CISSCAL User Guide

March 20, 2009

Contents

1 Introduction

CISSCAL is the Cassini Imaging Science Subsystem Calibration software. It is essentially a graphical interface for performing the steps outlined in the theoretical_basis.ps document written by Bob West and included with this package. It performs standard CCD calibration steps such as bias and dark subtraction and flatfield correction, as well as ISS-specific calibrations such as bitweight correction and removal of 2-Hz noise. CISSCAL only reads and writes images in VICAR format.

CISSCAL was developed in stages, beginning in August 1998, under the auspices of the Cassini Imaging Central Laboratory for Operations (CICLOPS) directed by Dr. Carolyn Porco, ISS Team Leader. The theoretical basis and pipeline design for Cassini image calibration was developed by ISS team member Dr. Robert West of JPL. The systems design for the CISSCAL software, and its original implementation and development, was the work of Dr. Kevin Beurle at Queen Mary, University of London from August 1998 through March 2002. Beginning in April 2002, major design and algorithmic modifications, software additions, debugging and maintenance have been performed by Benjamin Knowles of CICLOPS/Space Science Institute, Boulder, CO. under the direct supervision of West. Significant contributions (algorithms, software, evaluation and validation, etc) have been made by West, Vance Haemmerle of JPL, Daren Wilson of CICLOPS, and other members of the Cassini ISS Team.

All questions regarding CISSCAL should be directed to Ben Knowles at the Space Science Institute, Boulder, CO (knowles@ciclops.org).

2 Setting Up the Environment

CISSCAL is written in the Interactive Data Language (IDL), and thus requires that IDL (version 5.5 or later) be installed on the computer on which CISSCAL is to be run, and its executable placed in the user's PATH.

Once IDL has been installed, the user needs only to edit his or her login shell script (.cshrc, .tcshrc or similar) to define the CISSCAL-specific environment variables CisscalDir, CalibrationBaseDir, and ImageBaseDir. Assuming that the user is running the csh or tcsh shell and has installed CISSCAL in his or her home directory in a subdirectory named "cisscal3_6," and that the calibration support directory CALIB has been downloaded, the following lines should be added to the .cshrc/.tcshrc file:

```
setenv CisscalDir ~/cisscal3_6
setenv CalibrationBaseDir ~/CALIB
setenv ImageBaseDir ~/images
```

Specifically, CisscalDir specifies the location of the CISSCAL software files, CalibrationBaseDir specifies the location of the calibration support directories (version 3 of this software requires: antibloom/, bitweight/, correction/, darkcurrent/, distortion/, dustring/, efficiency/, lut/, offset/, slope/), and ImageBaseDir is the default directory where CISSCAL will look for images to be calibrated.

It is also a good idea to add the CISSCAL directory to your IDL_PATH system variable if you wish to call CISSCAL from a directory other than that specified in CisscalDir. If the csh or tcsh shells are being used, this may be accomplished by adding the following line to your shell script:

```
setenv IDL_PATH {$IDL_PATH}:~/cisscal3_6
```

CISSCAL has been tested under IDL version 5.5 and newer, in Linux and Unix environments. It may not be compatible with other operating systems such as Windows.

3 Starting CISSCAL and General Layout

CISSCAL is launched by typing "@cisscal" at the IDL prompt, or "idl cisscal" from the terminal prompt. Doing so will launch the main graphics widget.

The CISSCAL menu bar contains several pull-down menus. The menu buttons are labeled "File," "Image," "Tools," "Log Options," and "Batch Mode."

Below the menu bar is a text window which logs to the GUI any messages generated by CISSCAL. Various logging options can be adjusted by the user from the "Log Options" menu item, as discussed below.

To the right of the log window is a list of calibration options to be set by the user. These options are executed in the order listed, and can be toggled on and off using the buttons in the "On/Off" column. The calibration options are:

- LUT conversion
- Bitweight correction
- Subtract bias
- Remove 2-Hz noise
- Subtract dark
- A-B pixel pairs
- Linearize
- Flatfield
- Convert DN to flux
- Correction factors
- Geometric correction

With the exception of the geometric correction, the default value for each of these options is "ON."

