

Calstis2: Cosmic Ray Rejection in the STIS Calibration Pipeline

R. Shaw, J. C. Hsu

and the Spectrographs Group pipeline block: S. Baum, I. Busko (SSG), J. Christensen, H. Ferguson, P. Goudfrooij, P. Hodge (SSG), S. Hulbert, R. Katsanis, and M. McGrath

May 4, 1998

ABSTRACT

*We describe calstis2, the **calstis** calibration module which combines CRSPLIT and repeated exposures to produce a single, cosmic ray rejected image. Cosmic ray rejection in the STIS pipeline employs a noise model and parameterized rejection criteria to identify and exclude discrepant values in forming the output image. The **calstis** pipeline is able to perform this cosmic ray rejection because the individually commanded exposures are associated together into a single dataset by TRANS and generic conversion. The functionality of calstis2 is also available as a host-level application and as a task in STSDAS to allow users to re-perform the cosmic ray rejection outside the pipeline with different input parameters.*

*This document updates and supersedes STIS ISR 96-18, and describes the functionality that was implemented for **calstis** V1.7.*

1. Introduction

In this ISR we describe the portion of the STIS calibration pipeline known as *calstis2*, which processes multiple CRSPLIT and repeated exposures to produce a single cosmic-ray (CR) rejected output image. We present below the motivation for performing cosmic ray rejection in the pipeline, and in the next section we briefly describe the association and packaging of CRSPLIT and repeated exposures by the ground system, which allows the processing of these exposures as a unit through **calstis**. We describe in §3 the place of *calstis2* within the context of the overall STIS calibration pipeline, including a description of the CRR calibration reference file which provides the control parameters for *calstis2*.

Section 4 gives a detailed description of the algorithm employed for cosmic ray rejection, the output product for which is described in §5. We briefly describe the error trapping in §6, and in §7 we describe how to perform CR rejection on STIS data using independent applications either within or outside of STSDAS.

Motivation

One of STIS's three detectors is a CCD operating from the near-UV to the near-IR (2500Å-11000Å). Long slit CCD spectroscopy comprises ~40% of the prime science conducted with STIS, and the STIS CCD is frequently used in parallel observations, either in imaging mode or with a first order grating to produce slit-less spectroscopic images of the sky. As with WF/PC and WFPC2, CCD images are subject to cosmic rays that deposit substantial charge in one or (usually) more contiguous pixels, compromising the usefulness of single-exposure CCD observations. The STIS Proposal Instructions support splitting CCD exposures in up to multiple associated exposures (specified in the Phase 2 proposal by the `Number_of_Iterations=N` parameter or by the `CR-SPLIT=N` optional parameter). These multiple exposures will all have roughly the same exposure time (to within ~20%).

In the WF/PC and WFPC2 calibration pipelines the individual `CRSPLIT` exposures are passed through the pipeline individually and sent individually to the archive. GO's and archival users must extract the exposures and combine them to reject cosmic rays before a scientifically (or cosmetically) useful image is produced. But performing the cosmic ray in the pipeline itself renders the products delivered to the user and the archive more immediately useful. For STIS, the motivation to reject cosmic rays in the pipeline and deliver cosmic-ray free end products to the user and archive is doubly strong: STIS spectroscopic data are not only processed through basic CCD two-dimensional image reduction (see the description of *calstis1* by Hodge & Baum, 1995) but are also spectroscopically calibrated, with 2-D and 1-D spectral extractions performed on the data (see Hodge et al. 1998).

