674-SO-43, Version 4.0,
SSC-PD-4032

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: SLOPECORR

18 September 2001

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	December 20, 2000
2.0	Read in an uncertainty image, perform error propagation and output a corresponding uncertainty image.	May 21, 2001
3.0	Changed option to set a d-mask bit ONLY if a pixel could NOT be linearized.	September 18, 2001
4.0	1. Changed algorithm to allow two "Ignore_Frame" parameters. 2. Re-arranged the DN_{lin} *vs.* DN_{obs} equation to avoid possible numerical instability when the non-linearity is very small. 3. Set pixels to the maximum value that can be linearized by the quadratic model if their slope values lie above the non-linearity curve.	February 25, 2002
4.1	Modified algorithm to reflect DCENUM keyword starting at zero.	July 29, 2002

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

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.

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 SLOPECORR program reads (SUR-mode) “slope” image data from a FITS file and corrects the slopes for detector non-linearity using a quadratic non-linearity model derived by the LINCAL program. The primary product of this software is a 32-bit/pixel output image of “linearized” slopes at pixel locations corresponding to the input “slope” image with the same units. Optionally, the software can read in an uncertainty image and output a corresponding uncertainty image for the linearized slope data.

The software also optionally reads in p-mask, d-mask and c-mask images to handle “fatal” pixels where the non-linearity correction cannot be applied. This program is specifically designed to linearize image data acquired with SIRTF’s 24μm array in the “SUR-mode”. SLOPECORR is written in standard C.

2.1. SLOPECORR Requirements

SLOPECORR 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, a non-linearity model image, uncertainty and mask images.

C.) Produce as primary output a new SUR-mode image with a “linearized” slope plane.

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 SLOPECORR 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 (p- and d-mask inputs)

SOSDL-SIS-CL-3001 (calibration inputs)

SOSDL-SIS-CL-3003 (c-mask image input)

2.3. Version History

2.3.1. Version 1.0

Initial version created on December 20, 2000.

2.3.2. Version 2.0

Version 2.0 includes the option of reading in an uncertainty image to produce corresponding uncertainties in linearized slopes using formal error propagation.

2.3.3. Version 3.0

Version 3.0 changes the option of updating the d-mask if a pixel was linearized to only if a pixel could not be linearized.

2.3.4. Version 4.0

Version 4.0 includes the following: a modification to accommodate two initial Ignore_Frame parameters, a refinement to the DN_{lin} *vs.* DN_{obs} equation for the quadratic case (Eqn. 8, Appendix I) to avoid possible cases of numerical instability when the non-linearity is very small, and, if pixels have slope values which lie above the maximum that can be linearized according to the non-linearity model, they are forced to the maximum value.

2.3.5. Version 4.1

Modified the algorithm to account for the DCENUM keyword starting at zero.

2.4. Liens

No major liens have been identified.

3. Input

3.1. SLOPECORR Input

SLOPECORR 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. SLOPECORR NAMELIST Input

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

Namelist variable	Description	Dim.	Type	Units	Default
FITS_Image_Filename1	Required input FITS-image filename containing slopes to be linearized.	161	C	-	Null
FITS_Image_Filename2	Required input FITS-image filename containing quadratic-model coefficients and their uncertainties.	161	C	-	Null
FITS_Image_Filename3	Optional input FITS-image filename containing uncertainties in slopes and first difference.	161	C	-	Null
FITS_Image_PMask_Filename	Optional p-mask FITS-image	161	C	-	Null
FITS_Image_DMask_Filename	Optional d-mask FITS-image	161	C	-	Null
FITS_Image_CMask_Filename	Optional c-mask FITS-image	161	C	-	Null
FITS_Out_Filename1	Required output FITS-image filename containing linearized slopes (and first difference).	161	C	-	Null
FITS_Out_Filename2	Optional output FITS-image filename containing slope and first difference uncertainties.	161	C	-	Null