To the immediate left of the calibration option names is another column of buttons which toggles the "Option Parameters" field for that option. These parameters appear in the lower right of the GUI. Some calibration options will not have any user-definable parameters, while others have many, and can be controlled quite specifically. See the section on each calibration option for details.

When an image is read into CISSCAL, another field pops up at the bottom of the GUI which gives the keyword values pulled out of the VICAR label. This table is not editable, and is for information purposes only.

The general order of operations in CISSCAL is as follows: 1) read in an image file, 2) select the desired calibration options, 3) select the desired parameters for each calibration option, 4) go to "Calibrate Image" under the image menu to execute desired calibration steps in the order listed, 5) save output to a real-format VICAR image file.

3.1 Default Options File

The default calibration options used by CISSCAL are specified in the cisscal_default_options.txt file included with the CISSCAL distribution. This file is user-editable, though care should be taken not to corrupt the formatting.

The default options file consists of a list of 23 keywords and their values, separated by a colon. Allowed values are given in the file itself, and should be relatively straightforward. Only the first 22 keywords are calibration options per se; the 23rd is the default output filename suffix, which will replace the input filename suffix when the output VICAR file is written. This is initially set to '.IMG.cal', but alternatives such as '_cal.IMG' may also be used. Note that the period is included as part of the suffix.

4 Pull Down Menus

The entries in the pull-down menus across the top of the main CISSCAL widget are as follows:

4.1 File Menu

The 'File' menu contains the following menu buttons:

4.1.1 Open

This menu item opens a VICAR image file and stores its image data, binary header, and label information in memory as an image object. Two ancilliary arrays are also generated at this time: a missing pixel array, and, if the VICAR image is in integer format (that is, presumed to be raw and uncalibrated) then a saturated pixel array as well. Once an image has been read into memory, the user can operate on it with the selected calibration options by choosing "Calibrate Image" from the Image menu.

While only one image may be read into memory at a time, CISSCAL does have a limited batch mode. In this mode the "Open" button is not used at all, and images are read in sequentially and individually.

4.1.2 Save Image

Selecting the "Save Image" item prompts the user to name an output file to which the currently defined image will be saved in VICAR format. This output VICAR file is not identical to the input file. The primary differences are:

- Output image is in real (floating-point) format instead of integer
- Output image lacks binary header (including overclocked and extended pixels)
- VICAR label has PROCESSING_HISTORY_TEXT keyword appended, with record of calibration tasks performed
- VICAR label has CALIBRATION_STAGE keyword appended, which allows CISSCAL to automatically resume calibration of a partially-calibrated image at the same stage that it left off

On output, CISSCAL automatically sets the missing pixels (defined either when the image was first read in, or by the user with the "Read missing pixel file" menu item) back to a value of 0. In the future this may be changed to NaN to avoid confusion with "real" pixel values of 0.

4.1.3 Save Calibration Options File

This menu item allows the user to save the current calibration options, including On/Off status as well as individual option parameters, to a binary-format file that can easily be read in at a later time.

4.1.4 Load Calibration Options File

This menu item loads a previously-saved calibration options file.

4.1.5 Write saturated pixel file

The user will be prompted to specify an output filename (default: <input image name>.sat) for writing the saturated pixel array to an output image file. This array has 1s for pixels with DN values of 255 (8-bit image) or 4096 (12-bit image), and 0s otherwise. The output file will be a real-format VICAR image with the same label as the normal output image file.

4.1.6 Write dark sky mask file

The user will be prompted to specify an output filename (default: <input image name>.mask) for writing the masked pixels array to an output image file. This array is generated by the 2-Hz noise removal algorithm (image mean method), which uses it as a way to exclude non-dark sky pixels from the noise estimate. Pixel values of 1 are considered masked and 0s are unmasked. The output file will be a real-format VICAR image with the same label as the normal output image file.

Note that the dark sky mask created by the 2-Hz algorithm is based on a simple theshold method which may not yield adequate results in all cases, particularly if there are detailed surface or ring features in the image. In such cases an external mask file may be created by the user and read in from the 2-Hz noise parameter window.

