be performed by looking at the values of the *calibration switches* keywords in the primary header of the input science data files (see Table 15.5). The tasks select the reference files to use in the calibration of the data by retrieving the reference file names from the *reference file* keywords. The appropriate values of the calibration switches and reference file keywords depend on the instrumental configuration used, the date when the observations were taken, and any special pre-specified constraints. They are set in the headers of the raw data file in the pipeline during the generic conversion process. The *calibration indicators* keywords record which steps have been performed on the data, and get updated after processing. In particular, the indicators for completed steps will have been assigned the value “PERFORMED”, while the indicators for the steps that were not performed will have been set to “OMITTED” or “SKIPPED”. The calibration indicators keywords should be examined in the primary header of the calibrated science data (`*_cal.fits`) to determine what calibration steps were applied to the data.

The **calnica** and **calnicb** tasks are available in STSDAS under the **hst\_calib\$nicmos** package. By using these tasks, observers can recalibrate data using the same software as the routine calibration pipeline at STScI.

## 16.2.2 Grism Spectroscopy

A unique capability of NICMOS is the grism mode, which permits multi-object, slitless spectroscopy at low resolution. Grism data are processed with separate calibration software, **calnicc**, which performs a series of steps that identify and extract spectra from the images. The inputs to **calnicc** are the calibrated images (`*_cal.fits`) produced by **calnica**.

**Calnicc** was developed using the Interactive Data Language (IDL) software at the Space Telescope European Coordinating Facility (ST-ECF). Currently **calnicc** *is not part of the automatic pipeline processing*, and users must apply the calibration software to their grism images. The manual (written by W. Freudling and R. Thomas) describing the software, its installation, and use can be found at the following WWW address:

<http://ecf.hq.eso.org/nicmos/calnicc/calnicc.html>

Most users will want to start the spectrum extraction processing using **NICMOSlook**, the interactive counterpart to **calnicc**. **NICMOSlook** is written in IDL, and is a quick-look spectrum extraction tool for grism spectra. Unlike **calnicc**, **NICMOSlook** requires users to specify parameters interactively (e.g., the best way to find an object, the weights to be given in the spectral extraction). This tool is recommended for first-time users or users with a small number of grism data. Once you are familiar with the extraction process and parameters, use **calnicc**. The user manual (written by W. Freudling, R. Thomas, and L. Yan), the software, and instructions for its installation can be found at the ST-ECF WWW address:

<http://ecf.hq.eso.org/nicmos/nicmoslook>

In this handbook, the main steps of the reduction and extraction performed by **calnicc** will be described.



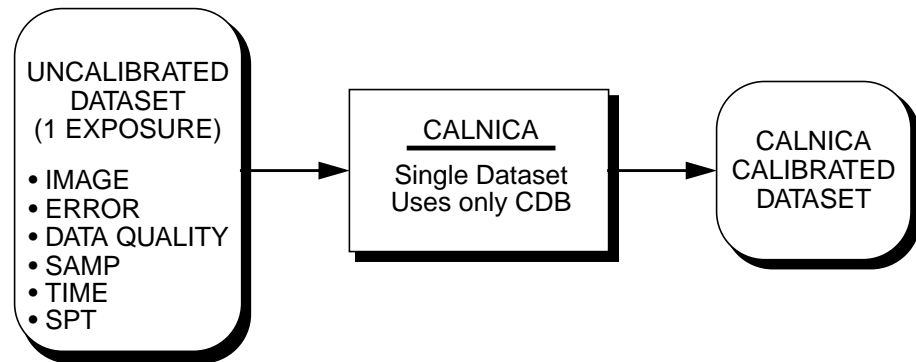

---

You cannot run either **calnicc** or **NICMOSlook** unless you have an IDL licence.

---

## 16.3 Basic Data Reduction: calnica

The **calnica** task (Figure 16.1) operates on individual NICMOS datasets and performs the job of removing the instrumental signature from the raw science data. The **calnica** task also tries to identify cosmic ray hits in the MULTIACCUM images.

**Figure 16.1:** Conceptual calnica Pipeline; CDB is the Calibration DataBase

The inputs to **calnica** are the raw science (`*_raw.fits`) files. The output of **calnica** is usually a single file containing the *calibrated* science data (`*_cal.fits`). For **MULTIACCUM** mode datasets there is an additional intermediate output file (`*_ima.fits`) which contains the *calibrated* data from all the intermediate readouts. The `_ima.fits` data are fully calibrated up to, but not including, the cosmic ray rejection. The format and contents of the input and output science data files are identical so that the output data can be reused as input to **calnica**, if desired. One could, for example, process a science data file through some subset of the normal calibration steps performed by **calnica**, examine or modify the results, and then process the data through **calnica** again, performing other calibration steps or using alternate calibration reference files.

Figure 16.2 shows the portion of a calibrated NICMOS science file header containing the switches and reference file keywords that pertain to the processing performed by **calnica**. The accompanying flow chart (Figure 16.3) shows the sequence of **calnica** calibration steps, the input data and reference files and tables, and the output data file. Each calibration step is described in detail in the following sections.

### ZOFFCORR

The **ZOFFCORR** step of **calnica** performs the subtraction of the zeroth read from all readouts in a **MULTIACCUM** file. This step is performed for data generated by the **MULTIACCUM** readout mode only. For other readout modes, the subtraction of the zeroth read is performed on-board, because the images returned to the ground are formed by taking the difference of initial and final non-destructive detector readouts.

The pipeline will subtract the zeroth read image from all readouts, including the zeroth read itself. Furthermore, the self-subtracted zeroth-read image will be propagated through the remaining processing steps and included in the output products, so that a complete history of error estimates and data quality (DQ) flags is preserved. After this step is performed, the science data are in the same form as the raw science data from any other observing mode and are processed the same way throughout the remaining steps of **calnica**. No reference files are used by this step.