Ignore_Frames1	Number of initial samples to ignore if image cube is the first in a sequence.	1	I*2	-	0
Ignore_Frames2	Number of initial samples to ignore if image cube has order > 1 in image sequence.	1	I*2	-	0
CmdFrm_Keyword	FITS keyword name designating total frame count	8	C	-	DCE_FRMS
PmaskFatal	Fatal PMask data bits	1	I*2	-	8192
DmaskFatal	Fatal DMask data bits	1	I*2	-	8192
CMaskFatal	Fatal CMask data bits	1	I*2	-	512
DMaskNotLin	DMask data bit flag indicating linearization *was not* performed	1	I*2	-	4096
Log_Filename	Optional output log filename	161	C	-	Stdout
Ancillary_File_Path	Pathname where supporting source files are installed.	161	C	-	./ (current directory)

Table 1. Namelist file

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

```
&SLOPECORRIN
 Comment = 'Generic namelist file for slopecorr, default values.',
 Ancillary_File_Path = '../slopecorr_v2',
 FITS_Image_Filename1 = './testing/SUR_mode_unsat.fits',
 FITS_Image_Filename2 = './testing/lincal_error_out.fits',
```

```
  FITS_Image_Filename3 = './testing/SUR_mode_unsat_errors.fits',
  FITS_Image_PMask_Filename = './testing/pmask.fits',
  FITS_Image_DMask_Filename = './testing/dmask.fits',
  FITS_Image_CMask_Filename = './testing/cmask.fits',
  FITS_Out_Filename1 = './testing/linearized.fits',
  FITS_Out_Filename2 = './testing/uncertainties.fits',
  Log_Filename = 'stdout',
  Comment = 'FITS header keyword for total frame count',
  CmdFrm_Keyword = 'DCE_FRMS',
  Comment = 'Number of frames to ignore for linearization',
  Ignore_Frames1 = 1,
  Ignore_Frames2 = 1,
  PMaskFatal = 8192,
  DMaskFatal = 8192,
  CMaskFatal = 256,
  Comment = 'Dmask bit indicating pixel was not linearized',
  DMaskNotLin = 4096,
 &END
```

3.1.2. SLOPECORR 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. SLOPECORR FITS Input

SLOPECORR 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

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

-i1	FITS_Image_Filename1
-i2	FITS_Image_Filename2
-i3	FITS_Image_Filename3
-ip	FITS_Image_PMask_Filename
-id	FITS_Image_DMask_Filename
-ic	FITS_Image_CMask_Filename
-o1	FITS_Out_Filename1
-o2	FITS_Out_Filename2
-l	Log_Filename
-a	Ancillary_File_Path
-g1	Ignore_Frames1
-g2	Ignore_Frames2
-k	CmdFrm_Keyword
-fp	PMaskFatal
-fd	DMaskFatal
-fc	CMaskFatal
-fn	DMaskNotLin
-v (verbose switch)	-
-vv (super-verbose switch)	-

-d (debug switch)	-

Table 2. Command-line options

4. Processing

4.1. SLOPECORR Processing

SLOPECORR 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, then 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 SLOPECORR Processing Phases

SLOPECORR operates in seven phases: initialization, slope-image data input, model-coefficient data input, optional uncertainty and (p-,d-,c-) mask inputs, linearity computation, results output, and termination. This processing level is depicted in Figure 1.

4.1.1. SLOPECORR Initialization

SLOPECORR 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.

