674-SO-43, Version 2.1,
SSC-PD-4066

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: SLOPERROR

12 August 2002

California Institute of Technology
SIRTF Science Center

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

THIS PAGE IS INTENTIONALLY LEFT BLANK

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

SIRTF Science Center

Subsystem Design Specification

Prepared by:

Frank Masci

Concurred by:

Bill Latter Dave Shupe

Approved by:

Mehrdad Moshir

Concurred by:

Bill Green

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

1 Revision History

Version	Description	Date
1.0	Initial version	July 3, 2002
2.0	Updated uncertainty algorithm to account for correlations amongst data samples in ramp. Also modified algorithm to account for DCENUM starting at zero.	July 29, 2002
2.1	Updated algorithm to use a simplified form for the covariance between any two planes in the ramp due to correlated Poission-noise.	August 12, 2002

2 Table of Contents

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

List of Figures

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

List of Tables

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

1. Introduction

1.1. Purpose and Scope

The Subsystem Design Specification is a document that describes the basic requirements, assumptions, definitions, software-design details and necessary interfaces for each subsystem. The document will be used to trace the incremental development of each subsystem and also to allow trace-back of levied requirements; this document should have sufficient detail to allow future modification or maintenance of the software by developers other than the original developers. This document is an evolving document as changes may occur in the course of science instrument hardware design and maturity of operational procedures. This document is not intended to repeat sections or chapters from other Project documents; when appropriate, references to proper sections of primary reference documents will be made.

1.2. Document Organization

This document is organized along the major themes of Requirements; Assumptions; Operational Concept; Functional Descriptions; Functional Dependencies; Input; Output; Other S/S Interfaces; Algorithm Descriptions (when applicable); and Major Liens.

The material contained in this document represent the current understanding of the capabilities of the major SIRTF systems. Areas that require further analysis are noted by TBD (To Be Determined) or TBR (To Be Resolved). TBD indicates missing data that are not yet available. TBR indicates preliminary data that are not firmly established and are subject to change.

1.3. Relationship to Other Documents

The requirements on the operation of SIRTF flow down from the Science Requirements Document (674-SN-100) and the Facility Requirements Document (674-FE-100). The Science Operations System is governed by the SOS Requirements Document (674-SO-100). The current document is also cognizant of the requirements that appear in the Observatory Performance and Interface Control Document (674-SEIT-100) as well as the Flight Ground Interface Control Document (674-FE-101). This document is also affected by the FOS/SOS Interface Control Document (674-FE-102) that governs interfaces between the Flight Operations System and the Science Operations System. Related Software Interface Specifications (SIS) will be as indicated in Section 2.2 of this document.

1.4. Change Procedure

This document is a level 4 document according to the SIRTF Project Documentation Plan (674-FE-103). Changes to this document after approval require the approval of the SOS Change Board (TBD). The process for change control is described in the SOS Configuration Management Plan.

2. Overview

The SLOPERROR program reads (SUR-mode) "slope" image data from a FITS file and estimates associated uncertainties using a simple noise model. The model consists of read-noise, confusion noise and Poisson noise components. The possible covariance between Poisson-noise components between any two samples in the ramp is accounted for. Output noise units are in input slope image units (nominally DN/sec). The software optionally reads in p-mask and d-mask images to handle "fatal" pixels where the noise cannot be robustly computed. This program is specifically designed to compute "a-priori" noise images for raw-data acquired with SIRTF's 24μm array in the "SUR-mode". The noise images will serve as input for error-propagation in the SUR-mode pipelines. SLOPERROR is written in standard ANSI C.

2.1. SLOPERROR Requirements

SLOPERROR is initiated by a startup script under the control of the pipeline executive and does its required functions for a given DCE image or pre-processed DCE image; this involves performing the following tasks.

A.) Retrieve the command line parameters passed by the start up script and use them to run the program.

B.) Read in as input a standard SUR-mode FITS file, mask images and noise parameters.

C.) Produce as primary output a SUR-mode noise image.

D.) Provide exit codes to the pipeline executive and also provides logon and logoff messages identifying the version number and write any error messages to the standard output devices.

E.) Produce a processing summary.

2.2. Applicable Documents

The following documents are relevant to the SLOPERROR program of the AOT PRODUCTS Subsystems.