4.1.7 Write missing pixel file

The user will be prompted to specify an output filename (default: <input image name>.miss) for writing the missing pixel array to an output image file. "Missing pixel" is here defined as one where DN = 0 in a contiguous block, and the array has 1s for missing pixels, and 0s otherwise. This array is generated whenever an image file is first read into CISSCAL, and is used by the 2-Hz noise removal algorithm as a way to ignore these pixels when constructing the 2-Hz signal to be removed. The missing pixel file will be a real-format VICAR image with the same label and dimensions as the output image file.

4.1.8 Read missing pixel file

If the user is processing a previously-calibrated image file, it may be necessary to read in a missing pixel file created during the previous CISSCAL session. The array read in using this menu option will replace the missing pixel array that was automatically generated upon first reading in the image file.

4.1.9 Quit

Self-explanatory.

4.2 Image Menu

This menu contains the all-important "Calibrate Image" button, as well as utilities for viewing and examining the image array itself.

4.2.1 Calibrate Image

Selecting this menu item causes CISSCAL to execute all the currently-selected calibration options under the calibration option listing, with their associated option parameters.

Note that in batch mode (see below), this button will not be used at all.

4.2.2 View Image and View Sliding Image

These are simple image display utilities that allow one to display the image array with a variety of different scalings: linear, logarithmic, square root, histogram-equalized, and range-stretched.

The "View Sliding Image" option displays the image array in a smaller, two-sided graphics widget, with the complete image on the left, shrunk to fit the window, and a full-resolution version of the image in the right window with vertical and horizontal scrollbars.

4.3 Tools Menu

This menu contains various utilities for examining the image pixel values.

4.3.1 Histogram and Histogram (log)

Displays a histogram of pixel values. This plot can be output to a postscript file if desired (file name = <input file name>_hist.ps).

4.3.2 Profiles and Average Profiles

Plots horizontal and vertical profiles (ie. "slices") of image pixel values across a given line or down a sample column. The line and sample can be specified with the cursor, othewise it defaults to the center pixel line and sample. "Average Profiles" simply plots the average of all horizontal and vertical profiles. This plot can be output to a postscript file if desired (file name = <input file name>_pro.ps or _ave.ps).

4.3.3 Inspect Pixel Values

Displays image and allows user to view pixel values and line/sample by moving the cursor over regions of interest.

4.4 Log Options Menu

This menu allows the user to customize various message logging options.

4.4.1 Log to...

The user may choose to log all messages to one of four locations: the GUI log window (GUI), the IDL terminal window (stdout), the IDL error system variable (stderr), or a user-specified log file (file). If the latter, the specified file is opened in append mode, written to, and closed every time a new message is written, so that it is continually updated during the CISSCAL session.

4.4.2 Set Log Level...

There are three "log levels" from which the user may choose: 0 (no messages logged), 1 (standard message logging - the default setting), 2 (all messages logged).

4.5 Batch Mode

In batch mode, the user selects the calibration options and option parameters he or she desires, as usual, but then clicks on the "Batch Mode" menu item instead of "Calibrate Image." Doing so will bring up a separate dialog from which the user sets the following batch options:

- Input Directory: directory containing images to be calibrated
- Input Filter/File List: see description below
- Output Directory: where calibrated images will be written
- Output Filename Extension: file suffix to replace that of the input image filename
- Dark Subtraction Options: allows user to specify a list of dark files corresponding to each image (only used in lieu of the interpolation dark model)

If the "Input Filter" toggle is selected, the user can enter a regular expression to designate which files in the directory are to be calibrated. The default is *.IMG, or all image files. Another example would be N* or W*, i.e. all NACs or WACs.

Alternatively, the user can specify use of a file list. This is just an ASCII text file containing a single-columned list of the names of images to be processed. If this file is located in the Input Directory, then full image path names are not necessary.

A batch file list can be easily constructed using a command such as:

```
> ls -1 *.img > batch.txt
```

There are several limitations that come with working in batch mode. For one, there is currently no way to save ancilliary image data such as the missing/masked/saturated pixels array. But more generally, some images may just require more individual attention than others, and a particular set of calibration options suitable for one image may not be suitable for all.