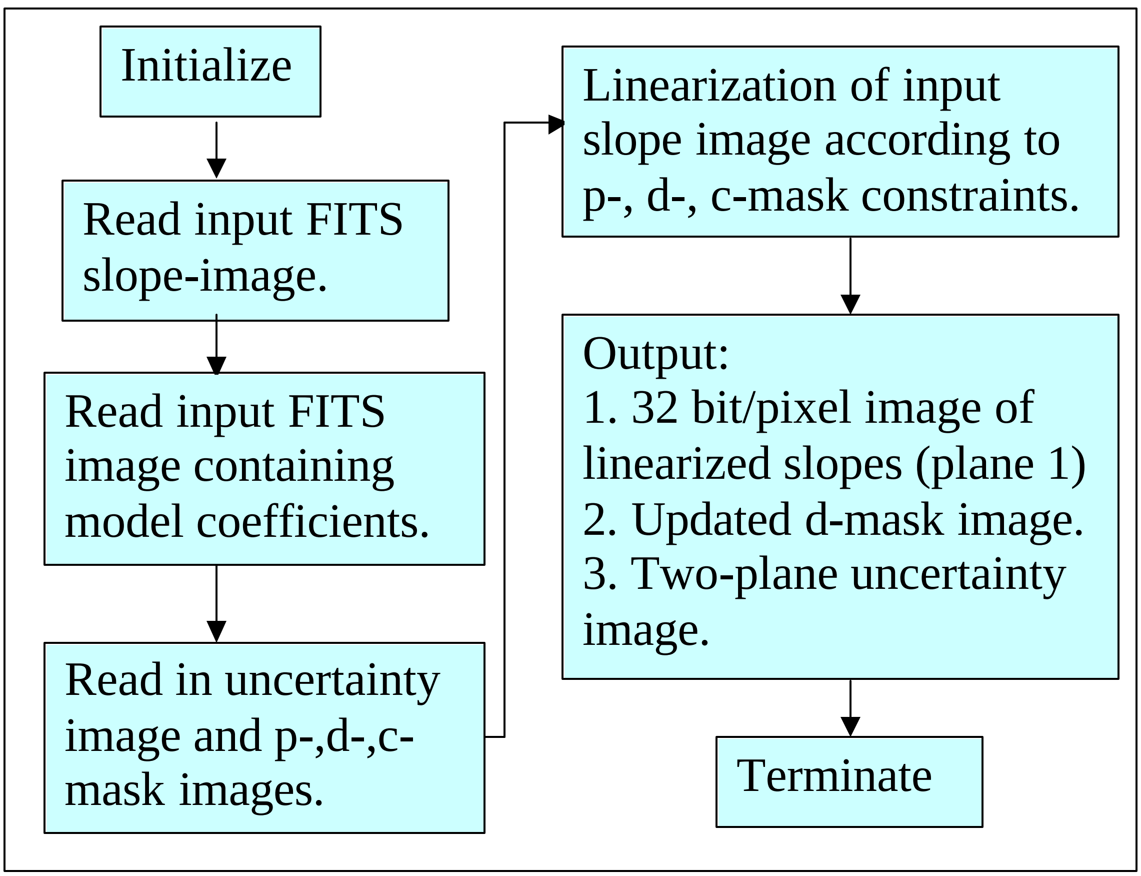

Figure 1. SLOPECORR data and processing flow

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

4.1.2. Slope-Image Data Input

The input image (namelist parameter: FITS_Image_Filename1) is read as a standard FITS file and stored in memory. If the input image is a data cube (as currently assumed in SIRTF's 24μm SUR-mode where plane 1 is the slope image and plane 2 the difference image), SLOPECORR assumes the first plane contains the slope values to be linearized.

4.1.3. Model Coefficients Input

Coefficients of the quadratic model used for the non-linearity correction are read from a multi-plane FITS image as generated by the LINCAL program (see Eqn. 1). A unique set of coefficients is stored in memory for each pixel. The linear coefficient m and quadratic coefficient A (see equation 1, Section 4.1.4) are read in as the combination A/m^2 from the first plane of LINCAL's output cube (input namelist parameter: FITS_Image_Filename2).

Uncertainties associated with the above quadratic model coefficients are read from the third plane of LINCAL's output cube. These are read as one-sigma values: $\sigma(A/m^2)$.

4.1.4. Optional Uncertainty and Mask Image Inputs

An optional uncertainty image corresponding to the input SUR-mode image cube can also be read. This reflects the input SUR-mode image with plane 1 containing the one-sigma uncertainties in slopes and plane 2 the one-sigma first-difference uncertainties.

SLOPECORR also optionally reads in p-mask (pixel), d-mask (dce) and c-mask (calibration) images. These are also stored in memory and used in the processing stage (see below).

4.1.5. Linearity Computation

The input (SUR-mode) slopes are initially derived from samples of non-destructive read images using on-board linear regression software. Here we correct for the effect of non-linearity between cumulative "detected" and "expected" photon counts using a novel method. This section gives a brief review of the equations used by SLOPECORR and a complete derivation is given in Appendix I. Corresponding uncertainties are described in Appendix II.