Figure 16.2: Partial NICMOS Header

```

/ CALNICA CALIBRATION REFERENCE FILES
MASKFILE= "nref$H4214599n_msk.fits" / static data quality file
NOISFILE= "nref$H4216218n_noi.fits" / detector read noise file
NLINFILE= "/data/mimir2/eddie/7116/customlin.fits"
DARKFILE= "/data/mimir2/eddie/darkplace/c2_step256_med.fits"
FLATFILE= "/data/mimir2/eddie/Flatplace/c2_f110w_new.fits"
PHOTTAB = "nref$H1t0826ln_pht.fits" / photometric calibration table
BACKTAB = " " / background model parameters table

/ CALNICA CALIBRATION REFERENCE FILE PEDIGREE
MASKPGR= "GROUND 15/01/1997" / static data quality file pedigree
NOISPGR= "GROUND 27/01/1997" / detector read noise file pedigree
NLINPGR= "GROUND 27/01/1997" / detector nonlinearities file pedigree
DARKPGR= "INFLIGHT 10/03/1997" / dark current file pedigree
FLATPGR= "GROUND 15/01/1997" / flat field file pedigree
PHOTPGR= "GROUND 17/10/1996" / photometric calibration table pedigree
BACKPGR= " " / background model parameters table pedigree

/ CALNICA CALIBRATION SWITCHES: perform,omit
BIASCORR= "PERFORM" / subtract ADC bias level
ZOFFCORR= "PERFORM" / subtract MULTI-ACCUM zero read
MASKCORR= "PERFORM" / data quality initialization
NOISCALC= "PERFORM" / calculate statistic errors
NLINCORR= "PERFORM" / correct for detector nonlinearities
DARKCORR= "PERFORM" / dark correction
FLATCORR= "PERFORM" / flat field correction
UNITCORR= "PERFORM" / convert to count rates
PHOTCALC= "PERFORM" / calculate photometric keywords
CRIDCALC= "PERFORM" / identify cosmic ray hits
BACKCALC= "PERFORM" / calculate background estimates
WARNCALC= "OMIT" / generate user warnings

/ CALNICA CALIBRATION INDICATORS: performed, skipped, omitted
BIASDONE= "PERFORMED" / subtract ADC bias level
ZOFFDONE= "PERFORMED" / subtract MULTI-ACCUM zero read
MASKDONE= "PERFORMED" / data quality initialization
NOISDONE= "PERFORMED" / calculate statistic errors
NLINDONE= "PERFORMED" / correct for detector nonlinearities
DARKDONE= "PERFORMED" / dark correction

```

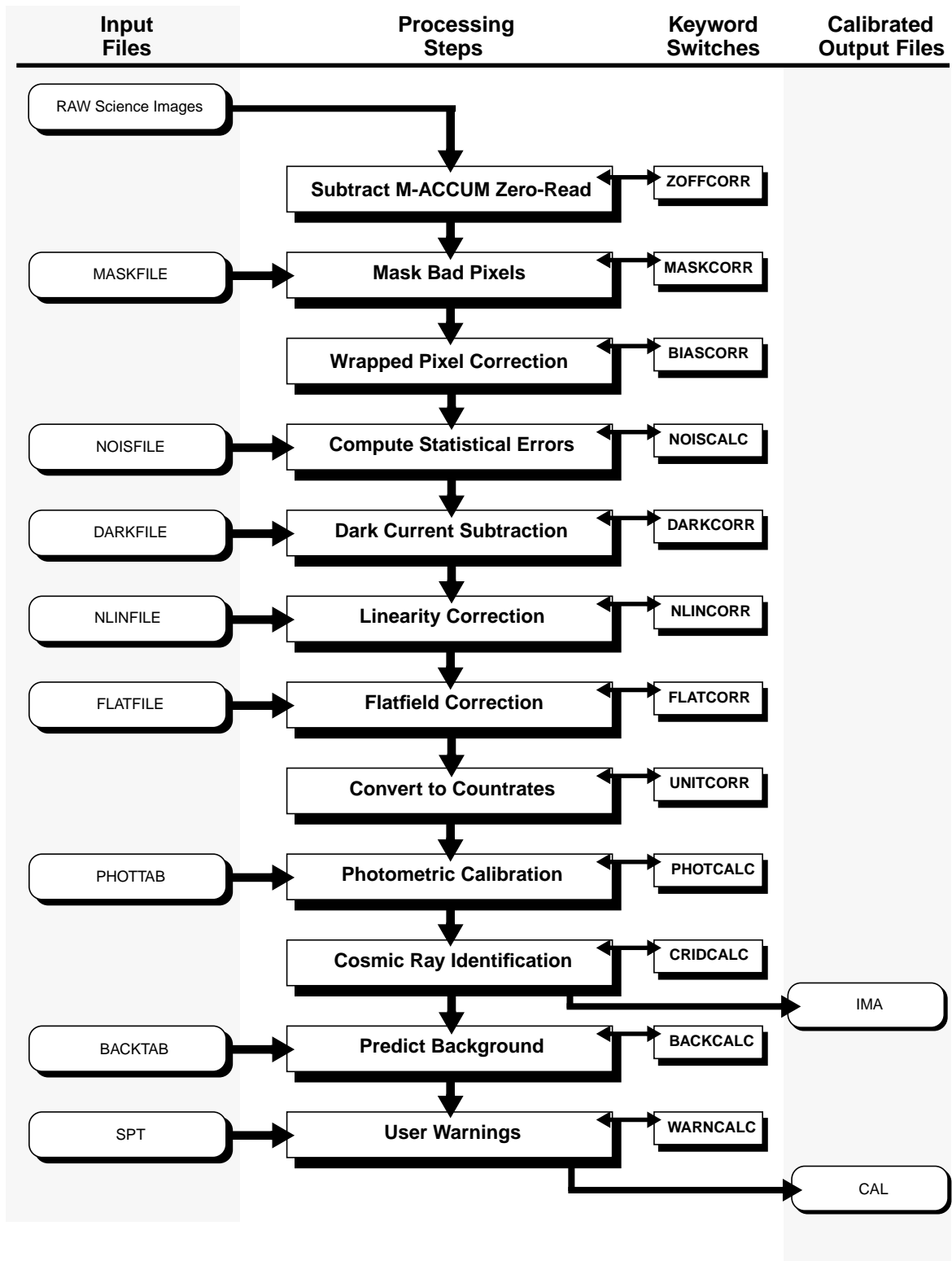
## MASKCORR

Flag values from the static bad pixel mask file are added to the DQ image. This uses the MASKFILE reference file, which contains a flag array for known bad (hot or cold) pixels. There is one MASKFILE for each detector.

## BIASCORR

NICMOS uses 16-bit analog-to-digital converters (ADCs), which convert the analog signal generated by the detectors into signed 16-bit integers. Because the numbers are signed and because the full dynamic range of the converter output is used, raw pixel values obtained from individual detector readouts can range from  $-32768$  to  $+32767$  DN. In practice the detector bias level is set so that a zero signal results in a raw value on the order of  $-23000$  DN. In ACCUM, BRIGHT OBJECT, and RAMP modes, where the difference of initial and final readouts is computed on-board, the subtraction is also performed in 16-bit arithmetic. Therefore, it is possible that the difference between the final and initial pixel values for a bright source could exceed the dynamic range of the calculation, in which case the final pixel value will wrap around the maximum allowed by the 16-bit arithmetic, resulting in a negative DN value. Given the level at which the

Figure 16.3: Pipeline Processing by CALNICA



NICMOS / 16

NICMOS detectors saturate, and the analog-to-digital conversion factor, the maximum “real” pixel value that is expected is on the order of +40000 DN. Such a value will be wrapped to about –26000 DN by the on-board difference calculation.

