1997 HST Calibration Workshop
Space Telescope Science Institute, 1997
S. Casertano, et al., eds.

NICMOS Data Processing Software in STSDAS

Ivo C. Busko

Space Telescope Science Institute, Baltimore

1. Motivation

Until now the standard form of accessing observational data from the Hubble Space Telescope was through GEIS (or STF) files. The GEIS format was designed to allow storage of multiple "images" in a single file. The files however must store the same kind of information, e.g. science pixels, data quality flags, etc. If a given instrument generates, e.g., both science and data quality arrays, they must be stored in separate files.

The new HST instruments STIS and NICMOS generate several "pixel" arrays from each exposure. Besides the usual science and data quality arrays, there is an *error* array, and NICMOS includes additional *exposure time* and *number of samples* arrays. It would be impractical to store all this associated information in separate files. In order to keep all information pertaining to a single exposure packed together in a single file, the standard data format adopted for STIS and NICMOS data files is FITS with multiple extensions.

Each associated "chunk" of information is named an IMSET, and a single file can store multiple IMSETs, each one identified by a number stored in the EXTVER keyword in each FITS extension header.

The existing IRAF and STSDAS tasks can operate upon individual FITS extensions as individual images with no problems, but in this way it becomes extremely cumbersome to properly propagate the error, data quality and other information from the input IMSETs into the result. Thus the need for brand-new tasks that could perform basic image processing and at the same time automatically propagate the associated information.

2. The `mstools` package

Most of the processing tasks that can handle IMSETs as a whole are included in the `mstools` package in STSDAS. There are also tasks to deconstruct and reassemble multi-IMSET FITS files into isolated IMSETs, and tasks that perform a variety of operations on individual FITS extensions in multi-IMSET files. All tasks support NICMOS *and* STIS files. The tasks included in STSDAS v.2 are:

ecdel	Deletes an entire class of FITS extensions from a FITS file.
ecextract	Selects all extensions of one selected type from a FITS file.
extdel	Deletes single extensions from a FITS file.
msarith	Image arithmetic with NICMOS and STIS files.
mscombine	Combines NICMOS or STIS files using gcombine.
msdel	Removes one IMSET from a FITS file.
mssort	Sorts a FITS file to get all extensions of like number together.
msjoin	Joins files containing single IMSETs into one file.
mssplit	Splits NICMOS or STIS IMSETs out into separate files.
msstatistics	Extended gstatistics for OIF, GEIS, NICMOS and STIS files.

The package also includes *psets* that help in selecting/masking specific data quality bits. These are used by tasks such as `mscombine` and `msstatistics`.

3. The `nicmos` package

This package includes NICMOS-specific tasks such as the calibration pipelines and some specialized display tasks. It also includes tasks that are used solely for the purpose of generating calibration reference files. Some of these tasks support other instruments in addition to NICMOS, and are actually installed in the `ctools` package (the entries here are links). The tasks included in STSDAS v.2 are:

calnica	Pipeline calibration for single NICMOS images.
calnicb	Pipeline calibration for NICMOS associations.
markdq	Marks DQ flags on a displayed image.
msbadpix	Detects bad pixels in STIS and NICMOS images.
msreadnoise	Measures readout noise in STIS and NICMOS images.
msstreakflat	Processing of Earth streaked images for WFPC and NICMOS.
ndark	Builds NICMOS DARKFILE calibration reference file.
ndisplay	Displays a science image with DQ flags superimposed.
nlincorr	Builds NICMOS NLINFILE calibration reference file.
pstack	Plots a stack of pixel values from a MULTIACCUM image.

As with the `mstools` package, the `nicmos` package includes *psets* that help in selecting/masking specific data quality bits.