The method assumes that a sample of initial non-destructive reads (observed cumulative counts DN_{obs}) as a function of time t for a particular pixel follows a simple quadratic:

$$DN_{obs} = m_{lab} t - A_{lab} t^2, \quad (1)$$

where m_{lab} and A_{lab} are coefficients derived by the LINCAL program. Denoting the input (SUR-mode) slope as m_{sur} , its corresponding linearized value can be written analytically as follows:

$$m_{lin} = \frac{1 - \sqrt{1 - 4Lm_{sur}}}{2L} \quad , \quad (2)$$

where:

$$L = \left(\frac{A_{lab}}{m_{lab}^2} \right) \sum_{i=N_{start}}^{N_{end}} f_1 t_i^2 - f_2 t_i^3, \quad \text{and} \quad (3)$$

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

$$N_{start} = \text{First sample number (plane) to process in input cube}$$
$$N_{end} = \text{Maximum sample (plane) number in input cube}$$
$$N_s = \text{Number of samples to which slope-fit applies} = N_{end} - N_{start} + 1$$

The last three parameters are discussed in section 5.3. See Appendix I for a full derivation.

The formalism which leads to Eqns (2) and (3) is a new method and has the advantage of being independent of the intensity (absolute DN) in a particular pixel. For a given pre-determined set of coefficients (m_{lab} , A_{lab}) for each pixel and a knowledge of the effective number of samples used in determining the on-board (SUR-mode) slope, Eqns (2) and (3) completely determine the “true” slope (count-rate) corrected for detector non-linearity.

4.1.6. FITS-Image Output

The primary product of this software is a 32-bit FITS image of slopes in units equivalent to the input data. The linearized slope image is contained in the *first plane* of the output FITS cube and the second plane contains the difference image copied from the input image cube.

Additional processing information concerning pixels which could not be linearized is contained in an “updated” d-mask image (with the same filename as the input d-mask image).

An uncertainty image is produced if its *output filename* is specified (namelist parameter FITS_Out_Filename2, command-line option –o2). If no *input uncertainty image* (parameter FITS_Image_Filename3) was specified, the output uncertainty image will be filled with zeros.

4.1.7. 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-mask (e.g. hot or dead pixels) are handled by replacing them by NaNs in the output image. Fatal pixels defined in the d-mask are not linearized. These pixels are copied to the output image in their original form. Accordingly, a fatal p- and/or d-mask pixel is tracked and reported by updating a relevant d-mask bit to indicate it could not be linearized (defined with bit number 12, integer value 4096). See below for current bit-mask definitions.

Problems and anomalies in the output model generated by LINCAL are obtained from its accompanying c-mask image output. SLOPECORR will not linearize pixels for which a model could not be determined. These are copied to the output image in their original form. Accordingly, the relevant d-mask bit (bit number: 12, integer value 4096) is also set to indicate this has occurred. Furthermore, a pixel is not linearized if it is saturated. SLOPECORR detects saturated pixels by reading a relevant d-mask bit set by the SATMASK module.

Furthermore, there may exist pixels whose "raw" slope values lie above the maximum that can be linearized according to the *quadratic non-linearity model*. For this model, such pixels lead to negative values for the argument of the square root in Eqn. (2) above implying an unphysical solution. These pixels are set to the maximum value that can be linearized as defined by the non-linearity model: DN_{lin}(max) = $2\ DN_{obs}$ (see Appendix I). If this occurs, a warning is printed to stdout (even when the program is executed in non-verbose mode).

The input image cube containing the model coefficients (LINCAL output) must have the same (NAXIS1 and NAXIS2) dimension as the input SUR-mode slope-image. If not, an error is sent to stdout and the program is aborted.

5.2. Default Bit-Mask Settings

All fatal bit settings in the (input) p-mask, (input/output) d-mask and (input) c-mask used by SLOPECORR 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)

CMaskFatal = 512 (bit 9)

DMaskNotLin = 4096 (bit 12 - fatal; pixel *was not* linearized)

5.3. Algorithm-Implementation Details

Version 4.0 of SLOPECORR uses a specific algorithm to compute the first (N_{start}), last (N_{end}) and total number of read samples (N_s) for determining the linearized slope via Equations (2) and (3). 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 on whether we're dealing with the first DCE in a sequence of exposures, the first non-zero (valid) 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 equation (3) 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. SLOPECORR assumes that the slope-image containing slope-values to be linearized is the *first plane* in the input FITS-image cube.