The BIASCORR step therefore searches for pixel values in the range –26000 to –32768 and adds an offset of 65536 DN to these pixel values to reset them to their original real values.

No reference files are used by this step.

## NOISCALC

NICMOS calculates statistical errors for the science data only in the RAMP observing mode. Errors for all other modes are computed in the **calnica** pipeline. The NOISCALC step performs the task of computing an estimate of the errors associated with the raw science data based on a noise model for each detector. Currently the noise model is a simple combination of detector read noise and Poisson noise in the signal, such that:

$$\sigma = \frac{\sqrt{\sigma_{rd}^2 + \text{counts} \cdot \text{adcgain}}}{\text{adcgain}} \quad \text{Eq. 16.1}$$

where  $\sigma_{rd}$  is the read noise in units of electrons, *adcgain* is the analog-to-digital conversion gain factor (in electrons per DN) and *counts* is the signal in a pixel in units of DN. Noise is computed in units of electrons, but the result is converted to units of DN for storage in the error image. The detector read noise is read pixel-by-pixel from the NOISFILE reference image and depends on the read rate of the observation, as well as the number of initial and final reads (NREAD). Separate NOISFILES are required for each combination of read rate and NREAD. The data quality flags set in the DQ image of the NOISFILE are propagated into the DQ images of all image sets (imsets) being processed.

For RAMP mode observations, the error estimate computed by the instrument (and therefore, present in the raw science file) is a variance. For these observations the NOISCALC step simply computes the square-root of the raw error image so that it is on the same basis as other modes.

Throughout the remaining steps in **calnica**, the error image is processed in lock-step with the science image, getting updated as appropriate. Errors are mostly propagated through combination in quadrature.

## DARKCORR

The detector dark current is removed from the science image by subtracting a dark current reference image appropriate for the exposure time of the science data.

A simple scaling of a single dark reference image to match the exposure time of the science data is, unfortunately, not possible due to the non-linear behavior of the dark current as a function of time. Therefore, a library of dark current images is maintained, covering the range of exposure times of the MULTIACCUM sequences and a subset of ACCUM exposure times and NREAD values (see the *NICMOS Instrument Handbook*). The reference dark appropriate for the exposure time and sequence used in MULTIACCUM, or the exposure time and NREAD values used in ACCUM, is determined and subtracted from the data. For

non-standard MULTIACCUM sequences (see the *NICMOS Instrument Handbook*) and for some ACCUM exposure times, the appropriate dark image is interpolated from existing dark files. There is one reference file (DARKFILE) per detector, which contains the set of dark images (at various exposure times) for that detector. Error estimates of the dark current, stored in the ERR images of the DARKFILE, are propagated in quadrature into the ERR images of all processed IMSETs. Data quality (DQ) flags set in the DARKFILE are also propagated into the DQ images of all processed imsets.

Sets of synthetic dark reference files have been recently generated to populate the calibration database. These dark files reproduce all the characteristics of on-orbit darks, which do not suffer from the *pedestal* (see Chapter 17).

Dark subtraction is skipped in BRIGHTOBJ mode, because the short exposure times should result in insignificant dark current.

### NLINCORR

The linearization correction step corrects the integrated counts in the science image for the non-linear response of the detectors. The observed response of the detectors can conveniently be represented by 3 regimes:

- At low signal levels the response is linear and no correction is needed; the low signal level for NICMOS is pixel- and camera-dependent, and is about 14,500 DN and below, with a standard deviation of about 400 DN.
- At intermediate levels the detector response deviates in a linear fashion from the incident flux and is easily correctable via the expression:

$$F_c = (c1 + c2 \times F) \times F$$

where  $c1$  and  $c2$  are the correction coefficients,  $F$  is the uncorrected flux (in DN) and  $F_c$  is the corrected flux.

- At high signal levels—as saturation sets in—the response becomes highly non-linear and is not correctable to a scientifically useful degree; the saturation level is about 30,500 DN, with a standard deviation of about 2,000 DN.

The NLINCORR step applies no correction to pixels in the low signal regime, applies a linear correction term to intermediate level pixels, and applies no correction to pixels in the high signal regime but rather flags them in the DQ image as saturated. This step uses the NLINFILE reference file, which consists of a set of images containing correction coefficients and variances at each pixel. Error estimates on the correction applied to intermediate-level pixels are propagated into the ERR images of all imsets processed. Data quality flags set in the NLINFILE are also propagated into the processed DQ images. There is one NLINFILE per detector.

### FLATCORR

In this step the science data are corrected for variations in gain between pixels by multiplying by an (inverse) flatfield reference image. This step is skipped for

observations using a grism because the flatfield corrections are wavelength dependent. This step uses the FLATFILE reference file, which contains the flatfield image for a given detector and filter (or polarizer) combination. Error estimates and DQ flags contained in the FLATFILE are propagated into the processed images. There is one FLATFILE per detector and filter combination.

### UNITCORR

The conversion from raw counts to count rates is performed by dividing the science and error image data by the exposure time. No reference file is needed.

### PHOTCALC

This step provides photometric calibration information by populating the photometry keywords PHOTMODE, PHOTFLAM, PHOTFNU, PHOTZPT, PHOTPLAM, and PHOTBW with values appropriate to the camera and filter combination used for the observation. The photometry parameters are read from the PHOTTAB reference file, which is a FITS binary table containing the parameters for all observation modes. The values of PHOTFLAM and PHOTFNU are useful for converting observed count rates to absolute fluxes in units of  $\text{erg/s/cm}^2/\text{Angstrom}$  or Jy, respectively.

### CRIDCALC

This step identifies and flags pixels suspected of containing cosmic ray (CR) hits. For MULTIACCUM mode observations, this step also combines the data from all readouts into a single image. In MULTIACCUM mode, the data from all readouts are analyzed pixel-by-pixel, searching for and identifying the data from individual readouts that appear as outliers using an iterative sigma-clipping technique. In each pixel, the signal from each readout is ascribed a standard deviation given by the combination of readnoise and Poisson noise. Values which deviate more than  $5\sigma$  from the mean slope of the counts-versus-time relation are identified as outliers, and the corresponding pixels in the DQ images of the intermediate MULTIACCUM (\*\_ima.fits) file are flagged, however, the pixel values themselves in the SCI and ERR images are unchanged. Once all outliers have been identified, a final, combined value is computed for each pixel using only non-flagged samples. The result of this operation is stored as a single imset in the output \*\_cal.fits file in which the number of unflagged samples used to compute the final value for each pixel and the total exposure time of those samples will be reflected in the SAMP and TIME images. The variance ascribed to the final mean countrate is the uncertainty in the slope of the counts-versus-time relation at each pixel location. Pixels for which there are no unflagged samples, e.g., permanently hot or cold pixels, will have their output SCI, ERR, SAMP, and TIME values set to zero, with a DQ value that contains all flags that were set. The cosmic ray rejection threshold,  $5\sigma$ , is currently hardwired in the **calnica** code, and seems to perform a reasonable job of identifying and rejecting cosmic rays.