4. Basic image processing

For generic image processing of NICMOS images with full propagation of the extra arrays into the output, two tasks are available in STSDAS v.2:

4.1. `msarith`

This task works *only* with NICMOS and STIS files, it does not replace the standard `imarith` task in the IRAF `images` package. The task was designed to look and perform in a similar way as `imarith` but with added features to take into account the peculiarities found in NICMOS and STIS data:

- It fully propagates all extra arrays into the output. The rules for propagation are described in Table1.
- It can operate on all IMSETs in the input files or just a subset of them.
- In NICMOS images each pixel has its own associated exposure time, which can differ from pixel to pixel. The task takes this factor into account when performing operations in which the factor can be of significance, e.g., when adding two calibrated images (whose pixels store count rate instead of raw counts).

4.2. `mscombine`

This task can be used to combine NICMOS or STIS exposures using either an average or median. A variety of clipping/cleaning algorithms are provided, including discarding pixels based on any data quality flag combination. In STSDAS v.2 the task is actually an IRAF CL script that invokes the `gcombine` task to operate upon the input files' IMSETs. Thus most of the features and capabilities of `gcombine` are available to `mscombine`.

```
operand1 = "file1.fits"    >File template/list or constant + error
      op = "+"             >Operator
operand2 = "file1.fits"    >File template/list or constant + error
  result = "result"        >Resultant file name/list or directory path
  (list1 = "1,2,21,10")    >IMSETs in operand1
  (list2 = "3")            >IMSETs in operand2
  (crate = no)             >Count rate ? (NICMOS only)
(divzero = 0.)             >Replacement value for division by zero
(verbose = 1)              >Verbosity level
(version = "23May97")      >Date of installation
   (mode = "al")
```

Figure 1. `msarith` task parameters. Any combination of input file lists/IMSETs can be supplied to the task. It chooses automatically the appropriate mode (raw counts vs. count rate) based on information retrieved from the file header. If that information is not present or is contradictory, the "count rate" switch forces the task to work in the designated mode.

Table 1. Rules for operations performed by `msarith`.

Operation/ 2nd operand	EXTNAME				
	SCI	ERR	DQ	TIME	SAMP
add / image	add[1]	comb.[1,3]	OR	add	add
sub / image	sub	comb	OR	copy[4]	copy
mult / image	mult	comb	OR	copy	copy
div / image	div	comb	OR	copy	copy
add / const.	add	comb.	—	—	—
sub / const.	sub	comb.	—	—	—
mult / const.	mult	comb.	—	mult[2]	—
div / const.	div	comb.	—	div[2]	—

(1) if pixels are in raw counts - add raw counts.
if pixels are in count rate - translate count rate to counts in both input images; add counts and integration times; translate result back to count rate.
(2) if pixels are in raw counts - multiply/divide time array by constant.
if pixels are in count rate - copy time from input image.
(3) errors are combined in quadrature
(4) copy from first operator into result

```
        input = "@list1.lst"    >List of file names to combine
       output = "result"        >Output file name
      (dqbits = "")             >DQ bits to reject input pixels (pset)
     (nsmod_e = no)             >Use noise model for computing errors ?
      (reject = "ccdcrrej")     >Type of rejection
     (combine = "average")      >Type of combine operation
      (weight = "none")         >Type of weighting scheme
     (nsmod_w = no)             >Use noise model for weighting ?
       (blank = 0.)             >Output value when zero pixels survive
       (scale = "median")       >Image scaling
        (zero = "none")         >Image zero level offset
     (statsec = "")             >Image section for computing statistics
     (expname = "SAMPTIME")     >Image header exposure time keyword
  (lthreshold = INDEF)          >Lower threshold
  (hthreshold = INDEF)          >Upper threshold
        (nlow = 1)              >minmax: Number of low pixels to reject
       (nhigh = 1)              >minmax: Number of high pixels to reject
       (nkeep = 1)              >Min. to keep (pos) or max. to reject (neg)
       (mclip = yes)            >Use median in clipping algorithms ?
      (lsigma = 3.)             >Lower sigma clipping factor
      (hsigma = 3.)             >Upper sigma clipping factor
     (rdnoise = "30")           >Readout noise (electrons)
        (gain = "5")            >Gain (electrons/DN)
      (snoise = "0.")           >Sensitivity noise (fraction)
     (tempdir = "tmp$")         >Directory for temporary files
     (version = "07May97")      >Date of installation
        (mode = "al")
```