Goals of Pipeline CR-Rejection

The goals of cosmic-ray rejection in the pipeline are to produce an output product which is significantly less compromised by cosmic rays than the input product, which can be immediately used for scientific investigation in most cases, and which significantly improves the value of the quick-look products in all cases. It is not, however, realistic within the confines of a pipeline to produce the optimum cosmic ray rejected product for all observational and scientific scenarios. We fully anticipate that many users (perhaps more than 20%) will wish to re-perform the cosmic-ray rejection step on their data or will perform additional CR-rejection (e.g., flagging residual cosmic rays based on shape) on the pipeline-delivered CR-rejected image. For this reason, the **calstis** pipeline also processes each of the `CRSPLIT` or repeated exposures through *calstis1*, creating an

intermediate output file which contains the original exposures calibrated through flat-fielding. In addition, a task that implements *calstis2* is available both as a host-level application, and as an IRAF/STSDAS task which users can run outside of the pipeline environment, changing the input parameters to better optimize the rejection process for their own specific data and science aims.

2. Associations and CRSPLIT Data

Data taken using the parameters `Number_of_Iterations=N` or `CR-SPLIT=N` (where $N > 1$) on the Phase II proposal logsheet will be associated together by TRANS (the software system which TRANSLates the Phase II proposal instructions into commandable units) for OPUS (see MacKenty & Baum 1996). During the generic conversion, OPUS will package the individually commanded but associated multiple exposures into a single dataset, with a name of the form *rootname_raw.fits*¹. For STIS, each individual exposure, with its associated data quality (DQ) and error (ERR) arrays, form a triplet known collectively as an IMSET. The IMSETs for each CRSPLIT or repeated exposure populate successive image extensions in the file. Each extension will have its own image extension header with exposure specific header keywords. The primary header file contains keywords describing the observation set as a whole, while the extension headers contains exposure specific information. For example, the primary header keyword `TEXP-TIME` gives the sum of the exposure time in all the exposures, while the image extension header keyword `EXPTIME` gives the exposure time in the specific single exposure in its extension. The primary header keyword `NRPTEXP` indicates the number of such exposures in the file. (See the *HST Data Handbook* for additional details.) The calibration switch header keyword `CRCORR`, which indicates that a cosmic-ray rejected image should be produced, will be set to `PERFORM` whenever `CRPSLIT > 1` or if `NRPTEXP > 1`.