The algorithm is not yet completely defined or implemented for ACCUM, BRIGHTOBJ, and RAMP mode observations, therefore the \*\_cal.fits output file for these modes will still contain cosmic ray hits.

### BACKCALC

This step computes a predicted background (sky plus thermal) signal level, based on models of the zodiacal scattered light and the telescope plus instrument thermal background. This step uses the BACKTAB reference table which contains the background model parameters. Results are written to the BACKEST1, BACKEST2, and BACKEST3 header keywords. The image data are not modified in any way. This step is not yet implemented.

### WARNCALC (User Warnings)

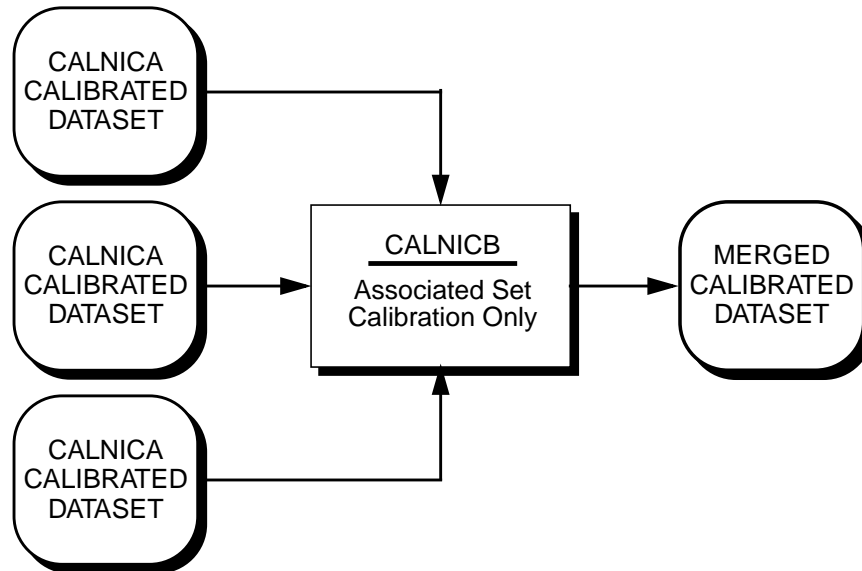
In this step various (as yet undetermined) engineering keyword values are examined and warning messages are generated if there are any indications that the science data may be compromised due to unusual instrumental characteristics or behavior. This step is not yet implemented.

---

## 16.4 Mosaicing: Calnicb

Observing strategies with NICMOS vary according to the nature of the target object and of the wavelength chosen for the observation. Extended objects may require mosaicing. Long wavelength observations will need chopping onto the sky to remove the telescope thermal background from the target frame. Multiple repetitions of the same exposure may be requested to improve cosmic ray removal, to control statistical fluctuations, and to increase the signal-to-noise on one target while avoiding saturation on another. Dither (mosaicing) and chop patterns of exposures are specified at the Phase II proposal level via the optional PATTERN parameter; multiple exposures at the same pointing are specified in Phase II by setting the Number\_of\_iterations to a value greater than one. All these options (which can also be set simultaneously) create an *association* of datasets (see “Associations” on page B-4).

The **calnicb** task (Figure 16.4) produces the combined, or *mosaiced*, image from the multiple images contained in a NICMOS association. The task also performs background subtraction and source identification on the images in the association.

**Figure 16.4:** Conceptual calnicb Pipeline

### 16.4.1 Input Files

Two levels of input data are needed by **calnicb**:

1. The *association table* (*assoc\_id\_asn.fits*): this is a table containing three columns with the list of members in the association and the relevant information on the association type, as given in Table 16.1.

**Table 16.1:** Columns of the Association Table (input to calnicb):

Column Name	Meaning
MEMNAME	Rootname ( <i>IPSSOOT</i> ) of each image in the association.
MEMTYPE	Role or type of each member: EXP-TARG = input exposure for target EXP-BCK $n$ = input exposure of $n$ -th background (for chop patterns) PROD-TARG = output product containing target PROD-BCK $n$ = output product containing $n$ -th background (for chop patterns)
MEMPRSNT	Flag indicating whether or not a member is present (needed by the STScI automatic pipeline processing).

The header of the *assoc\_id\_asn.fits* table also contains the setting of the keywords which control the background illumination pattern correction (ILLMCORR); the keywords read are: ILLMCORR (whether or not the correction must be performed) and ILLMFILE (reference file name for the illumination correction); these are explained in Chapter 15. At the time of

this writing (August 1997) we have not been able to measure any such spatial variations, and the ILLMFILES have not yet been populated.

2. The input images (*ippssoot\_cal.fits*): the science data images which are part of the association, as listed in the first column of the association table. The images are usually the calibrated outputs of **calnica**.
3. The support files (*ippssoot\_spt.fits*), containing the engineering information, so that **calnicb** can transfer this information to the output support files.

## 16.4.2 Output Files

**Calnicb** produces two levels of outputs:

1. An updated copy of the association table (*assoc\_id\_asc.fits*): this copy of the *assoc\_id\_asn.fits* file contains additional information about the processing that took place. The *assoc\_id\_asc.fits* file contains four additional columns, listed in Table 16.2.

**Table 16.2:** Additional Columns of the output Association Table:

Column Name	Meaning
BCKIMAGE	Flag indicating whether or not the image was used to compute the background.
MEANBCK	Values of the mean background for the image (DN/sec).
XOFFSET	X-offset (in pixels) of the image from the reference frame; a positive value means a positive offset of the image (not of the sources) relative to the reference.
YOFFSET	Y-offset (in pixels) of the image from the reference frame; a positive value means a positive offset of the image (not of the sources) relative to the reference.

Additional information contained in the header of the *assoc\_id\_asc.fits* table is the MEAN\_BCK keyword, which gives the constant background signal level subtracted from all images in the association.