Figure 2. `mscombine` task parameters. The parameters are basically the same as the `gcombine` parameters but with no provisions for input/output of error maps or masks, since these are already stored internally to the input/output IMSETs. An important new parameter is the data quality flags pset, which enables `mscombine` to discard pixels based solely on their DQ value.

5. Image statistics

Task `msstatistics` is an extension of `gstatistics`, in the sense that it supports not only all file formats already supported by `gstatistics` (OIF and GEIS) but can process the new NICMOS and STIS formats as well. Its parameter list resembles `gstatistics`' with some additional parameters used to handle the extra complexities of the new FITS files. Two notable additions in relation to `gstatistics` are:

- Two new statistics were added to the list supported by `gstatistics`: weighted mean and weighted variance of the pixel distribution. These are possible for NICMOS and STIS exposures since they always have an associated error map.
- Pixels can be rejected out from the statistical computations based on specific bits set in their data quality flags.

```
      input = "f.fit,g.hhh"    >Images template/list
      (masks = "")             >Masks template/list
        (dqf = "")             >Data Quality File template/list (OIF/GEIS only
     (groups = "*")            >Groups/IMSETs to be processed
      (stats = "npix,mean,stddev") >Statistics to be computed
      (lower = INDEF)          >Lower cuttof
      (upper = INDEF)          >Upper cuttof
     (gaccum = no)             >Accumulate across groups ?
      (egstp = "")             >Statistics results of last image (pset)
    (namecol = 25)             >columns used by file name in output printout
      (dqon  = yes)            >Turn on pixel masking by Data Quality bits ?
     (dqbits = "")             >Generic Data Quality bits (pset)
  (nsstatpar = "")             >NICMOS and STIS parameters (pset)
    (wfdqpar = "")             >WFPC parameters (pset)
    (version = "16Apr97")      >Date of installation
       (mode = "al")
```

Figure 3. `msstatistics` task parameters. The parameters are basically the same as the `gstatistics` parameters but with additional ones that take care of specific aspects of the NICMOS and STIS file structure.

```
# image                          npix        mean      stddev
#
u2a31504t.c0h[1]               640000.     299.373     108.757
u2a31504t.c0h[2]               640000.     1396.86     426.502
u2a31504t.c0h[3]               640000.     1438.88     455.329
u2a31504t.c0h[4]               640000.     1432.47     486.197
n3ug06cyr_ima.fits[SCI,1]       13764.    0.271452    0.114495
n3ug06cyr_ima.fits[ERR,1]       13764.   0.0158293  0.00723336
n3ug06cyr_ima.fits[TIME,1]      13764.     575.952          0.
n3ug06cyr_ima.fits[SAMP,1]      13764.         26.          0.
n3ug06cyr_ima.fits[SCI,2]       14182.    0.272135    0.117022
n3ug06cyr_ima.fits[ERR,2]       14182.   0.0165362  0.00757553
n3ug06cyr_ima.fits[TIME,2]      14182.     543.953          0.
n3ug06cyr_ima.fits[SAMP,2]      14182.         25.          0.
n3ug06cyr_ima.fits[SCI,3]       14617.    0.273101    0.120651
n3ug06cyr_ima.fits[ERR,3]       14617.   0.0173648  0.00796676
n3ug06cyr_ima.fits[TIME,3]      14617.     511.953          0.
n3ug06cyr_ima.fits[SAMP,3]      14617.         24.          0.
```

Figure 4. Example output from `msstatistics`. File types can be mixed at will in the input list. There are parameters for selecting which IMSETs and which extensions within each IMSET must be operated upon.