3. The Place of Calstis2 in the STIS Pipeline

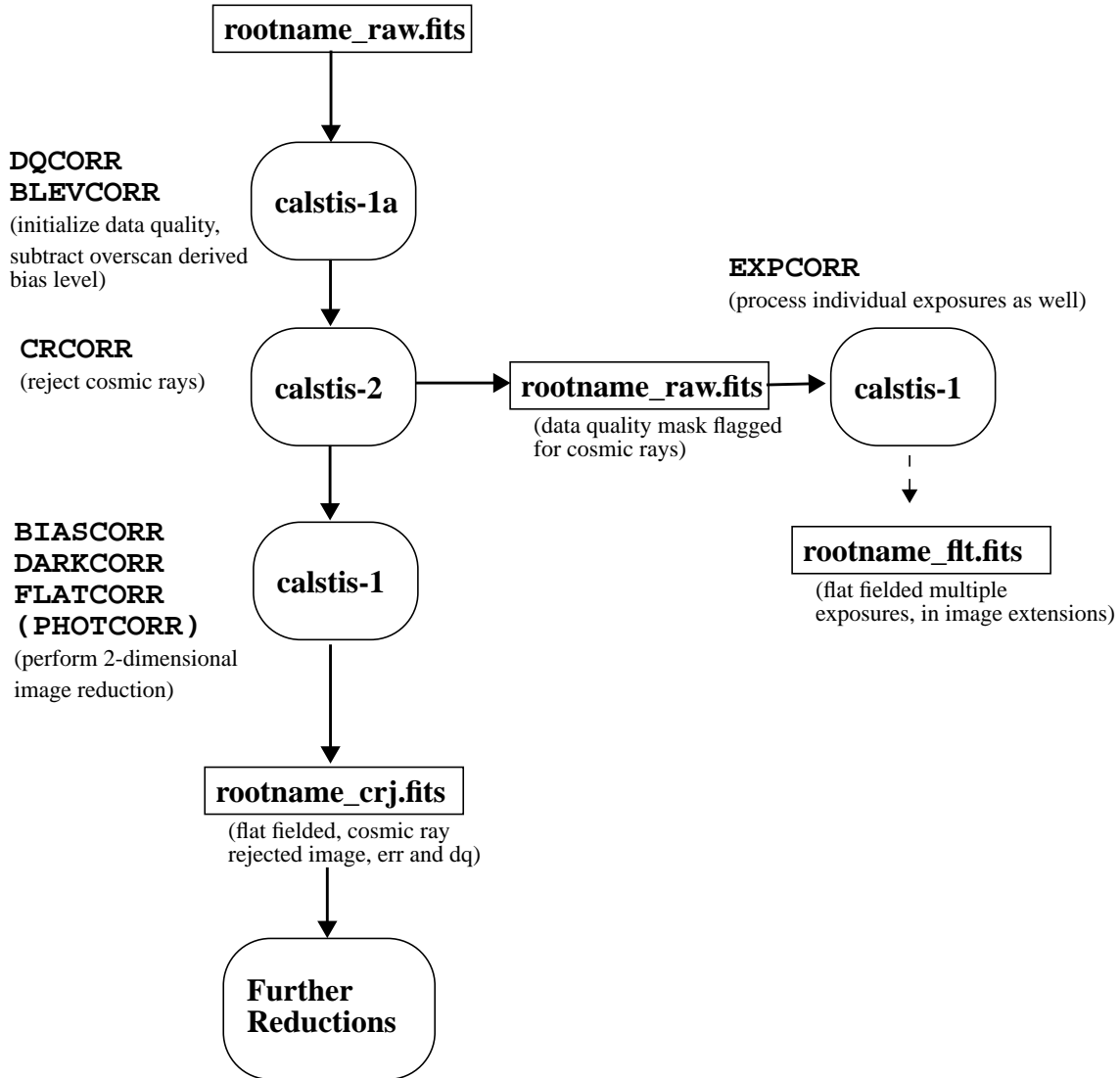
As described by Hodge et al. (1998), **calstis** consists of several individual executable elements, dubbed *calstis-N*, where “N” numbers nearly a dozen. The CR-rejection module of **calstis** is dubbed *calstis2*. *Calstis2* will be executed if the `CRCORR` header keyword is equal to `PERFORM`.

Figure 1 shows the flow of data through the **calstis** pipeline for data with `CRSPLIT` or `NRPTEXP > 1`. The combination of the multiple exposures into a single cosmic-ray rejected frame is performed early in the flow: after each exposure has had its data quality file initialized and the overscan bias level subtracted (`BLEVCORR`), but prior to the subtraction of the bias (`BIASCORR`) and dark (`DARKCORR`) frames, and the application of the

1. The dataset names have three main parts: a *rootname*, a *rootname suffix* (such as `_raw` or `_crj`), and a *file type* (which is always `.fits`). Hereafter, the file names will be referred to by the suffix only.

flat-field (FLATCORR). The CR-rejected, bias-level subtracted image is then passed through the remainder of the two-dimensional image reduction (*calstis1*) to produce a flat-fielded, CR-rejected image (with the suffix_crj). This CR-rejected flat-fielded image is then passed through the subsequent processing steps in **calstis**: geometric correction for imaging data and 2-D and/or 1-D spectral extraction and calibration for spectroscopic data.

Figure 1: Flow of CRSPLIT and NRPTXP>1 data through **calstis**



The CR-rejection step was placed early in the flow for two reasons. First, this is the optimal place to perform the CR-rejection, as the data values (DN) have not yet been scaled by the flat field response. If the flat fields exhibit the effects of vignetting or other strong large scale spatial variations, then performing CR-rejection following flat fielding exhibits poorer results; the flat fielding increases the dispersion of the DN values in

regions of low flat field response reducing the effectiveness of the CR-rejection algorithm. Second, by performing the CR-rejection prior to the bulk of two-dimensional image reduction, we assure that the errors produced in the output file are correct. If we did the CR-rejection following flat fielding, the errors calculated as the individual exposure errors added in quadrature would be incorrect, as they would incorrectly scale the errors contributed by the calibration reference files. Calibration contributes to the errors in a non-linear fashion and those errors cannot be separated from the Poisson errors due to counting statistics following calibration.