A.) The SOS Requirements Document

B.) The SOS Downlink Requirements Document

C.) The SOS Downlink Software Development Guidelines

D.) The following Software Interface Specifications (SIS)

SOSDL-SIS-PD-3000 (real*4 DCE data output)

SOSDL-SIS-PD-3001 (d-mask specifications)

SOSDL-SIS-CL-3005 (p-mask specifications)

2.3. Version History

2.3.1. Version 1.0

Initial version created on July 3, 2002.

2.3.2. Version 2.0

Created on July 29, 2002. This version uses a more robust uncertainty algorithm which accounts for non-zero covariances amongst data samples in the ramp.

2.3.3. Version 2.1

Created on August 12, 2002. This version replaces the formula for the covariance between any two samples in the ramp (due to correlated Poission-noise) with a more simple formula provided by John Fowler.

2.4. Liens

No major liens have been identified. However, a possible capability in the long term is to read the gain and read noise parameters on a pixel-by-pixel basis from FITS images.

3. Input

3.1. SLOPERROR Input

SLOPERROR takes all of its input from either the command line or namelist file, which is set up by the startup script that is controlled by the pipeline executive or standalone. If the namelist is not specified, then all required inputs are expected from the command line. If both namelist and command-line inputs are specified, then the command-line inputs override the namelist values. Prior to reading namelist and/or command-line parameters, default values for the relevant parameters are assigned.

3.1.1. SLOPERROR NAMELIST Input

SLOPERROR reads the NAMELIST file whose name is passed to it by start-up script. The name of the NAMELIST is SLOPERRIN. The parameters that can be defined in the NAMELIST are listed in Table 1.

Namelist variable	**Description**	**Dim.**	**Type**	**Units**	**Default**
FITS_Image_Filename	Required SUR-mode (2-plane) input FITS-image filename.	256	C	-	Null
FITS_Image_PMask_Filename	Optional p-mask FITS-image	256	C	-	Null
FITS_Image_DMask_Filename	Optional d-mask FITS-image	256	C	-	Null
FITS_Out_Noise_Filename	Required output FITS-image filename containing slope and first difference uncertainties (1st and 2nd planes)	256	C	-	Null
Read_Noise	Required detector read-noise	1	R*4	electrons	-

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

Confusion_Sigma	Optional confusion noise estimate per pixel area.	1	R*4	electrons	0
Gain_chan1	Required channel 1 gain factor	1	R*4	e-/DN	-
Gain_chan2	Required channel 2 gain factor	1	R*4	e-/DN	-
Gain_chan3	Required channel 3 gain factor	1	R*4	e-/DN	-
Gain_chan4	Required channel 4 gain factor	1	R*4	e-/DN	-
Ignore_Frames1	Optional number of data samples ignored in ramp for slope estimate in first DCE of a sequence. Only used if not in FITS header.	1	I*2	-	0
Ignore_Frames2	Optional number of data samples ignored in ramp for slope estimate in DCE numbers > 1 of a sequence. Only used if not in FITS header.	1	I*2	-	0
PmaskFatal	Optional fatal PMask data bits	1	I*2	-	8192
DmaskFatal	Optional fatal DMask data bits	1	I*2	-	8192
Log_Filename	Optional output log filename	256	C	-	Stdout
Ancillary_File_Path	Pathname where supporting source files are installed.	256	C	-	./ (current directory)

Table 1. Namelist file

The following is an example of the contents of a SLOPERRIN NAMELIST file that might be used, where the values specified are not necessarily realistic.

```
 &SLOPERRIN
  Comment = 'Generic namelist file for sloperror, default values.',
  Ancillary_File_Path = '../sloperror_v1',
  FITS_Image_Filename = './testing/sursci2_cvtDNsec.fits',
  FITS_Image_PMask_Filename = './testing/pmask.fits',
  FITS_Image_DMask_Filename = './testing/dmask.fits',
  FITS_Out_Noise_Filename = './testing/noise.fits',
  Log_Filename = 'stdout',
  Read_Noise = 27,
  Confusion_Sigma = 15,
  Gain_chan1 = 5,
  Gain_chan2 = 5,
  Gain_chan3 = 5,
  Gain_chan4 = 5,
  Comment = 'Number of frames to ignore in input image if not in header',
  Ignore_Frames1 = 1,
  Ignore_Frames2 = 1,
  PMaskFatal = 8192,
  DMaskFatal = 8192,
 &END
```

3.1.2. SLOPERROR Command-Line Input

Alternatively, all inputs can be specified via command line, in which case, a namelist file is not needed. Or, inputs can be provided with a hybrid of both namelist and command-line mechanisms, with the latter overriding the former. Table 2 lists the available command-line options associated with their namelist-variable counterparts, as well as other options for specifying the namelist-file name and making the standard output more verbose.