B. SLOPECORR assumes that the input FITS-image cube containing the model coefficients (namelist parameter: FITS_Image_Filename2) is exactly that produced by the LINCAL program. The *first plane* of this cube is assumed to contain the parameter combination A/m^2 (see equation 1, Section 4.1.4). The *third plane* of this cube is assumed to contain the corresponding one-sigma uncertainties in this parameter.

C. The input uncertainty image cube must have the same dimensions as the input data image cube containing the slopes to be linearized. The NAXIS1, NAXIS2 and NAXIS3 header keywords must be equivalent.

D. SLOPECORR 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 for the initial SUR-mode slope estimate, DCENUM (see above), DCE_FRMS or its equivalent if specified via the namelist parameter CmdFrm_Keyword (see also above) and FRMFLYBK – the number of samples in a scan-mirror "fly-back" duration. If any of these are not specified, an error message is sent to stdout and the program aborts.

E. An uncertainty image is only produced if its *output filename* is specified in the namelist (parameter FITS_Out_Filename2) or command-line (with option –o2). If no *input uncertainty* image (parameter FITS_Image_Filename3) was specified, the output uncertainty image will be filled with zeros.

6. Output

6.1. SLOPECORR Output

SLOPECORR is capable of generating the following output:

A.) Standard-output processing and status messages.

B.) A 32-bit FITS image representation of pixels in terms of "linearized slopes" with units equivalent to the input data. This is represented by the *first plane* in the output FITS cube. The *second plane* contains the first difference values copied directly from the input image.

C.) An "updated" d-mask image containing information on pixels which could not be linearized.

D.) A two-plane 32-bit uncertainty image with the same data units corresponding to the output image of linearized slopes (first plane) and first difference values (second plane).

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

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

6.1.1 SLOPECORR FITS Output

SLOPECORR 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 SLOPECORR 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.

7. Testing

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

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

1. Tested the linearization algorithm used by SLOPECORR directly on RAW-mode image data consisting of different intensity levels. Results show that one set of model coefficients for a pixel with specific intensity level can be used to linearize all other pixels of varying intensity. This proves that only one set of model coefficients is necessary to linearize any image, independent of intensity.

2. Executed SLOPECORR 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 SLOPECORR with input images consisting of simulated MIPS 24μm images containing rad-hits and pixels loaded with NaNs.

4. Executed SLOPECORR with different p-mask, d-mask and c-mask inputs to test that they integrate and function properly.

5. Executed SLOPECORR with a model uncertainty image and tested that the error propagation is performed correctly.

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

7. Executed SLOPECORR on a non-square, large (COSMIC) image.

8. Usage Examples

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

```
SLOPECORR –n slopecorr.nl -v | & tee out.log
```

Without using a namelist file:

```
SLOPECORR –i1 input_slopes.fits –i2 lincal_model.fits –i3
input_uncert.fits –a ../ancpath –ip pmask.fits
-id dmask.fits –ic cmask.fits –o1 output_slopes.fits –o2
output_uncert.fits –g1 1 –g2 1 –fp 8192 –fd 8192
-fc 512 –fn 4096 -v
```

9. Glossary

DCE Data Collection Event

DN	Data Number
IOC	In-Orbit Checkout
SDS	Subsystem Design Specification
SIS	Software Interface Specification
TBD	To Be Determined
TBR	To Be Resolved

10. Appendix I

For comparison with equation (3) of section 4.1.5, we assume here $N_{start} = 1$, $N_{end} = N_s = N$. From a set of N non-destructive read samples (RAW-mode acquisition), software on board SIRTF will convert this data to SUR-mode format which consists of slopes m_{sur} for every pixel in the array. This is achieved using a linear regression algorithm and fitting for the equation:

$$y_i = m_{sur} t_i + c, \qquad (1)$$

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

$$c^2 = \sum_{i=1}^{N} (y_i - m_{sur} t_i - c)^2, \quad \text{and evaluating} \quad \frac{\partial c^2}{\partial m_{sur}} = \frac{\partial c^2}{\partial c} = 0 .$$

