Appendix A. STEREO/SECCHI Level-0.5 to Level-1 Calibration CMAD

Version 1.0 / 30 June 2021 – Original release

Version 1.1 / 06 December 2021 – Updated references

A.1 Algorithm Description

The software approach for calibration of the SECCHI image data from Level-0.5 (described in [**PIPELINE**]) to Level-1 products follows the well-established 'prep' SolarSoft framework. Namely, a level-0.5 FITS image is passed to an instrument-specific *_prep.pro routine (i.e., trace_prep, eit_prep, etc.). The routine applies the latest calibration and correction routines to produce a level-1 FITS image suitable for scientific analysis.

Since SECCHI is an instrument suite, SECCHI_PREP is the front-end calibration procedure for all SECCHI instruments. It calibrates the raw FITS images (level 0.5) into level 1.0 and level 2.0 files suitable for quantitative photometric analysis.

A.1.1 Purpose

SECCHI_PREP applies a set of calibrations common to all four telescopes (EUVI, COR1/2, HI-1/2) while it parses instrument-specific calibration to the corresponding *_PREP routines. The latter are described in the instrument-specific CMADs. Here, we describe the SECCHI_PREP-specific operations, which we split into three categories, as follows

Section A.2	Common Calibration Operations
Section A.3	Image Utility Operations
Section A.4	Flight Software Operations

A.1.2 Contents

Section A.2 describes the calibration procedures common to all four SECCHI instruments; namely, bias subtraction, exposure time normalization, and calibration factor and calibration image retrieval. Section A.3 describes image utility operations such as replacement of missing pixels, retrieval of masked pixels, calculation of image statistics, removal of non-imaging pixels, placing images in a standard grid, and updating header information. Finally, A.4 describes corrections for flight software image processing operations.

A.1.3 References

SECCHI Science Operations Manual

A.2 Common Calibration Operations

A.2.1 Product Description

For calibration of the raw Level-0.5 FITS files to photometric units, all SECCHI images must first be transformed to DN/s and then have the calibration factor and an optional calibration image (e.g. vignetting) applied. These operations are performed via the routines GET_BIASMEAN, GET_CALFAC, GET_CALIMG, and GET_EXPTIME.

A.2.1.1 Heritage

This is a standard procedure for the calibration of solar images.

A.2.2 Theoretical Description

GET_BIASMEAN: Because the Analogue-to Digital Converter (ADC) requires a positive signal to operate correctly, flight electronics injects an electronic offset in the readout of each row. This is the bias level and is calculated by finding the mean of a column in the underscan area of the CCD. This is column 25 for a full sized 2176x2048 image. Smaller binned images will use a different column to calculate the bias. SECCHI will always take images including the underscan area due to the need to calculate the bias. Every image taking has a mean and standard deviation of the CCD bias calculated with results put into the image header. The flight software (SW) provides the option to subtract the bias onboard although this option is rarely used. For summed images (notably HI) and square root images (used for testing purposed only) the bias is subtracted from the image onboard. This is important because the CCD bias has been found to vary from image to image and from camera to camera. The GET_BIASMEAN program corrects the mean bias for SECCHI Electronics Box (SEB) Imaging Processing (IP) bias subtraction and SEB IP image summing.

GET_EXPTIME: The exposure time for a given image is recorded in the downlinked image header. The program reads the value from the header.

GET_CALFAC: The program retrieves the latest calibration factors (DN/s to MSB (COR1/2 & HI-1/2) or Photons/sec (EUVI) adjusting them for image summing, If applicable. The derivation of these factors is explained in the specific instrument's CMAD.

GET_CALIMG: The recorded intensity may suffer from vignetting effects due to the presence of occulters or stops in the optical path. To correct for these effects an appropriate, so-called 'flatfield' image. The program retrieves the appropriate calibration image for flatfield/vignetting correction.

A.2.3 Error Analysis and Corrections

The bias level was determined during ground testing and during the early commissioning phase. Because of the very stable CCD and flight electronics operations since launch, there has been no need to adjust it.

The exposure times were calibrated during ground testing. No corrections are necessary.

The calibration factors and flatfield images were determined during ground testing. These have been updated after flight calibrations. The calibration images are stored in the secchi calibration directory in solarsoft.

A.2.4 Calibration and Validation

The exposure times and calibration factors have been validated via in-flight calibrations which are reported in the individual instrument CMADs.

A.2.5 References

N/A

A.3 Image Utility Operations

A.3.1 Overview

To facilitate scientific analysis, SECCHI images may require replacement of missing pixels, trimming to a common size and other operations that are unrelated to calibration. These operations are described in this section.

A.3.1.1 Heritage

Many of these corrections are specific to the SECCHI instruments.