3.1.3. SLOPERROR FITS Input

SLOPERROR uses the FITSIO library routines to read in the FITS-formatted input data file. The routines used are: fits_open_file, fits_read_keys_lng, fits_read_keys_dbl, fits_read_img, and fits_close_file.

Command-line option	**Variable**
-n	Namelist_Filename
-i	FITS_Image_Filename
-ip	FITS_Image_PMask_Filename
-id	FITS_Image_DMask_Filename
-o	FITS_Out_Noise_Filename
-l	Log_Filename
-a	Ancillary_File_Path
-r1	Ignore_Frames1
-r2	Ignore_Frames2
-rn	Read_Noise
-cn	Confusion_Sigma
-g1	Gain_chan1
-g2	Gain_chan2
-g3	Gain_chan3
-g4	Gain_chan4
-fp	PMaskFatal
-fd	DMaskFatal

-v (verbose switch)	-
-vv (super-verbose switch)	-
-d (debug switch)	-

Table 2. Command-line options

4. Processing

4.1. SLOPERROR Processing

SLOPERROR begins processing by writing its name and version number to standard output (verbose mode only), and then it initializes relevant variables with defaults values, and checks that the required namelist parameters and/or command-line parameters were passed to it. If this condition is not true, it writes a message stating which parameters are missing, recommends a look at this document, and terminates by issuing an appropriate exit code to the pipeline executive; otherwise it proceeds as follows.

If an error occurs during processing, then an error message is written to standard output, a termination-status code is written to the log file, and an exit code to the pipeline executive issued.

After processing, the program name and version number, namelist filename (if used), input, and output filenames, values of other input parameters, date and time, processing time, and a termination-status code are written a log file.

4.2. SLOPERROR Processing

SLOPERROR operates in six phases: initialization, slope-image data input, noise model parameter and p, d mask inputs, noise computation, results output, and termination. This processing level is depicted in Figure 1.

4.2.1. SLOPERROR Initialization

SLOPERROR initializes itself by performing the following tasks.

A.) A message is printed to STDOUT (verbose mode only), which includes the program name and version number.

B.) If specified on the command line, the NAMELIST file is opened and read. If any errors are encountered, a message is printed, and execution aborts.

C.) The remaining command-line inputs are read and checked for correct data range, consistency, etc. If any errors are encountered, a message is printed, and execution aborts.