If the calibration switch `EXPSCORR` (in the primary header) is set to `PERFORM`, the pipeline will process each IMSET in the `_raw` file through `calstis1`, outputting an intermediate file, `_flt`, containing the flat-fielded individual exposures in successive IMSETs of a single file. This file will not be passed through the subsequent calibration steps, but will be retained as an intermediate data product to allow users to examine the effects of the pipeline cosmic-ray rejection.

Control Parameters

Control parameters for `calstis2` are read from both a calibration reference file (CRR) and from the primary header of the input science (`_raw`) file. The keywords to be read, and those that are updated after processing, are summarized in Table 1.

Table 1: Image Header Keywords Accessed by `calstis2`

Keyword	Description	Location	Populated by	Output Updated?
ATODGAIN	Calibrated gain, where $DN = \text{counts} / \text{ATODGAIN}$	Primary header	<code>calstis1</code> (read from reference file)	No
CRCORR	Calibration switch governing CR-rejection in the pipeline	Primary header	Generic conversion	Yes
CRREJTAB	Reference file containing control parameters	Primary header	Generic conversion	No
CRSPLIT	Number of CRSPLIT IMSETS in the file	Primary header	Generic conversion	No
EXPTIME	Integration time for each CRSPLIT exposure (seconds)	Extension header	Generic conversion	Yes
NCOMBINE	Number of IMSETs combined.	Extension header	<code>calstis2</code>	Yes
NEXTEND	Number of extensions in this file	Primary header	Generic conversion	No
NRPTXP	Number of IMSETs to combine.	Primary header	Generic conversion	
READNSE	Read noise in DN	Primary header	<code>calstis1</code> (read from reference file)	No
REJ_RATE	Average incidence of CRs detected per second during exposure	Primary header	<code>calstis2</code>	Yes

Table 1: Image Header Keywords Accessed by calstis2

Keyword	Description	Location	Populated by	Output Updated?
SKYSUM	Sum of sky values from input IMSETs	Primary header	<i>calstis2</i>	Yes
TEXPTIME	Total exposure time in observation set (seconds)	Primary header	Generic conversion	No

The Cosmic Ray Rejection (CRR) calibration reference file contains the parameters which control the flow of *calstis2*. The values of those parameters may be set differently depending on the number of input IMSETs and depending on the exposure time of each image. The complete definition of the CRR table may be found in ICD-47, but a summary of the column attributes are summarized in Table 2 below. *Calstis2* selects the appropriate row of this table based first on the value of the NRPTXP keyword, and secondly on the value of TEXPTIME/NRPTXP. Specifically *calstis2* finds all rows with an exact match between NRPTXP and the value in the CRSPLIT column, and then selects the row in which the value of mean exposure time column (MEANEXP) is closest to but still greater than TEXPTIME/NRPTXP.

Table 2: Columns in the CRR Reference File

Column Name	Data Type	Units	Description	Permitted Values ^a or Examples
CRSPLIT	I	—	Number of images to combine	[2–8] ^b
MEANEXP	R	sec	Mean exposure time of all the CRSPLIT images	0.1 < MEANEXP
SCALENSE	A8	(percent)	Multiplicative scale factor, in percent, applied to expected noise	1.0
INITGUES	A8	—	Method for forming the initial image	[minimum median]
SKYSUB	A4	—	Method of sky subtraction	[none mode]
CRSIGMAS	A20	σ	Rejection threshold(s)	“6,4.5,3”
CRRADIUS	R	pixel	Distance over which sympathetic rejection applies	1.0
CRTHRESH	R	—	Relative threshold for sympathetic rejection	1.
BADINPDQ	I	—	Integer value corresponding to DQ bits to mask for rejection	0
CRMASK	L1	—	Flag DQ-rejected pixels in _flt IMSETs?	[yes no]

- a. Permitted values are enclosed in square brackets; valid choices are separated by vertical bars, and ranges are indicated with a dash.
- b. Limit of CRSPLIT images is 8, but the number of IMSETs for repeated observations is limited only by the number of images that an IRAF application can open at once.