5 Calibration Options

This section will discuss the individual calibration options and their user-definable parameters. Not all of the calibration steps apply to both cameras and all camera modes; in some cases this is by design, and in some cases it's simply due to incompleteness in the calibration analysis. In situations where a given option or parameter does not apply to the currently defined image, the step will simply be skipped (and a relevant

message logged). CISSCAL is more-or-less self-contained in the sense that the image itself tells it whether a certain calibration step should be executed. This also means that, when in doubt, it usually doesn't hurt to leave a given option "ON," and let the software decide whether or not it needs to be performed.

The steps discussed below are always performed in the same order that they are listed here, and in the main CISSCAL GUI window. This is by design, as it is generally necessary to perform subtractive corrections (like bias and dark removal) before multiplicative corrections (like flatfielding, converting to flux units, etc). Also note that some calibration options, most notably flatfielding and converting from DN to flux values, are actually comprised of multiple calibration steps. This will be discussed in more detail below.

5.1 LUT conversion

This option applies only to images with a DATA_CONVERSION_TYPE of "TABLE." For these images, a reverse-lookup table will be applied to convert the pixel values from an 8-bit range (0-255 DNs) to a 12-bit range (0-4096 DNs). The lookup table in question is hardcoded into CISSCAL, although the table is also available in the lut.tab file included in the lut/ subdirectory of the calibration support directory.

This algorithm has no user-definable parameters.

5.2 Bitweight correction

This algorithm corrects for uneven bit-weighting, as described in the ground calibration report. It is not applied to LOSSY-compressed or TABLE-encoded images, due to loss of information in the compression/encoding processes. The bitweight corrections are a function of camera, gain state and temperature, and the files used to correct for it can be found in CalibrationBaseDir/bitweight/.

This algorithm has no user-definable parameters.

5.3 Subtract bias

The "bias" is the zero-exposure DN level of the CCD chip, and bias subtraction is a standard CCD image calibration procedure.

There are two ways in which CISSCAL can remove a bias level from an image: either by simply subtracting the BIAS_STRIP_MEAN value found in the image header, or by using the overclocked pixel array taken out of the binary line prefix.

The overclocked pixel method is generally preferred, as it removes a line-dependent bias level as opposed to a single value. Using this method, the bias is derived by simply performing a linear fit to the overclocked pixel array. Note, however, that the overclocked pixels will work better for some images than others, and in some cases the overclocks may be corrupted in some fashion or otherwise unusable. To check this, the user may find it useful to view a plot of the overclocked pixels by clicking on the "Plot overclocks..." button.

In practice, the results of the two methods will usually be almost identical, since the linear fit to the overclocked pixels tends to be very flat.

5.4 Remove 2-Hz noise

Cassini ISS images suffer from a particularly bothersome type of coherent noise that results in a horizontal banding pattern across the image. This noise is introduced during image readout, and has been found to have two peaks in its power spectrum near 2 Hz.

No model has yet been derived that reproduces the 2-Hz signal with exact accuracy, so removal of this noise must be accomplished by using the image itself to determine the noise component. This can be done in two ways: by looking at the overclocked pixels (which give a good representation of line-dependent noise sources such as this one), or, if possible, by looking at dark sky areas in the image itself, and constructing a 2-Hz signal by use of an image mean.

Using the first method, the overclocked pixel array is first smoothed, then filtered to remove high-frequency random noise components, then filtered again to remove slow-varying and DC-offset components (which is essentially the same as the bias which has just been subtracted).

The image mean method is somewhat more complicated, and requires that the image not only feature large regions of dark sky pixels, but that these regions are present for all image lines. If this is not the case, such as in images where planetary features completely or mostly fill the field of view, this method will not produce good results.

The image mean algorithm requires the use of a dark-sky mask, which will allow the software to ignore all areas of the image that are not dark sky. This mask can either be created externally as a VICAR image by the user and read into CISSCAL using the "Choose mask file" option, or it can be created automatically by using the "Auto mask" feature. If "Auto mask" is selected, CISSCAL will create a mask based on the threshold and pixel range variables supplied by the user. Threshold is simply the DN value below which the image is assumed to be dark sky - if no threshold is supplied (ie. the parameter field is left as 0.0) then a threshold will be automatically calculated. Note, however, that this automatic threshold calculation assumes the image is a star field, so for images with large non-stellar bright regions, the calculated threshold may not reflect the ideal cut-off.