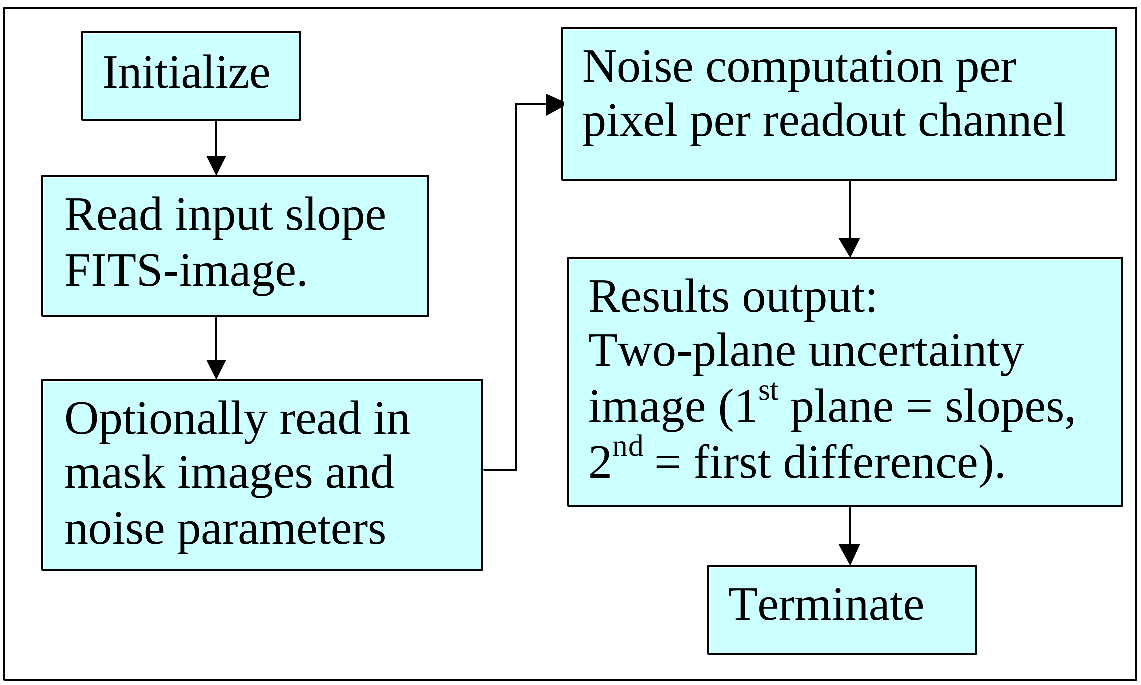

Figure 1. SLOPERROR data and processing flow

4.2.2. Slope-Image Data Input

The input image (namelist parameter: FITS_Image_Filename1) is read as a standard FITS file and stored in memory. The input image is assumed to be a data cube where (as assumed in SIRTF's 24µm SUR-mode format) plane 1 is the slope image with pixel values representing linear-least square fits to entire data ramps and plane 2 pixel values represent the difference between first and second sample reads from each ramp.

4.2.3. Mask and Noise Parameter Inputs

The SLOPERROR program can also optionally read in p-mask and d-mask images, where the p-mask records the "hardware" status of each pixel (eg. hot, dead etc…) and the d-mask records a history of fatal conditions encountered during prior pipeline processing of the DCE. Both images have 16 bits/pixel are are stored in memory. Accompanying the masks are user-specifiable "fatal" bit template values: namelist parameters PMaskFatal and DMaskFatal which are AND'd with the mask images to avoid fatal pixels in processing.

Parameters for the uncertainty computation are also read in. Five of these parameters are required (namelist parameters: Read_Noise and Gain_chan1 $\rightarrow$ Gain_chan4). If any of these are not specified, the program will abort with a message sent to standard output The Confusion_Sigma parameter which specifies the expected level of confusion noise per pixel is optional and defaults to zero if not specified.

4.2.4. Noise Computation

The MIPS-24μm SUR-mode consists of image data with pixel values represented by slope fits (m) to uniformly time-sampled ramp data. The slopes are computed on-board using a linear-least squares fit of the equation:

$$y_i = mt_i + c, \tag{1}$$

where $y_i = DN_{obs}$ is the observed (cumulative) DN after sample time t_i. m is computed by minimising c^2 where

$$c^2 = \sum_{i=N_{start}}^{N_{end}} w_i (y_i - mt_i - c)^2, \quad \text{and evaluating} \quad \frac{\partial c^2}{\partial m} = \frac{\partial c^2}{\partial c} = 0.$$

All weights w_i (or uncertainties) associated with the y_i are implicitly set to 1 in the on-board slope fit.

Carrying out the differentiations and solving the simultaneous equations for m leads to the analytic expression for observed slope:

$$m = \sum_{i=N_{start}}^{N_{end}} [f_1 - f_2 t_i] y_i , \qquad (2)$$

where $$f_1 = \frac{\sum_i t_i}{\left(\sum_i t_i\right)^2 - N_s \sum_i t_i^2}, \qquad f_2 = \frac{N_s}{\left(\sum_i t_i\right)^2 - N_s \sum_i t_i^2} \qquad (3)$$

N_{start} = First sample number to process in ramp
N_{end} = Maximum sample number to process in ramp
N_s = Number of samples to which slope-fit applies $= N_{end} - N_{start} + 1$

These last three parameters are further discussed in section 5.3.