2. One or more output mosaic images (*assoc\_idn\_mos.fits*): the number of output mosaic images depends on the pattern. The target field is always contained in the *assoc\_id0\_mos.fits* file. Patterns which involve chopping onto the sky to produce background reference images result in multiple *assoc\_idn\_mos.fits* files after processing through **calnicb**. In these cases, the target is still contained in the *assoc\_id0\_mos.fits* file; for each background position an additional mosaic, *assoc\_idn\_mos.fits* with  $n=1$  to 8, is created.
3. One *assoc\_id\_spt.fits* support file for each *assoc\_id\_mos.fits* file created.

### 16.4.3 Processing

The basic philosophy of the **calnicb** algorithm is to remove the background from each image after source identification, to align the images by calculating offsets, and to produce the final mosaic. The processing steps of **calnicb** can be summarized as follows:

1. Read the input `asn` table and input images.
2. Determine processing parameters from keyword values.
3. Combine multiple images at individual pattern positions.
4. Identify sources in the images.
5. Estimate and remove the background signal.
6. Create a mosaic image from all pattern positions.
7. Write the output association table and mosaic images.

The sections below discuss steps 2 through 6 in greater detail.

#### Processing Parameters

Header keywords from the input `*_cal.fits` images are read and evaluated in order to guide the processing. One set of keywords (Table 16.3) pertains to the association as a whole and therefore are read only once from the first input member image.:

**Table 16.3:** Common Keywords to all Association Datasets

Keyword	Purpose
INSTRUME	Check whether they are NICMOS data.
CAMERA	Camera number.
FILTER	Filter; if BLANK, the association is made of darks.
IMAGETYP	Type of image (EXT=external, DARK=dark frames, FLAT=flat-field images).
NUMITER	Number of iterations for each exposure.
PATTERN	Pattern used.
NUMPOS	Number of independent positions in the pattern.

A second set of header keywords (Table 16.4) are specific to each member of the association, and must be read from each single input image.:

Based on this information, an inventory is taken of what input images exist, where they belong in the pattern, how many there are at each pattern position, which images are from the target field, which ones are from background fields, and which output mosaic image each input image will eventually end up in. As part of the input process, the appropriate ILLMFILE reference file is loaded.

**Table 16.4:** Dataset-specific Keywords

Keyword	Purpose
PATT_POS	Position of the image in the pattern.
BACKEST $n$	Background estimates from <b>calnica</b> .
CRPIX $n$ , CRVAL $n$ CD $n_n$ CTYPE $n$	World Coordinate System (WCS) information (see Table 15.5).

### Combination of Multiple Exposures

If there is more than one image at any pattern position ( $\text{NUMITER} > 1$ ), the multiple images at each position are first registered and then combined into a single image. The coordinates (as determined by the WCS keywords) of the first image at a given pattern position are used as a reference for the registration. The offsets to all other images at that pattern position are first computed by comparing their WCS data, and then refined using a *cross-correlation* technique, down to a level of 0.15 pixels. The cross-correlation technique employs an algorithm which minimizes the differences between fluxes in the images. The computed offsets, in units of pixels, are recorded in the output association table. After determining the relative offsets, the images are aligned using *bilinear interpolation* and are then combined on a pixel-by-pixel basis. The combined pixel values are computed as a weighted mean of all unflagged (i.e.,  $\text{DQ} = 0$ ) samples, using the input image ERR values as weights. If three or more samples are present, iterative  $\sigma$ -clipping is performed to reject outliers. The number of samples used at each pixel and the total integration time are retained.

### Source Identification

The source identification step is used for excluding sources when the background in the images at each pattern position is estimated. The images at each pattern position are searched for pixels suspected to contain signal from a source. The mean signal level in the image is computed and pixels that are more than  $3\sigma$  above the mean are searched for. Spurious results, such as pixels containing cosmic-ray hits, are filtered out by searching neighboring pixels and only retaining those that have two or more neighbors that are also above the threshold. The DQ flag of the source-affected pixels is then set to 1024.

### Background Estimation and Removal

The background signal is estimated and removed from the images at each pattern position. Two types of background are subtracted from the images:

1. A *constant background* signal level, which is estimated from the images themselves.
2. The two-dimensional residual signal that may exist due to spatial variations in the thermal emission of the telescope and instrument; this is removed by subtracting the ILLMFILE reference image from each image (this second step can be turned on or off via the ILLMCORR keyword in the association table header). At the time of this writing (August 1997) we have not been

able to measure any such spatial variations, and the ILLMFILES have not yet been populated.

The *constant background* signal level is estimated and removed as follows.

1. With chop patterns, the mean and standard deviation of the signal in the image at each chop position is computed. In addition to excluding bad and source-flagged pixels, the calculation of the mean also uses iterative  $\sigma$ -clipping to reject outliers.
2. With dither-only patterns, or with multiple-exposure single pointings, the mean and  $\sigma$  of the target images is computed. The result for each image is compared to the estimate provided by **calnica**, which is stored in the BACKEST1 header keyword of each image. The value computed by **calnicb** is accepted if it is less than 5  $\sigma$  deviant from that of **calnica**, otherwise the **calnicb** value is assumed to be biased by the presence of sources and the **calnica** value is substituted for it.
3. The global mean and standard deviation of the background is computed from the values of each image, using iterative  $\sigma$ -clipping to reject outliers.
4. The final mean background value is subtracted from all images (both target and background images, if present).

### Mosaic Construction

Mosaic (MOS) images are created for each independent pointing within the pattern. For example, a combination DITHER-CHOP pattern will produce one mosaic image out of the dithered pattern at each CHOP location on the sky. Each mosaic image is created as follows:

1. The relative offsets between images within the mosaic are computed from their WCS information and refined using cross-correlation (as in the case of multiple exposures at each pattern position, see above). The first image in the list for each mosaic is used as a reference image.
2. An empty mosaic image is created with  $x$  and  $y$  dimensions large enough to encompass the maximum offsets in each direction.
3. Pixel values in the mosaic image are computed by combining samples from overlapping images. The individual images are aligned using bilinear interpolation and the value at a given mosaic pixel location is computed from the error-weighted mean of the samples at that location. Samples flagged as bad are excluded and, if three or more samples are present, iterative  $\sigma$ -clipping is used to reject remaining outliers. The number of samples retained for a given pixel and their total integration time is recorded in the SAMP and TIME images. If all samples are rejected for a pixel, the mosaic image SCI, ERR, SAMP, and TIME values are set to zero and a combination of all DQ flags is retained.

## 16.5 Grism Data Reduction: Calnicc

**Calnicc** is an IDL program designed for the automatic extraction of spectra from NICMOS multi-object grism images. The basic capabilities of the program are the following. Objects are identified on a direct image and classified as stars or galaxies using a neural network approach implemented as the SExtractor program (Bertin and Arnouts, 1996, *A & A Suppl.* 117, 393). The positions of the objects are used to extract spectra from a grism image of the same region. After extraction, the spectra are wavelength calibrated, flat-fielded, and flux calibrated. The extracted spectra are corrected for contamination from nearby objects. Subsequently, the extracted spectra are automatically searched for emission and absorption lines, and the continuum level of each spectrum is determined. Figure 16.5 shows the flow chart for **calnicc**.