4. The CR Rejection Algorithm

The algorithm used in *calstis2* is similar to that employed in the STSDAS task **wfpc.crrej**. The basic philosophy is to take multiple exposures of the same field and combine the images by rejecting values in each pixel stack that deviate from the true image by some amount that is based on a global noise model. The CR-rejection for STIS differs from **wfpc.crrej** mainly in that it outputs an error array for the CR-rejected image, based on the input error arrays, and *calstis2* can accommodate a variable sky background.

Calstis2 does the following specific things:

1. Collect a stack of images to be combined.
2. Form an initial guess image.
3. Reject outliers at each pixel, using the guess image and the noise model.
4. Iterate, using different (usually decreasing) rejection thresholds to produce a new guess image and new summed image at each stage.
5. Output the final CR-rejected image (`_crj`), including the science, data quality and error extensions; and update various header keywords.
6. Update the data quality arrays of the IMSETs in the `_flt` files to indicate where outliers have been rejected.

The algorithmic details of each step are summarized in the subsections below.

Form Image Stack for Combination

This step performs a simple input of the science data and data quality flags for each image into a stack. The science values are scaled to the total exposure time (read from the `TEXPTIME` primary header keyword) of the multiple series on input. The `EXPTIME` extension header keyword gives the exposure time of each exposure in the stack.

If `sky=mode`, a sky value is calculated as the mode in all pixels for each input image and subtracted from each pixel value prior to placement in the stack. If `sky=none`, no background is subtracted. Note that the sum of the sky values is added back to the final `_crj` image.

Form Initial Guess Image

The method for forming the initial guess image is controlled through the parameter `initial`, which takes the values “median” or “minimum” to indicate that the minimum or the median value in the stack at each pixel is to be used. Only input pixels whose `badinpdq` data quality flags are not set (determined by the performance of a bitwise logical AND between `badinpdq` and the input pixel DQ flag) are used to form the guess.

That is, bad pixel masking occurs prior to forming the initial guess image and to the outlier rejection step. This capability prevents bad data in a single input image from contaminating the output product.

Iterative Rejection

A CR-rejected, summed output science image is created by setting the value at each pixel to the average values of all good pixels in the stack whose values are within plus or minus `sigmas*noise` of the initial guess image. Deviant (out of range) stack pixels are flagged as cosmic ray impacted by setting their stack data quality flags to $2^{13} = 8192$.

The model for **noise** (in DN) is computed as:

$$\mathbf{noise} = \text{SQRT} [(Noise_{\text{Read}} / \mathit{gain})^2 + (\text{DN} / \mathit{gain}) + (Noise_{\text{Scale}} * 0.01 * \text{DN})^2]$$

where:

- DN is the value of the stack pixel value, in counts.
- $Noise_{\text{Read}}$ is the constant read noise in electrons, read from the primary header keyword `READNSE`
- gain is the calibrated conversion from electrons to DN, read from the primary header keyword `ATODGAIN`, and
- $Noise_{\text{Scale}}$ is a linear term in the noise model, read from the CRR reference table.