A.3.1.2 Product Description

GET_SMASK: Pixels outside the imaging area of the instrument contain noise and are usually masked to prevent errors during analysis. The routine returns an image containing a smooth mask for each instrument. The mask image is a binary array with values 1 or 0. The function crops and scales mask image to match the data image.

SCC_GET_MISSING: data packets may occasionally be lost during downlink resulting missing pixels or even blocks of pixels in images that may create problems with analysis algorithms. The SCC_GET_MISSING function returns the subscript of pixels that have either missing or corrupted data due to transmission loses. The function uses the header values in MISSLIST to find the missing or corrupted image pixels.

SSC_IMG_STATS: Past experience has shown that statistics of image intensities (e.g. mean, quartiles, max/min values) can be very useful for trending or image quality assessments or large-scale data processing. The SCC_IMG_STATS procedure calculates the image statistics of the input image and returns the values in the output variables. The keyword MISSING allows the specification of which pixels are missing data in the image and thus are not to be included in the image statistics.

SCC_IMG_TRIM: Some SECCHI images include an over and under scan region in the image. These regions are used to monitor the mean bias and other calibrations factors.. The SCC_IMG_TRIM function removes the over/under scan region from the data and updates the effected header keyword values. By default, SECCHI_PREP trims off the over/under scan regions of the image array. The images will not be trimmed if the keyword TRIM_OFF is set in SECCHI_PREP

SCC_PUTIN_ARRAY: not all SECCHI image are acquired with the same pixel size. Some are subfields of the 2048x2048 array, while others are binned to save telemetry. Many analysis algorithms require images of uniform size. The SCC_PUTIN_ARRARY function resizes images to the desired output array. The function will position subfield images in a blank full field, rebin the images to the size given in the SECCHI_PREP keyword OUTSIZE and update the effected header values.

SCC_UPDATE_HDR: The SCC_UPDATE_HDR function updates any header keyword values that change as a result of the calibration. It calls the SCC_IMG_STATS procedure to calculate the image statistics.

SCC_UPDATE_COMMENTS: The SCC_UPDATE_COMMENTS function updates any header keyword comments that change as a result of the calibration. Only used to write FITS files.

A.3.2 Theoretical Description

N/A

A.3.3 Error Analysis and Corrections

N/A

A.3.4 Calibration and Validation

N/A

A.3.5 References

[PIPELINE] STEREO/SECCHI Level-0 to Level-0.5 FITS Pipeline CMAD, STEREO_SECCHI_Prep_CMAD_20211206.pdf, Version 1.1, 06 December 2021

A.4 Flight Software Operations

A.4.1 Overview

Once an image is read out, it must undergo further processing. The image processing software is akin to a pocket calculator with each function operating on an image instead of a number. Also, similarly to calculator memory registers, the onboard software uses image buffers for certain operations. For the level-0.5 to 1 calibration discussed here, we are concerned only for operations on pixel intensities.

A.4.1.1 Heritage

The onboard image processing software is based on software and algorithms developed for SOHO/LASCO and EIT and TRACE image acquisition. The specific operations discussed here were developed during ground testing and are dictated by the needs of the image compression algorithms.

A.4.1.2 Product Description

Table 1 Snippet from a COR1 Level-0.5 FITS header containing the onboard image processing operations for that image.

IP_PROG0=	41 / Use SSR1 APID
IP_PROG1=	76 / IP Trim
IP_PROG2=	3 / Pixel Summing
IP_PROG3=	50 / Divide by 4
IP_PROG4=	3 / Pixel Summing
IP_PROG5=	50 / Divide by 4
IP_PROG6=	1 / Divide by 2
IP_PROG7=	97 / ICER7
IP_PROG8=	0 / No Operation
IP_PROG9=	0 / No Operation
IP_00_19=	' 41 76 3 50 3 50 1 97 0 0 0 0 0 0 0 0 0 0 0 0' /

SECCHI_PREP calls the SCC_SEBIP function to correct the Level-0.5 images for any onboard (SEB) image processing (IP) operations involving pixel intensities. For example, the SEB IP can be set to take the square root of the image or divide the image by a factor of 2, 3, or 4, multiple times. The numerical codes for the applied IP functions are kept in the Level-0.5 header (IP_00_19; see example from COR1 FITS header in *Table 1*) and are explained in the SECCHI Operations Manual. The ones relevant to this section are shown in Table 2 for easier reference.