We represent the total noise (variance) in any pixel with some measurement $y_i = DN_{obs}$ in the ramp using a three component model:

$$s^2(y_i) = \frac{s^2_{readnoise}}{g^2} + \frac{s^2_{confusion}}{g^2} + \frac{y_i}{g} , \qquad (4)$$

where $s_{readnoise}$ is the detector read-noise in *electrons*, $s_{confusion}$ is the confusion noise per pixel (or per resolution element) also in *electrons* and y_i is in measured *DN*. g is the channel dependent gain factor in units of *electrons/DN*. It is important to note that the confusion noise is an a-priori estimate since it is a strong function of source surface density on the sky.

The last term in equation (4) is the photon (Poisson) noise. This component is expected to introduce a correlation between the total noise between any two samples in the ramp since the total number of photons at any sample is the sum of all prior photon accumulations, including Poisson noise fluctuations. On the other hand, the read-noise and confusion-noise components are "external" and *uncorrelated* between samples. The covariance between any two samples is derived as follows. Assume we have any two samples labeled k and l (where $k \neq l$). The total uncertainty in each sample can be represented as the sum of *correlated* photon noise increments from previous samples and *uncorrelated* noise in the sample (e.g. read and confusion noise):

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

$$e_k = \left(\sum_{i=1}^{k} \Delta e_i\right) + e_{uk} \tag{5}$$

$$e_l = \left(\sum_{j=1}^{l} \Delta e_j\right) + e_{ul} \tag{6}$$

where $\Delta e_i \equiv (e_i - e_{i-1})$, $\Delta e_j \equiv (e_j - e_{j-1})$ and e_{uk} , e_{ul} are the *uncorrelated* components.

We are interested in the expectation value of the product (covariance) between the total error in samples k and l:

$$s_{kl}^2 \equiv \langle e_k e_l \rangle = \sum_{i=1}^{k}\sum_{j=1}^{l} \langle \Delta e_i \Delta e_j \rangle + \sum_{i=1}^{k} \langle \Delta e_i e_{ul} \rangle + \sum_{j=1}^{l} \langle \Delta e_j e_{uk} \rangle + \langle e_{uk} e_{ul} \rangle \tag{7}$$

The last three terms are zero since the uncorrelated components are not correlated with any other samples except themselves (when $k = l$). Furthermore, the photon-noise increments on the ramp are independent and are only correlated with themselves (when $i = j$). Assuming $k < l$, the double sum in equation (7) reduces to:

$$s_{kl}^2 \equiv \sum_{i=1}^{k} \langle \Delta e_i^2 \rangle \equiv \sum_{i=1}^{k} \langle s_{photon}^2 (y_i - y_{i-1}) \rangle \equiv \sum_{i=1}^{k} \frac{(y_i - y_{i-1})}{g} \equiv \frac{y_k}{g}, \tag{8}$$

where in the last step we have assumed Poisson errors and g is the gain in e-/DN. Equation (8) therefore implies the simple result for the covariance between any samples k and l in a ramp:

$$s^2(y_k, y_l) = \frac{y_k}{g} + s_{uncorrelated}^2(k,l) \text{ where } k \le l, \tag{9}$$

where for completeness, we have included the *uncorrelated* noise term which is zero when $k < l$. When $k = l$, equation (9) reduces to the “total” noise formula (equation 4) with the uncorrelated noise represented by the read and confusion noise components.

We now derive the variance in the slope accounting for correlated photon-noise between any two data samples y_i , y_j in the ramp (i.e. their covariance). As we shall see below, ignoring these correlations will lead to an underestimate of the slope error. The general error propagation equation (for slope m) obtained by expanding the standard variance formula to second order can be written:

$$s^2(m) = \sum_{i=N_{start}}^{N_{end}} \sum_{j=N_{start}}^{N_{end}} \left(\frac{\partial m}{\partial y_i}\right)\left(\frac{\partial m}{\partial y_j}\right) s^2(y_i, y_j), \qquad (10)$$

or equivalently:

$$s^2(m) = \sum_{i=N_{start}}^{N_{end}} \left(\frac{\partial m}{\partial y_i}\right)^2 s^2(y_i) + 2 \sum_{i=N_{start}}^{N_{end}} \sum_{j>i}^{N_{end}} \left(\frac{\partial m}{\partial y_i}\right)\left(\frac{\partial m}{\partial y_j}\right) s^2(y_i, y_j), \qquad (11)$$

where $s^2(y_i, y_j) \equiv \mathrm{cov}(y_i, y_j)$, i.e. the covariance between samples y_i and y_j . Equation (6) actually represents the sum of the variance and covariance in m where the first sum is the variance and the second double-sum term the covariance:

$$s^2(m) = \mathrm{var}(m) + \mathrm{cov}(m). \qquad (12)$$

The variance in slope as defined by Equation (11) with the slope defined by Equation (2) can be written:

$$\mathrm{var}(m) = \sum_{i=N_{start}}^{N_{end}} [f_1 - f_2 t_i]^2 s^2(y_i), \qquad (13)$$