Carrying out the differentiations and solving these equations for m_{sur} leads to the analytic expression:

$$m_{sur} = \sum_{i=1}^{N} [f_1 - f_2 t_i] y_i, \qquad (2)$$

where $f_1 = \dfrac{\sum_i t_i}{\left(\sum_i t_i\right)^2 - N\sum_i t_i^2}$, $\quad f_2 = \dfrac{N}{\left(\sum_i t_i\right)^2 - N\sum_i t_i^2}$ (3)

We now outline the method used to relate m_{sur} to our desired linearized slope m_{lin} (see Fig.2). For a quadratic non-linearity model, we will assume that a sample of (RAW-mode) "non-destructive reads" from laboratory data can be fit by the following equation:

$$y_i = m_{lab} t_{lab} - A_{lab} t_{lab}^2 , \quad (4)$$

where m_{lab} and A_{lab} are coefficients computed using a least-squares regression algorithm in the LINCAL program.

For a "real" source with arbitrary uniform illumination level (i.e. constant photon output rate) m_{lin} (effectively the slope in the linear regime of a detector), its slope is related to a pre-determined laboratory value at any time t_i by equating the total accumulated counts:

$$m_{lab} t_{lab} = m_{lin} t_i \quad (5)$$

Using Eqn. (5), Eqn. (4) can be transformed to describe a new non-linearity curve for a source with arbitrary count rate m_{lin}:

$$y_i = m_{lin} t_i - m_{lin}^2 \left(\frac{A_{lab}}{m_{lab}^2} \right) t_i^2 . \quad (6)$$

Substituting Eqn. (6) into the general SUR-mode slope definition (Eqn. 2) leads to the following simple quadratic equation in m_{lin} :

$$L\, m_{lin}^2 - m_{lin} + m_{sur} = 0, \quad (7)$$

where L is a constant depending on pre-determined values of the model coefficients and summations over the sampling time interval t_i (see Eqn. 9 below). The only physically acceptable solution to Eqn. (7) is:

$$m_{lin} = \frac{1-\sqrt{1-4Lm_{sur}}}{2L} \quad , \qquad (8)$$

where

$$L = \left(\frac{A_{lab}}{m_{lab}^2}\right)\sum_{i=1}^{N} f_1 t_i^2 - f_2 t_i^3 , \qquad (9)$$

and f_1 and f_2 are defined by Eqn. (3).

One may expect to run into precision problems in Eqn.8 when L becomes very small (as a consequence of small α values, or when the non-linearity becomes negligible). Taking the limit as $L \rightarrow 0$ in Eqn. 9 formally yields $m_{lin} = m_{sur}$ which is expected when the non-linearity becomes very small. However as $L \rightarrow 0$, the numerator and denominator in Eqn. 8 both approach zero which could lead to numerical instability given the available 4-byte floating point precision used. To avoid this, we can convert Eqn. 8 to a more stable form by multiplying the numerator and denominator by the factor $1+(1-4LDN_{obs})^{1/2}$. A little algebra leads to the equivalent result:

$$m_{lin} = \frac{2m_{sur}}{1+\sqrt{1-4Lm_{sur}}} \quad . \qquad (10)$$

Equation (10) implies that no physical solution exists if $m_{sur} > (4L)^{-1}$. This is equivalent to saying that the maximum linearized value that can result from the above model occurs when $m_{sur} = (4L)^{-1}$ and hence,

$$m_{lin}(\max) = 2\,m_{sur} \quad . \qquad (11)$$

For a given pre-determined set of (m_{lab} , A_{lab}), which is given by LINCAL as the convenient combination A_{lab}/m_{lab}^2 for each pixel in the array, and a knowledge of the number of samples acquired in the RAW-mode, eqns (8) and (9) completely determine the "true" slope (or effective linear count-rate) that would result in the absence of detector non-linearity. Figure 2 shows a simple schematic of an exaggerated SUR-mode slope fit and its corresponding linearized slope.