The `sigmas` parameter is read from the CRR calibration reference file. `Sigma` is a string, e.g., `sigmas = "4.5, 3"`. The number of entries in the string dictates the number of iterations to be performed (in this example two) and the values in the string indicate the value of `sigmas` for each iteration. In this example in the first iteration non-masked stack values deviating from the guess image value by more than $4.5*\mathbf{noise}$ are considered outliers and are excluded from the average. A second iteration is then performed in which `sigmas` is set to 3 and non-masked stack values disparate by more than $3*\mathbf{noise}$ from the guess image are excluded when determining the average. If `radius` is one or greater, then pixels neighboring rejected pixels are also excluded when forming the average (see below).

`Scalenoise` is linearly proportional to the signal level. If `scalenoise = "2.0"`, then the term $0.02*\text{DN}$ is added in quadrature to the **noise**. Experience has shown that this multiplicative factor is important to model the effects of flat-field variations on the noise so that, for example, the centers of stars are not flagged preferentially to the sky background.

Rejection Propagation to Neighboring Pixels

Most cosmic ray impacts affect more than one pixel. For this reason, *calstis2* was designed to permit neighboring pixels to be examined as a separate case, with possibly

more stringent rejection thresholds to exclude cosmic rays. This “sympathetic” rejection of neighboring pixels is controlled by the `radius` and `pfactor` parameters. `radius` is a floating-point number with units of pixels, and is measured from the center of the pixel in question. If `radius=1.0` a total of 5 neighboring pixels in a “+” shaped pattern will be re-examined with the rejection threshold scaled by the `pfactor` parameter. If `radius = 1.5`, a total of 9 pixels in a square centered on the original pixel will be examined for rejection. Values in the neighboring pixels are discarded if they deviate from the guess image by more than `pfactor*sigmas*noise`. If `pfactor = 0.`, then all values in the stacks of the affected neighboring pixels are rejected.

5. The Output CR-Rejected Image

The output of `calstis2` is an image containing a single IMSET: the `_crj` file. This file has a primary header, and a triplet of image extensions: one each for the science (SCI), error (ERR) and data quality (DQ) arrays. Header keywords that are updated during processing are listed in Table 1.

Science Extension

The output science image is constructed as the sum of all the non-masked and non-rejected values in the stack, where the value at each output image pixel is weighted by the exposure times of the constituent images. For example, if `CRSPLIT=3` and `TEXPTIME = 180`, and each extension had `EXPTIME=60`, then if for a given output pixel, one of the input pixels were flagged as bad, the summed value would be scaled by 3/2. The `EXPTIME` extension header keyword is set to the value of `TEXPTIME` in the `_crj` file. A block of header keywords (Cosmic Ray Rejection Parameters) is updated in the output file, indicating the parameter values used to control the cosmic ray rejection process.

Error Extension

Errors for the output file are calculated as the errors from *all stack pixels which contribute to the SCI extension*, added in quadrature, and weighted by the corresponding `CRSPLIT` or repeated image exposure times.

Data Quality Extension

The output data quality file is formed as the bitwise OR of all stack data quality flags for stack pixels which contributed to the SCI extension. Note that usually no stack pixel flagged as a cosmic ray will be in the output SCI extension, since usually there are no deviant pixels in the input images that survive the rejection process.

DQ Flagging in the Intermediate Images

If CRMASK is set to yes, *calstis2* will also update the data quality values in the intermediate output (`_flt`) file IMSETs to flag pixels determined to be cosmic-ray impacted. If CRMASK=yes the CR-impacted bit will be *set* either on or off in the input file. That is, if the pixels was not flagged as cosmic ray impacted then the corresponding flag ($2^{13} = 8192$) will be set *off*, if it was flagged as cosmic ray impacted the flag will be set *on*. This resetting of the bit is important: it allows users to run the off-line version of *calstis2* multiple times on the same input file without the need to re-set the bits explicitly each time.

The parameter CRMASK is set to “yes” in the pipeline. Thus the data quality extensions in the `_flt` file will be updated to indicate pixels flagged as being cosmic ray impacted.