With the noise model defined by equation (4), we can re-write equation (13) as follows:

$$\mathrm{var}(m) = \left(\frac{s^2_{readnoise} + s^2_{confusion}}{g^2}\right) \sum_{i=N_{start}}^{N_{end}} [f_1 - f_2 t_i]^2 + \frac{1}{g} \sum_{i=N_{start}}^{N_{end}} [f_1 - f_2 t_i]^2 y_i, \qquad (14)$$

To good approximation, we can predict the data-number y_i anywhere in the ramp from the fitted slope m and the elapsed integration time:

$$y_i \cong m t_i \equiv m i \Delta t, \qquad (15)$$

where Δt is the read-sample time (input FITS header keyword T_INT). Substituting equation (15) into (14), we have the final expression for the slope variance (i.e. the first sum-term in Equation 11):

$$\mathrm{var}(m) = \left(\frac{s^2_{readnoise} + s^2_{confusion}}{g^2} \right) \sum_{i=N_{start}}^{N_{end}} \left[f_1 - f_2 t_i \right]^2 + \frac{m}{g} \sum_{i=N_{start}}^{N_{end}} \left[f_1 - f_2 t_i \right]^2 t_i \,, \tag{16}$$

The covariance term (second term in Equation 11) can be written by making use of Equations (2) and (9) with zero *uncorrelated* noise term since $j > i$:

$$\mathrm{cov}(m) = 2 \sum_{i=N_{start}}^{N_{end}} \sum_{j>i}^{N_{end}} (f_1 - f_2 t_i)(f_1 - f_2 t_j) \frac{y_i}{g} \tag{17}$$

Making use of Equation (15), this can be written:

$$\mathrm{cov}(m) = \frac{2m}{g} \sum_{i=N_{start}}^{N_{end}} \sum_{j>i}^{N_{end}} (f_1 - f_2 t_i)(f_1 - f_2 t_j) t_i \tag{18}$$

The final slope uncertainty is computed from Equation (12) where $s(m) = \sqrt{s^2(m)}$ with var(m) and cov(m) defined by Equations (16) and (18) respectively. This is computed for each of the four independent read-out channels (with characteristic gains $g1 \rightarrow g4$) and every pixel there-in.

The uncertainty in the **difference image plane**, where a pixel value is defined as the difference between the first and second reads in the ramp divided by the read-sampling time:

$$d = \frac{y_2 - y_1}{\Delta t}, \tag{19}$$

can be derived using the same formalism. Using Equation (19) in our error-propagation Equation (11) leads to:

$$s^2(d) = \frac{1}{\Delta t^2} \left[s^2(y_1) + s^2(y_2) - 2s^2(y_1, y_2) \right], \tag{20}$$

Using Equations (4) , (9) and (15), we have the final result :

$$s(d) = \frac{1}{\Delta t}\left(\frac{2s^{2}_{readnoise}}{g^{2}} + \frac{2s^{2}_{confusion}}{g^{2}} + \frac{m\Delta t}{g}\right)^{1/2}. \tag{21}$$

4.1.5. FITS-Image Output

The primary product of this software is a 32-bit FITS image of uncertainty values in the same units as the input slope image (usually DN/sec). The uncertainty slope image is contained in the *first plane* of the output FITS cube and the *second plane* contains uncertainties in the difference image in the same units.

4.1.6. Termination

Summary output is appended to the log file (the log file is created if previously non-existent), which includes diagnostic reports for the Q/A Subsystem and the appropriate exit code issued to be picked up by the pipeline executive. A detailed list of log file contents is given in Section 6.1.2.

5. Algorithm

5.1. Algorithm Specifics

The simple algorithm employed in this software has been adequately described in the previous section. As a detail, "fatal" pixels defined in the p- and d-masks are handled by replacing them by NaNs in the output image.