Table 2 SECCHI Flight Software Image Processing Functions discussed in this document. The full List is available in the SECCHI Operations Manual

Number	Hex	Name	Description
0	0x00	No Operation	Signals End of Image Processing
1	0x01	Divide by 2	Divide each pixel value by 2. May be applied multiple times to an image.
2	0x02	Square Root	Square Root of each pixel
16	0x10	H-compress - scale factor[8]	Send to Ground with H – Compress (see Note 2 below) May 2009 changed to HI1SPW 4 x 4 bin HI1 summing buffer, clip to 2^20, divide by 16 Sends sunward 256 rows and 128 cols
17	0x11	H-compress - scale factor[9]	Send to Ground with H – Compress (see Note 2 below) May 2009 changed to HI2SPW 4 x 4 bin HI2 summing buffer, clip to 2^20, divide by 16 Sends sunward 256 rows and 128 cols
50	0x32	Divide by 4	Divide by 4
53	0x35	Pixel Sum + Divide by 4	Pixel Sum followed by Divide by 4 (result of IP 52 – not callable)
82	0x52	IP_32DIV2	Used for reporting Divisor used by IP C0mmand 0x48 and 0x48. Should never uploaded by planning tool
83	0x53	IP_32DIV4	Used for reporting Divisor used by IP C0mmand 0x48 and 0x48. Should never uploaded by planning tool
84	0x54	IP_32DIV8	Used for reporting Divisor used by IP C0mmand 0x48 and 0x48. Should never uploaded by planning tool
85	0x55	IP_32DIV16	Used for reporting Divisor used by IP C0mmand 0x48 and 0x48. Should never uploaded by planning tool
86	0x56	IP_32DIV32	Used for reporting Divisor used by IP C0mmand 0x48 and 0x48. Should never uploaded by planning tool
87	0x57	IP_32DIV64	Used for reporting Divisor used by IP C0mmand 0x48 and 0x48. Should never uploaded by planning tool
88	0x58	IP_32DIV128	Used for reporting Divisor used by IP C0mmand 0x48 and 0x48. Should never uploaded by planning tool
118	0x76	Divide by 3	Divide image by 3. Useful for total brightness images by COR1 and COR2

A.4.2 Theoretical Description

Onboard image arithmetic operations such as square root or divide by 2 or other factors are dictated by the needs of the compression algorithms. SECCHI uses the ICER algorithm (Kiely & Klimesh 2003) which operates with images up to 13-bit. However, the SECCHI uses a 14-bit ADC. Therefore, the images are divided by 2 before compression to bring them within the ICER range.
The signal level on summed images (e.g. all COR2 and HI images) can reach 16-bit so the images are divided by 4 for digitization. Other operations are explain briefly below.

The level-0.5 headers report the IP processing type, via a number (0-255). If the particular IP step has been applied multiple times, then the IP number is repeated on the header (e.g. *Table 1*). The SCC_SEBIP function parses the input image headers, counts the IP types and number of times, ***n***, each IP step is applied and then corrects the images, as follows:

- Only IP steps 1, 2, 16-17, 50, 53, 82-88, and 118 are considered. Steps 82-88 have not been implemented onboard but are in the code for future compatibility. These steps apply divisions by 2^{0-7}.

- SEB_IP 1: division by 2. This is the most common operation for ICER-compressed images. Then, corrected_image = 2^*n**input_image
- SEB_IP 2: square_root. Only used for testing. Then, corrected_image = input_image$^{2^n}$
- SEB_IP 16 or 17: This processing applies to the images destined for the space weather beacon. These are highly compressed using H-compress. Then, corrected_image = 64^n*input_image
- SEB_IP 50: division by 4. Used for summed images. Then, corrected_image = 4^*n**input_image
- SEB_IP 53: pixel summing then division by 4. Used for CCD summing. The pixel summing is accounted for in the calibration factor correct (get_calfac). The operation is applied only once. So, corrected_image = 4*input_image
- SEB_IP 118: division by 3. Used for COR1 and COR2 total brightness images. The operation is applied only once. So, corrected_image = 3*input_image

A.4.3 Error Analysis and Corrections

The pixel intensities of the SECCHI raw images are unsigned integer type. Therefore, a division of the pixel intensities by any factor introduces a roundoff error. For example, when an image is divided by 2, odd intensities become even. When that image is multiplied by 2 during SECCHI_PREP, the resulting pixel will be 1 DN fainter. This effect is most obvious at low signal levels at or near the noise level. The division operation removes subtle gradients in the intensities thus artificially smoothing the noise. Since ICER is a lossy compression scheme, such effects are expected and are acceptable for scientific analysis since the only affect intensities at the noise level.

A.4.4 Calibration and Validation

N/A

A.4.5 References

The descriptions of the IP functions by number are in Science Operations Manual (MSWord).

Kiely, A. & Klimesh, M. The ICER progressive wavelet image compressor. IPN Progress Report 42, 1–46 (2003).