### 16.5.1 Input Files

**Calnicc** expects two input files:

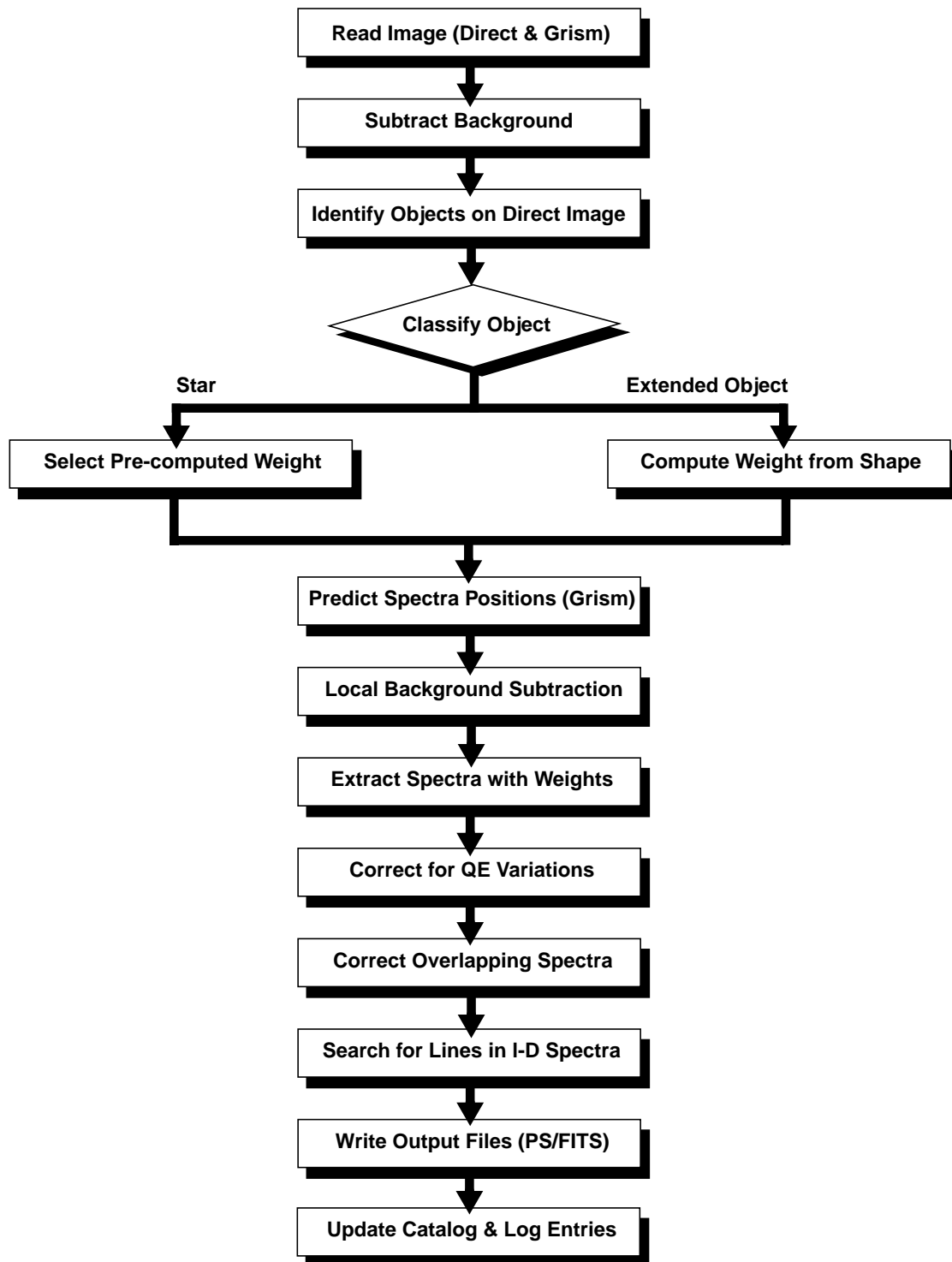
- *Direct image* of the target field (*\*\_cal.fits*); the image has been typically obtained with one of the NICMOS continuum filters, possibly at a wavelength within the range covered by the grism. The direct image given as input to **calnicc** is the calibrated output file of **calnica**.
- *Grism image* of the target field (*\*\_cal.fits*); this image has also been processed by **calnica**. By default, **calnica** does not flat-field grism images, because the flat-field is wavelength dependent and must be applied to the final extracted spectra. **Calnicc** provides the flat-fielding step.

### 16.5.2 Output Files

A number of outputs are generated by **calnicc**:

1. A FITS table containing the extracted spectra (*image\_tab.fits*). This file contains the successfully extracted spectra; it consists of a primary header and a series of table extensions, each extension consisting of a header and the associated table. There is one table extension for each spectrum extracted. The primary header of the file contains the relevant information regarding the observation, namely a subset of the keywords in the primary headers of the input direct and grism images. The table extension header contains keywords relevant for the individual spectrum; the keywords describe: the content of the table (the list of columns), the nature and position of the object, and the characteristics of the spectrum (line positions and fluxes, continuum level, etc.). The associated table contains five columns: the wavelength vector, the flux vector, and three vectors of the statistical, deblending, and total errors from the extraction process.

Figure 16.5: Flowchart of calnic Processing



2. Postscript files of the extracted spectra (*image\_n.ps*). The files contain graphical representations of the extracted spectra. One postscript file for each spectrum is generated, where n is a sequential number starting with 0.
3. Background image (*image\_po.fits*). This file is equivalent to the original grism image, except that all pixels used in the spectra extraction are replaced by the surrounding background level.
4. Background identification image (*image\_bg.fits*). This is an image where the pixels used in the background estimate calculation are highlighted.
5. The catalog of objects (*calnicc.cat*). This is a list of all the objects whose spectrum has been successfully extracted. For each object the catalog reports information about its position, nature, and characteristics of the extracted spectrum.
6. The log file (*calnicc.log*). The log file contains statistics on the **calnicc** processing.

### 16.5.3 Processing

**Calnicc** was developed using the Interactive Data Language (IDL) software. However the C program **SExtractor** (Bertin and Arnouts, 1996) is used for source object detection. Below a brief description of the algorithms used at each step of the processing is given, following the basic outline of the flow chart in Figure 16.5.