All fatal bit settings in the (input) p- and d-mask used by SLOPERROR are defined as input command-line/namelist parameters (see Table 1). Their variable names and default settings are defined below.

PMaskFatal = 8192 (bit 13)

DMaskFatal = 8192 (bit 13)

5.3. Algorithm-Implementation Details

Version 1.0 of SLOPERROR uses a specific algorithm to compute the first (N_{start}), last (N_{end}) and total number of read samples (N_s) for estimating the noise via equation (8). This is specific to image data acquired with SIRTF's 24μm array. The algorithm is as follows.

As a consequence of the data acquisition method, the FITS header keyword DCENUM is used to determine which read samples are used in the slope-fit. This parameter represents a DCE sequence number and depending whether we're dealing with the first DCE in a sequence of exposures, the first non-zero (useable) read sample will be different according to the following logic:

If DCENUM = 0:

$$N_{start} = 3 + Ignore\_Frames1$$

Otherwise, if DCENUM > 0:

$$N_{start} = 1 + Ignore\_Frames2,$$

where Ignore_Frames1 and Ignore_Frames2 = 0. These represent any initial unwanted read samples and are defined by FITS header keywords IGN_FRM1 and IGN_FRM2 respectively. If these do not exist in the input image header, they are read from corresponding namelist/command-line parameters. If they are not specified in the namelist or command-line, default values of 0 are assigned.

The "N_{end}" parameter in the summations of equation (8) is computed from the FITS header keywords DCE_FRMS and FRMFLYBK. DCE_FRMS represents the number of commanded samples (associated Ge frames) in an acquisition cycle. FRMFLYBK represents the effective number of samples during a "fly-back" of the scan-mirror. This parameter is given by:

$$N_{end} = 0.25 * (DCE\_FRMS - FRMFLYBK)$$

5.4. Assumptions and Requirements

A. SLOPERROR treats the slope-image and difference image planes in the input FITS-image cube the same way when computing uncertainties. Both represent measured slopes.

B. SLOPERROR requires that the following keywords be present in the FITS-header of the input slope-image: “T_INT” – the sampling time interval between non-destructive reads in the ramp, “DCENUM”, “DCE_FRMS” and FRMFLYBK (see above). If any of these are not specified, an error message is sent to standard output and the program aborts.

C. A tutorial which lists all the command-line options can be generated by typing “sloperror” on the command line with no arguments. This tutorial will indicate which parameters are required and those which are optional. Optional parameters are assigned the default values defined in Table 1. If any of the required parameters are not specified, or are unacceptably out of range, the program will abort with a message sent to standard output. Below is the tutorial generated when “sloperror” is executed without command-line arguments.

```
Usage: sloperror
  -n  <input_namelist_fname>           (Optional)
  -i  <input_sur_mode_image_fname>     (Required)
  -ip <input_pmask_image_fname>        (Optional)
  -id <input_dmask_image_fname>        (Optional)
  -o  <output_uncertainty_image_fname> (Required)
  -rn <read_noise>                     (Required, units=e-)
  -cn <confusion_noise>                (Optional, Default=0, units=e-)
  -g1 <gain_channel_1>                 (Required, units=e-/DN)
  -g2 <gain_channel_2>                 (Required, units=e-/DN)
  -g3 <gain_channel_3>                 (Required, units=e-/DN)
  -g4 <gain_channel_4>                 (Required, units=e-/DN)
  -r1 <ignore_initial_frames1>         (Optional, Default=0)
  -r2 <ignore_initial_frames2>         (Optional, Default=0)
  -fp <unusable_pmask_bits>            (Optional, Default=8192)
  -fd <unusable_dmask_bits>            (Optional, Default=8192)
  -l  <log_fname>                      (Optional, Default=`stdout')
  -a  <ancillary_file_path>            (Optional, Default = ./)
  -d  (prints debug statements)
  -v  (verbose output)
  -vv (superverbose output)
```

6. Output

6.1. SLOPERROR Output

SLOPERROR is capable of generating the following output:

A.) Standard-output processing and status messages.

B.) A 32-bit FITS image representation of pixels with uncertainties in input slope-image units. The *first plane* in the output cube contains uncertainties in the full ramp slope fit. The *second plane* contains uncertainties in the “second – first” difference values (also in slope units).