The pixel range parameter simply specifies how much smoothing is performed on the image before creating the mask. A default value of 9.0 is good for most cases. Higher values will create a more conservative mask (that is, more of the image will be masked, and less will be considered dark sky).

Once a mask file has been read in or the "Auto Mask" parameters chosen, the user can display the mask by clicking the "Show Mask" button. In most cases, judicious use of the image display and "Show Mask" features should allow the user to establish the appropriate values for threshold and pixel range.

When calibration is performed, the 2-Hz removal algorithm constructs an approximation of the 2-Hz signal as a function of line by replacing the masked and missing pixel regions by an average of the unmasked pixels in the same line, applying a smoothing filter, and then taking a median over the sample direction. This signal is then filtered the same way as in the overclocked pixel method, with a low-pass filter to remove high-frequency random noise, and a high-pass filter to remove the DC-offset/bias component, and the result is subtracted out.

Note that the 2-Hz removal algorithm, particularly when using the overclocked pixel method, often does not work well with TABLE-encoded images. This is because the 2-Hz noise is at such low DN levels, and the quantization of the TABLE encoding causes small increments in DN to be lost, thus making an accurate reconstruction of the 2-Hz contribution quite difficult.

5.5 Subtract dark

Dark current in the Cassini ISS cameras is comprised both of the traditional, slowly-increasing kind that we typically associate with dark current, as well as an effect called residual bulk image (RBI). This effect is also time-dependent, but results from the leaking of electrons into the pixel wells from whatever image was on the CCD previous to the current exposure. It is because of this RBI effect that the ISS cameras typically engage a "light-flood" step before every exposure - this ensures that the RBI pattern will at least be uniform across the array.

There are currently two possible methods for removing the dark current. The most straightforward is to simply read in an external VICAR-format dark file (created by the user from actual in-flight dark images, or with whatever method he or she feels is appropriate).

However, because the dark current is a function of how long a given pixel sits on the CCD chip before being read out, and because this in turn is a function of a number of camera parameters including compression mode, compression ratio, summation, read-out index, telemetry rate, etc., it is simply impossible to take in-flight dark images that represent the full range of possibilities. The alternative, therefore is to create a dark image by using a model.

The current best method for removing dark current is the interpolation method, which is now implemented for all camera modes. The interpolation method requires first defining a parameter file where each pixel is represented by 8 coefficients of the form:

$$DN_{dark} = a_0 + a_1 * t_{pix} + (1 - a_2 * e^{-t_{pix}/a_3}) + (1 - a_4 * e^{-t_{pix}/a_5}) + (1 - a_6 * e^{-t_{pix}/a_7})$$

This is similar to the (now defunct) 2-parameter model, but with exponential decay terms to more accurately represent the contribution due to residual bulk image. The dark image is reconstructed by calculating, for each pixel, the amount of time spent on the array by that particular potential well, and then applying the equation above.

The parameter file is not created within CISSCAL, but by an external code written by Bob West at JPL (robert.a.west@jpl.nasa.gov). (Once finalized, this code will be included in future versions of CISSCAL.) NAC and WAC parameter files are included with the software, and can be found in the darkcurrent/ subdirectory of the calibration support directory. Notice that the filename includes a time string, <subscript>. All darks created using a given parameter file will be placed in a subdirectory within darkcurrent/ called darks<subscript>/. For this reason, it is important that the user has proper write permissions to these directories. CISSCAL will always look here for a dark file that matches the parameters of the current image. If it does not find one, it will construct a new dark and write it to a VICAR file in the same directory.

5.6 A-B pixel pairs

This option causes CISSCAL to search for and remove the bright/dark pixel pair artifacts created by the anti-blooming mode. These pairs are only produced when anti-blooming mode is ON, so images with ANTIBLOOMING_STATE_FLAG = OFF will not need to have this step performed.

The creation of anti-blooming pixel pairs is discussed in the ground calibration report. Recent analysis has found that these pairs tend to change intensity and even position with time, so a static pixel pair map (like the one included in the antibloom/ directory, which was created during ground calibration) is insufficient to identify them.

The current algorithm identifies pairs automatically. There is only one user-definable parameter: a threshold level that defines the minimum DN difference between a pixel and its neighbors that will cause it to be identified as a candidate for a pair. The default value for this threshold is 30.0, which should be adequate for most cases. Lowering the threshold can lead to more pixel pairs being identified, and vice-versa, but the effect is much more significant in some images than others.