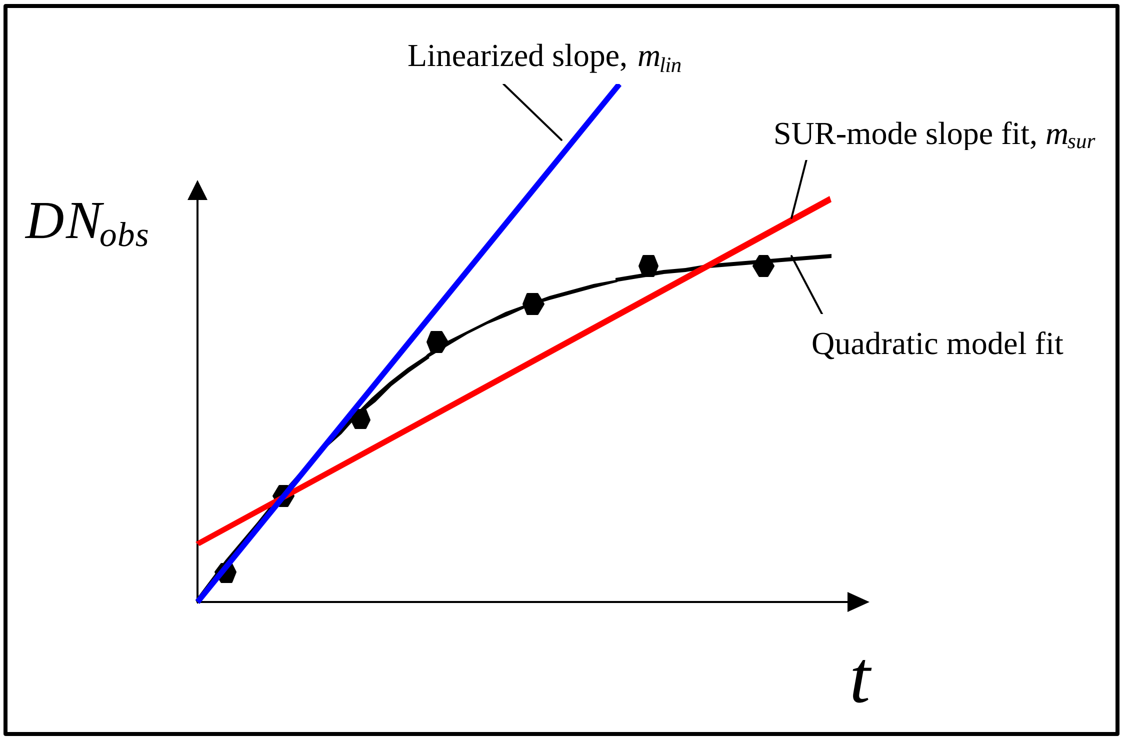

Figure 2: Schematic defining the different slopes

11. Appendix II

Given a one-sigma uncertainty in the fitted (measured) slope $\sigma(m_{sur})$, we derive the corresponding uncertainty in the linearized value $\sigma(m_{lin})$ using formal error propagation in Equations (8) and (9) (Appendix I). Since the non-linearity model coefficients (contained in the L parameter) and fitted slopes are independent, the variance can be written:

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

$$s^{2}_{mlin} = \left(\frac{\partial m_{lin}}{\partial L}\right)^{2} s^{2}_{L} + \left(\frac{\partial m_{lin}}{\partial m_{sur}}\right)^{2} s^{2}_{msur} .$$

Using Equations (8) and (9), the terms in this error formula are given by the following expressions:

$$\left(\frac{\partial m_{lin}}{\partial L}\right)^{2} = \left[\frac{m_{sur}}{L(1-4Lm_{sur})^{1/2}} - \frac{1-(1-4Lm_{sur})^{1/2}}{2L^{2}}\right]^{2}$$

$$\left(\frac{\partial m_{lin}}{\partial m_{sur}}\right)^{2} = (1-4Lm_{sur})^{-1}$$

$$s^{2}_{L} = s^{2}\left(\frac{A_{lab}}{m^{2}_{lab}}\right)\left[\sum_{i=1}^{N} f_1 t_i^2 - f_2 t_i^3\right]^{2}$$

s^{2}_{msur} = From input uncertainty image (error in slope fit or noise model).

The quantity $\sigma^2(A/m^2)$ is provided from the *third plane* of LINCAL's output image cube. The one-sigma uncertainties in the linearized slopes are stored in the first plane of the output uncertainty image. No error propagation is involved in the first difference value (second plane of the input SUR-mode cube). Uncertainties in this value are copied directly from the input uncertainty image to the output image.