#### Object Detection and Classification

After both the direct and grism images have been read by the program, **calnicc** checks whether there is a background estimate in the header of the *\*\_cal.fits* images and subtracts it from the data, if present.

A third-party program, **SExtractor** is used then to detect and classify (point-like or extended) objects on the direct image. This program is thoroughly documented in the *SExtractor 1.0 User's Guide*. **SExtractor** does not use the data quality flags or the error arrays to perform the detection, and spurious objects may be introduced in the catalog. To remove spurious detections, the pixels where the object lies are compared to the quality flag array. If the ratio of bad pixels to the total number of pixels used by the object is above a user-defined threshold, the object is considered as spurious and not processed any further. In addition, the total flux within any object is compared to the quadratic sum of the error estimates for the individual pixels within the object. Again, a user supplied threshold for the significance is used to remove spurious or weak objects.

The grism images are also searched for additional sources. Grism images may yield objects which are located outside of the detector (and therefore are not present in the direct image), but have part of the spectrum on the grism image. Or objects which have most of the flux contained in a single spectral line; in this case, the usually short exposure time of the direct image might not be enough to detect the object, while the typically longer exposure time of grism image may allow the

detection. The grism image is prepared by replacing pixels used by previously detected objects' spectra with the background level. Subsequently, DAOFIND is used to find objects more than  $DAOTHRESH \times \sigma$  above the background, where  $\sigma$  is the rms of the image after removing the spectra. Since no zero point for the wavelength scale is known for those spectra, the objects' location is simply noted in the log file, but no attempts are made to extract their spectra.

### Background Subtraction

After source identification, an estimate of the two-dimensional background level is derived and removed from each image.

The grism image is not flat-fielded and the QE variations across the NICMOS detectors are strong, implying that a significant structure is present in an image of blank sky. The QE variations depend significantly on the wavelength, and the expected background in the grism image will depend on the spectrum of the background in space. The present version of **calnicc** uses a pre-computed model of the background in space, which will be replaced by measures of the blank sky when these data become available. The model used for the background estimate includes three different thermal sources:

- Thermal radiation from the HST itself.
- Thermal radiation from the NICMOS optics.
- Near infrared zodiacal emission.

These three components and their wavelength dependence are discussed in the *NICMOS Instrument Handbook*.

The model background image for each grism is stored in an associated background FITS file. This image is scaled to the local flux within a region around each spectrum on the grism image, and the rescaled background is subtracted from the image itself. The scaling factor is calculated by taking the mean flux values of an ellipsoidal region surrounding each spectrum (but excluding the spectrum itself), and dividing it by the mean of the background image in the same region. Pixels belonging to overlapping spectra from two or more objects are excluded from the computation of the scaling factor. The uncertainty in the background estimate is given by the square root of the sum of the errors divided by the square root of the number of pixels.

### Spectra Extraction

#### *Wavelength Calibration*

Both the dispersion relation and the deviation of the spectra from a straight line (distortion) are parametrized as third degree polynomials. The coefficients of the polynomials and the orientation of the spectra relative to the direction of the rows are contained in the reference file `grismspec.dat`. The dispersion relation is given by:

$$\lambda = a_0 + a_1x + a_2x^2 + a_3x^3 \quad \text{Eq. 16.2}$$

where  $x$  is the deflection in pixels relative to the position of the object in the direct image and  $\lambda$  is the corresponding wavelength.

The distortion of the spectra is parameterized as:

$$\Delta y = b_0 + b_1 r + b_2 r^2 + b_3 r^3 \quad \text{Eq. 16.3}$$

where  $r$  is the distance of a pixel  $(x,y)$  from the object of coordinates  $(x_o, y_o)$  and  $\Delta y$  is the deviation in pixels of the spectrum from a horizontal line. The alignment of the spectrum is taken into account by rotating the grism image around the object position  $(x_o, y_o)$  prior to the extraction. The distortions in the spectra are taken into account by introducing a corresponding distortion in the weights used for the extraction.

### *Flux and Error Bars*

Once objects on the images have been detected, their spectra can be extracted. Extraction of spectra is accomplished by using a weight matrix to calculate the flux vector for each wavelength. The flux is then given by:

$$F_j = \sum_i w_{ji} g_{ji} \quad \text{Eq. 16.4}$$

where the sum over the flux  $g_{ji}$  of all pixels at wavelength  $\lambda$  is performed with weights  $w_{ji}$ . The weights used to compute the spectra depend on the size of the objects. Two scenarios are handled by **calnicc**: point sources and extended objects (e.g., galaxies). For point sources, the weights are computed from simulated PSFs generated via the TinyTim software. The weights are optimal for point sources with flux-independent noise, namely, background dominated. For extended objects, the weight matrix is built from the direct image, under the assumption that the shape of the object is independent of wavelength. The size and orientation of the object is computed from the direct image using the moments of the image. The weight matrix is then created by summing up all the pixel values in a given row (fixed wavelength) of the grism image that fall within the ellipse defined by the size and orientation the object in the direct image.

The error estimate  $\epsilon_{ji}$  for each pixel is taken from the ERR array of the input grism image. The error estimate  $\epsilon_j$  for each wavelength is then the weighed quadratic sum over the errors of all pixels at constant wavelength.

### **Flatfielding of Spectra**

After the spectra are extracted, the fluxes are corrected for the variations of the quantum efficiency of the detector (flatfielded). The QE variations depend both on the wavelength and on the position of the object on the detector. Because of this wavelength dependence, the flatfielding cannot be performed before the spectra are extracted. The correction factors are derived through interpolation from a set of monochromatic flatfield images stored as a single FITS file called `nicmosFF.fits`.

### **Deblending of Overlapping Spectra**

Since the NICMOS grisms are slitless, overlaps among different spectra are likely to happen. The strategy of observing the same target at different telescope roll angles helps removing overlap in many instances. In addition, **calnicc** has been designed to *deblend* the spectra—to remove or minimize contamination of one spectrum from another.

The deblending algorithm is described in detail in the **calnic** manual. The basic requirement for the algorithm to work is that, at each wavelength, different spatial portions of the spectrum to be deblended have different levels of contamination. The deblending algorithm relies on the assumption that the shape of the object is the same at all wavelengths. The deblending procedure produces also an error estimate which is reported in the output FITS table and indicated in the postscript file containing the spectrum.

---

## 16.6 Recalibration

This section is intended to help you decide whether your data were calibrated with optimal calibration reference files and to help you decide whether you need to recalibrate your data.