Once pixel pairs are identified, they are replaced by an average of their horizontal neighbors.

5.7 Linearize

The linearize function corrects for non-linearity of the CCD response, as discussed in the ground calibration report. This linearity correction is a function of camera and gain state.

This algorithm has no user-definable parameters.

5.8 Flatfield

This step is actually comprised of two separate steps: dustring removal and flatfield removal. Technically, there is no difference - any dustring signatures in the image are part of the flatfield - but since we are using flatfields derived from ground calibration data (called "slope files"), and since components of the instrument have likely shifted in flight since then, we need to make additional corrections if we wish to accurately represent the current flatfield of a given filter combination.

The dustring removal code currently applies two corrections to the NAC, and none to the WAC (although more can easily be added when more data becomes available). The NAC corrections are 1) the masking of a large dustring discovered on all NAC images during Venus flyby, and 2) the dividing out of a "mottle map." The mottle map is essentially a residual flatfield (exhibiting a pattern that could be described as "mottled"), and was constructed by averaging together multiple images where Titan covers a significant fraction of the field of view.

The flatfield removal algorithm simply reads in the slope file appropriate for the given filter combination and temperature, normalizes it to the average of its inner 400x400 pixels, and then divides it out of the input image.

The slope files are contained in the slope/ subdirectory of the calibration support directory, and their filter and temperature parameters are identified by the slope_db.tab database files.

The flatfield algorithm has no user-definable parameters.

5.9 Convert DN to flux

This calibration option is actually comprised of several separate calibration steps. In the order of execution, these are:

1. DNtoElectrons - multiplies image by appropriate gain to yield electrons.

2. DivideByExpoT - divides by exposure time, correcting for shutter offset effect. This effect is due to the fact that it takes the shutter a finite amount of time to open and close. The shutter offset correction is a function of camera and temperature, and the relevant data files can be found in the offset/ subdirectory of the calibration support directory.

3. DivideByAreaPixel - divides image by optics area and solid angle. (NAC has area = 264.84 cm^2, solid angle = $3.6x10^{-11}$; WAC has area = 29.32 cm^2, solid angle = $3.6x10^{-9}$.) In I/F mode (discussed below), solid angle is not divided out, as the input flux spectrum - typically solar - is assumed to be integrated over the source.

4. DivideByEfficiency - divides image by the system transmission integrated over wavelength (which is here called the system "efficiency") for the given filter combination. The total system transmission is defined as $QE*T0*T1*T2$, where QE is the quantum efficiency, T0 is the optics transmision, T1 is the transmission of the filter 1, and T2 is the transmission of filter 2. In the current implementation of the code, the quantum efficiency determined from ground calibration is also multiplied by a QE correction spectrum (contained in the files nac_qe_correction.tab and wac_qe_correction.tab in the correction/ subdirectory) that was derived from absolute calibration of star images taken in flight.

The main decision the user must make in converting to flux units is whether to simply convert to standard (cgs) intensity units ($phot/cm^2/s/nm/ster$), or to normalize the intensity in terms of some other flux. Typically this other flux will be the solar flux at the distance of the target from the sun, and the result is called I/F.

5.9.1 Calibrating to I/F

In I/F mode, the user will need to decide whether to normalize the image intensity in terms of the solar flux at a given solar radius, or in terms of some other user-defined flux. If the former, which is the usual approach, you must specify the solar distance by clicking on either "Jupiter" or "Saturn," (which causes the distance to be calculated automatically for the image time given in the image label) or by inputting your own.

If you choose instead to normalize the image intensity in terms of a user-defined flux - essentially replacing the F in I/F with something other than F_{solar} - select "User-input" under I/F options, and click on the "Browse" button to be prompted for a file. The user-input flux file should be ASCII text and contain wavelength-flux pairs where wavelength is in nanometers over the range of the instrument (approximately 200 - 1100 nm), flux is in $photons/cm^2/s/nm$($/ster$ if not integrated over object), and the wavelength and flux values are separated either by spaces or tabs. The top of the file must have a `\begindata` line, above which there can be a header of any length. For an example of this format (with slightly different units), the user should look at the 'solarflux.dat' file supplied with the calibration support directory.