6. Error Trapping

All error conditions in *calstis2* should be trapped and an informative message is output to the user. This message includes the module that detected the error condition, as well as a short message explaining the condition. *Calstis2* has been fairly extensively tested (Shaw, Katsanis, & Potter 1997), so that most detectable error conditions are likely to result from invalid parameter settings, corrupted header keywords, missing or corrupted images, and the like.

In general, all errors in *calstis2* are fatal, and should cause processing to stop. This prevents building in complex logic into the application to, in effect, guess how the user would want processing to continue. Such logic is notoriously difficult to validate, and would have the effect of insulating the user from error conditions that should be investigated. There are, however, some error conditions under which *calstis2* processing will continue. If the parameter `initgues` cannot be found or is invalid, then *calstis2* will proceed using the image minimum as the initial guess image. If the `crthresh` parameter cannot be found, then processing will continue only if the `crradius` parameter is valid and has the value “0.0”. Finally, if the parameter `crmask` cannot be found, *calstis2* will proceed by assuming this parameter has been set to “T” and issue a warning to that effect.

7. CR Rejection in the Off-line Environment

The *calstis2* module for cosmic-ray rejection can be applied to STIS data in one of three ways: during the course of **calstis** pipeline processing, or separately as a host-level application (**cs2.e**), or as a task in STSDAS (**stis.ocreject**). This allows users a good deal of flexibility to re-perform the cosmic ray rejection with control parameters optimized for their particular science goals. When run as part of the **calstis** pipeline, the control parameters come exclusively from the CRR reference file. When run at the host level as **cs2.e**, or within IRAF as the task **ocreject**, users can override the values in the CRR calibration reference file with command-line switches or task parameters, respectively. The corre-

spondence between the command-line switches for **cs2.e**, the **ocrreject** task parameters, and the CRR reference file column names, is given in Table 3 below. Details of the user interface for the stand-alone tasks is given in Katsanis, et al. (1998). In all these cases, the header keywords (e.g., READNSE , ATODGAIN) are updated in the image header to reflect the input parameters actually used.

Table 3: Control Parameters for cs2.e and ocrreject

Switch	Corresponding Task Parameter ^a or Header Keyword	Description
-table	CRREJTAB	CR rejection reference table to be used
-scale	SCALENSE	Multiplicative scale factor, in percent, applied to expected noise
-init	INITGUES	Method for forming the initial image
-sky	SKYSUB	Method of sky subtraction
-sigmas	CRSIGMAS	Rejection threshold(s)
-radius	CRRADIUS	Distance over which sympathetic rejection applies
-thresh	CRTHRESH	Relative threshold for sympathetic rejection
-pdq	BADINPDQ	Integer value corresponding to DQ bits to mask for rejection
-crmask	CRMASK	Flag DQ-rejected pixels in <code>_flt</code> IMSETS?

a. Task parameters for **ocrreject** have the same spelling, but are lower-case.

Finally, the off-line versions of *calstis2* have the ability to combine all IMSETs in multiple files, using standard IRAF image wildcard and template schemes. This facilitates combining large numbers of images with the same pointing (i.e., more than the eight allowed for CRSPPLITs), or for building bias, dark, and flat-field calibration reference files. In these cases the algorithm is exactly the same, except that the initial image stack is expanded to accommodate all the IMSETs, and the default control parameters are obtained somewhat differently. Since the CRR table will only contain entries for valid values of CRSPPLIT, and the maximum value (8) may be smaller than the total number of IMSETs to combine, the default values are taken from the row with the maximum value of CRSPPLIT, and the closest match of mean exposure time. The default control parameters can (and probably should) be over-ridden.

8. References

- Hodge, P., & Baum, S. 1995, STIS Instrument Science Report 95-07
Hodge, P. et al. 1998, STIS Instrument Science Report 98-10
Katsanis, R., et al. 1998, STIS Instrument Science Report, 98-14

MacKenty, J. W., and Baum, S. 1996, STIS Instrument Science Report 96-16

Shaw, R. , Katsanis, R., and Potter, M. 1997, STIS Technical Instrument Report 97-04