C.) A log file containing processing statistics, status messages and ancillary information.

All SLOPERROR disk output is written to the pathnames that are specified with the output filenames in the command-line or namelist inputs.

6.1.1 SLOPERROR FITS Output

SLOPERROR uses the FITSIO library routines to create FITS-formatted output data files. The routines used are: fits_read_key_lng, fits_insert_key_lng, fits_create_file, fits_open_file, fits_copy_hdu, fits_flush_file, fits_write_key, fits_update_key, fits_write_date, fits_write_key_str, fits_write_key_fixflt, fits_write_img, fits_get_hdrspace, fits_read_record, fits_write_record, and fits_close_file.

6.1.2 SLOPERROR Log-File Output

The information stored in the log file at the output of this program includes: program name and version number, values of all namelist and/or command-line inputs, a message indicating the type of calculation performed, status code, processing time, date and time, and a message indicating program termination. An example of the log file output is shown below.

```
sloperror_read_image: Keyword IGN_FRM1 not in image header
```

```
                       ...using command-line, namelist or default value of 1
sloperror_read_image: Keyword IGN_FRM2 not in image header
                       ...using command-line, namelist or default value of 1

Program sloperror, Version 1.0
Namelist File  = sloperror.nl
Input File 1 = ./testing/sursci2_cvtDNsec.fits
Input PMask Image = ./testing/pmask.fits
Unusable PMask Bits = 8192
Input DMask Image = ./testing/dmask.fits
Unusable DMask Bits = 8192
Output Image File = ./testing/noise.fits
Confusion Noise = 15.000000 electrons
Read Noise = 27.000000 electrons
Gain channel 1 = 5.000000 e-/DN
Gain channel 2 = 5.000000 e-/DN
Gain channel 3 = 5.000000 e-/DN
Gain channel 4 = 5.000000 e-/DN
Sampling integration time = 0.524288
DCE number = 1
Number of commanded frames in cycle = 84
Number of scan mirror fly-back frames = 4
Initial frames ignored for dcenum 1 = 1
Initial frames ignored for dcenum > 1 = 1
Ancillary Data-File Path = ../sloperror_v1
Verbose flag = 0
Super-verbose flag = 0
Debug flag = 0
Computed uncertainty image.
Program sloperror: Status Message: 0x0000
Normal exit from Function 0x0000: LOG_WRITER
sloperror Probs Warning: NaN's were produced in the results.
A total of 12 NaN's were produced in the results.
Processing time: 0.090000 seconds
Current date/time: Wed Jul  3 18:00:54 2002
Program sloperror, version 1.0, terminated.
```

7. Testing

SLOPERROR has been successfully unit-tested as a stand-alone program for a variety of different input cases. The tests were designed to check SLOPERROR robustness and capability of generating corrected results.

Here is a summary of the unit tests that were conducted:

1. Tested the noise model used by SLOPERROR directly on RAW-mode image data consisting of full data ramps and corresponding (simulated) SUR-mode slope images.

2. Executed SLOPERROR with inputs read from and output written to directories different from where the program was run. Both namelist and command-line input mechanisms were exercised.

3. Executed SLOPERROR with different p-mask and d-mask inputs to test that they integrate and function properly. Combined these with simulated MIPS-24μm data containing fatal pixels in the form of radiation-hits and NaNs.

4. Executed SLOPERROR for all combinations of input parameters, in order to test that they function properly.

5. Executed SLOPERROR on a non-square, large (1024×1024) image.

8. Usage Examples

Using a namelist file with verbose (-v) output re-directed to a file “out.log”:

```
SLOPERROR –n sloperror.nl -v | & tee out.log
```

Without using a namelist file:

```
SLOPERROR –i input_slopes.fits –a ../ancpath –ip pmask.fits
-id dmask.fits –o noise.fits –rn 27 –cn 15 –g1 5 –g2 5 –g3 5
```

```
–g4 5 –r1 1 –r2 1 –fp 8192 –fd 8192 -v
```

9. Glossary

DCE	Data Collection Event
DN	Data Number
IOC	In-Orbit Checkout
MIPS	Multi-band Imaging Photometer for SIRTF
SDS	Subsystem Design Specification
SIS	Software Interface Specification
TBD	To Be Determined
TBR	To Be Resolved