5.10 Correction factors

Because there is always going to be a slight discrepancy between the flux we expect given the integrated system transmission function, and what is actually observed, filter-specific correction factors have been derived to force the observation and theory to match. This calibration option simply divides the image array by these correction values, which are contained in a database file called correctionfactors_qecorr.tab, located in the correction/ directory.

This algorithm has no user-definable parameters.

5.10.1 Derivation of correction factors

The correction factors were calculated starting with component calibrations for the transmission of the optics, filters and quantum efficiency, shutter performance and gain. Of these, uncertainty in the quantum efficiency (approximately 20%) dominates the uncertainty budget. In flight we rely primarily on photometric standard stars with ancillary information from calibrated observations of icy satellites, Saturn, and Saturn's rings.

For the NAC correction factors, we used three different data sets: Vega images taken specifically for ISS calibration between S17 and S41, low-phase Enceladus images (confined to a narrow range in subspacecraft latitude and longitude to minimize systematic brightness differences) and a set of UV-bright stars (the "Bright 19") taken during cruise. All images were processed using CISSCAL, cosmic rays removed, and photometry performed in IDL. For the Enceladus images, both a phase correction and a wide-field PSF correction were applied. Filter-specific correction factors were then calculated by taking the ratio of the observed integrated stellar fluxes to the expected values. These factors have errors on the order of 10-15%, due largely to photometric uncertainty, as well as systematic problems such as noise from the horizontal banding.

For the WAC, we used images of Vega, 77 and 78 Tau, and HR996 from S17 through S41, and tied these star data to the NAC results using BOTSIM images taken using identical filter combinations where available. After performing calibration, cosmic ray removal and photometry as before, the resulting correction factors for all targets were normalized to Vega in the IR, where overlap was seen to be most consistent. We then used a weighted average to calculate final factors. As a final step we applied a linear correction to both the NAC and WAC QE correction functions as a way to "split the difference" between the absolute normalization level of each, while keeping the overall NAC/WAC ratios the same.

The uncertainty of stellar fluxes is on the order of 10%, so this is the uncertainty we expect to achieve when calibration is complete.

5.11 Geometric correction

This option performs a 2-D distortion transformation on the image array, to account for the geometric distortion introduced by the optical system. Fortunately, distortion in the ISS cameras is quite insignificant in most cases: the NAC has almost no distortion at all, and while the WAC has some noticeable pin-cushion distortion, it yields an image translation of only about a pixel at the image corners. Still, correcting for this distortion may be desirable in cases where calculating accurate positions of features is crucial.

This algorithm has no user-definable parameters, and is turned off by default.

6 Running CISSCAL from the IDL Command Line

This software distribution now contains a supplemental IDL routine, cisscal_cl.pro, which allows the user to run CISSCAL from the command line. Usage information is given in the file header. As with the GUI version of CISSCAL, the input and output files are in VICAR format (as opposed to, say, IDL image arrays). This program can be used to calibrate individual images or lists of images, the latter of which may be specified either as a regular expression, an array of filename strings, or alternatively, by using the 'readlist' keyword, as a single-column text file containing a list of filenames. In addition the user may designate a calibration options file to be read, allowing for the use of multiple options files suitable for different targets or image types, for example. Certain calibration options may also be specified explicitly by keyword, overriding the calibration options file settings.

7 Notes concerning Image Object

It may be helpful in getting the most out of CISSCAL to think in terms of its object-oriented data structure. Whenever an image is read into CISSCAL, it is stored as an array within an Image Object structure, of which only one may be defined at a time. Along with the image itself, this structure also contains ancilliary data such as the overclocked pixel array, binary line prefix, image label, missing and saturated pixel arrays, and the method functions that operate on the Image Object.

Each calibration option may be represented by single or multiple method functions. The "Convert DN to flux" option, for instance, is comprised of four separate method functions, whereas "LUT conversion" is a single function. Additional method functions include the routines for reading and writing image arrays and displaying or plotting image values.

The calibration options and their various parameters are stored in a data structure that is entirely separate from the image object. Thus, when you read in a new image it will not have an effect on the calibration options, and vice-versa.