### 16.6.1 Why Recalibrate?

Many users will find that the calibrated data produced by the OPUS pipeline—the standard calibration—are adequate for most scientific applications. The **calnica** pipeline calibrated your data with the most up-to-date, instrument configuration-specific reference files available at the time of the observation. However, updated or timely reference files sometimes do become available after your data were processed. Improved software for calibration (both **calnica** and **calnicb**) may also become available as our understanding of the instrument performance increases with on-orbit experience. This is especially true for data obtained during the first few months of the NICMOS lifetime in orbit. Particularly if you notice unusual features in your data, or if your analysis requires a high level of accuracy, you may wish to determine whether a better set of calibration reference files exist than were used to process your data. If better files are available or the calibration software changes significantly, you may choose to recalibrate your data using the new files or software.

Finding that a calibration reference file has changed since your data were calibrated doesn't always mean that you have to recalibrate. The decision depends very much on which calibration image or table has changed, and whether that kind of change to your data is likely to affect your analysis in a significant way. Before deciding to recalibrate, you might want to retrieve the new recommended reference file or table and compare it to the one used to calibrate your data at STScI in order to determine if the differences are important. You can use the table tools in the STSDAS **ttools** package to manipulate and examine calibration tables; images can be manipulated in the same way as your science data.

## 16.6.2 Recalibrating the Data

This section describes the mechanics involved in actually recalibrating a dataset. The basic steps in recalibrating a dataset are:

1. Assemble any necessary reference files or tables and your raw data files.
2. Set the desired calibration switches and reference file name keywords in the *primary header* of your raw (`*_raw.fits`) data file. These determine which steps will be executed by the calibration software and which reference files will be used to calibrate the data.
3. Run the calibration software.

### Assembling the Input Files

In order to recalibrate your data, you need to retrieve *all* of the reference files and tables that are used by the calibration steps you want to perform. The source of these files is the Calibration Database (CDBS) at STScI. A complete description of how to retrieve the reference files is given in Chapter 1.

### Setting the Calibration Parameters

The calibration software is completely data-driven, meaning that the calibration steps to be carried out are determined by the values of the *calibration switches* and the *calibration reference files* keywords contained in the primary header of the file to be processed. An important step is then to set the calibration switches and reference file keywords in the primary header of your raw data file (`*_raw.fits`) to reflect how you want the data recalibrated and which reference files you want to use at each step in the process. This is done most easily with the **chcalpar** task in the **hst\_calib.ctools** package of STSDAS or with the **hedit** task in the IRAF **images** package.




---

The calibration switch keywords reside only in the primary header of NICMOS FITS files: it is critically important to *specify extension number zero explicitly* when passing file names to tasks like **chcalpar** or **hedit** to modify these keywords. For example, to modify calibration keywords in the file `n3xe01bhm_raw.fits`, be sure to use the complete name `n3xe01bhm_raw.fits[0]` as input to these tasks. If you do not specify extension zero explicitly, the keywords you modify will end up getting written into the first extension header instead, where **calnica** will *not* find them.

---

The **chcalpar** task takes a single input parameter—the name(s) of the raw data files to be edited. When you start **chcalpar**, the task automatically determines that the image data are from NICMOS and opens a NICMOS-specific parameter set (*pset*) that will load the current values of all the calibration-related keywords. To edit the calibration keyword values:

1. Start the **chcalpar** task, specifying the image(s) in which you want to change keyword values. If you specify more than one image, for example using wildcards, the task will read the initial keyword values from the first

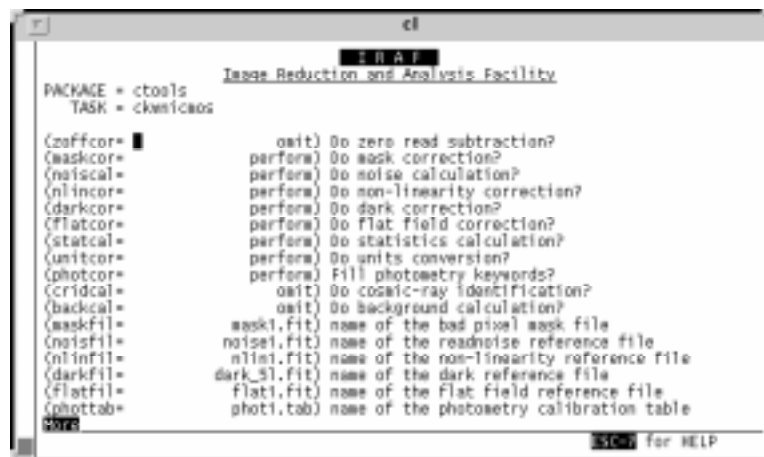
image in the list. For example, you could change keywords for all NICMOS raw science images in the current directory (with initial values from the first image), using the command:

```
ct> chcalpar n*raw.fits
```

An example of using **chcalpar** on a single NICMOS file is shown in Figure 16.6.

2. After starting **chcalpar**, you will be placed in **eparam**—the IRAF parameter editor, and will be able to edit the set of calibration keywords. Change the values of any calibration switches, reference files or tables to the values you wish to use for recalibrating your data.
3. Exit the editor when you are done making changes by typing `:q` two times. The task will ask if you wish to accept the current settings. If you type “y”, the settings will be saved and you will return to the IRAF `c1` prompt. If you type “n”, you will be placed back in the parameter editor to redefine the settings. If you type “a”, the task will abort and any changes will be discarded.

**Figure 16.6:** Editing Calibration Keywords with **chcalpar**



## Running the Calibration Software

After you change the header keyword values for your raw data files, you are ready to recalibrate your data. To run **calnica**, type the name of the task followed by the names of the input raw data file and desired output calibrated data file. For example, to recalibrate the dataset `n0g70106t`, you could type:

```
ni> calnica n0g70106t_raw.fits n0g70106t_cal.fits
```

**or:**

```
ni> calnica n0g70106t " "
```

To run **calnicb** the name of the association table must be given as input:

```
ni> calnicb assoc_id_asn
```

To run **calnicb** on a subset of the \*\_cal.fits files, it is sufficient to edit the \*\_asn.fits table and remove the undesired files.



---

The calibration routines **calnica** and **calnicb** will not overwrite an existing output file. If the calibration tasks are run in the directory where the original calibrated files are located, a different output file name must be specified.

---

### 16.6.3 Calculating Absolute Sensitivity

The PHOTCORR step in **calnica** does not alter the values of the data (which are always in countrates in the calibrated file), but only writes the information necessary to convert countrates to fluxes into the header of the file, i.e., sets the values of PHOTFLAM, PHOTFNU, PHOTPLAM, PHOTBW, PHOTZPT, and PHOTMODE. Unless you wish to recalculate the absolute sensitivity for your observation (e.g., because a more recent throughput estimate exists for your observing mode), there is no need to recompute these values and you can set PHOTCORR=OMIT in the recalibration.