674-SO-43, Version 2.0,
SSC-PD-4062

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: SURSIMSLOPER4

10 January 2002

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

California Institute of Technology
SIRTF Science Center

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.

Revision History

Version	Description	Date
1.0	Initial version	January 10, 2002
2.0	Data units in output FITS images have been changed to DN/second	February 19, 2002
3.0	Used more robust algorithm for slope fitting and uncertainty estimation. Changed algorithm so that DCENUM starts at 0.	July 29, 2002

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

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 SURSIMSLOPER4 program reads a FITS cube composed of "sample-up-the ramp" data represented by a set of non-destructive reads and converts this to a two plane FITS cube. The first output plane is a slope image derived using a linear least squares fit and the second plane is a difference image of the first two planes in the input cube. The input cube is often referred to as "RAW-mode" acquisition data and the output produced from this program, the "SUR-mode" image data.

Both planes in the output cube have pixel values in units of data number (DN) per second in signed 32-bit (R*4) format. This program is an update to the SUR-mode image simulator called SURSIMSLOPERI2 which simulates SUR-mode acquisition in signed 16-bit (I*2) integer format. The sole purpose of SURSIMSLOPER4 is to collapse RAW-mode image cubes that have undergone pipeline processing. SURSIMSLOPER4 is written in standard C.

2.1. SURSIMSLOPER4 Requirements

SURSIMSLOPER4 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 standard RAW-mode FITS cube and corresponding uncertainty cube.

C.) Produce as primary outputs a 2-plane SUR-mode FITS cube containing simulated slope-image and difference-image planes and corresponding uncertainty cube.

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 SURSIMSLOPER4 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)

SFO-SIS-3010 (16-bit integer RAW-mode DCE data)

2.3. Version History

2.3.1. Version 1.0

Initial version created on January 10, 2002.

2.3.2. Version 2.0

Created February 19, 2002. This version outputs slope and difference values in units of DN per second. Prior to this, they were output in DN per sampling-time interval.

2.3.3. Version 3.0

Created July 29, 2002. This version implements a more robust slope-fitting and uncertainty algorithm. The uncertainty algorithm accounts for correlations in the ramp. Also, the DCENUM parameter now starts at 0, not 1 as previously assumed.

2.4. Liens

One potential (although minor) lien is to read in p-mask and d-mask images and avoid computing slopes for pixels with fatal bit settings. The final status can be recorded in an output b-mask. This however will add redundancy since a user can always resort to the final multi-plane d-mask

produced by the pipeline if a peculiarity is noticed in the science data. Furthermore, this module will only be used in a pipeline which is NOT the primary science mode (ie. RAW-mode).

3. Input

3.1. SURSIMSLOPER4 Input

SURSIMSLOPER4 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. SURSIMSLOPER4 NAMELIST Input

SURSIMSLOPER4 reads the NAMELIST file whose name is passed to it by start-up script. The name of the NAMELIST is SURSIMSLOPEIN. The parameters that can be defined in the NAMELIST are listed in Table 1.

Namelist variable	Description	Dim.	Type	Units	Default
FITS_Image_Filename	Input n-plane FITS-image cube	161	C	-	Null
FITS_Noise_Image_Filename	Corresponding input n-plane uncertainty FITS-image cube	161	C	-	Null
FITS_Out_Filename	Output 2-plane FITS-image cube.	161	C	-	Null
FITS_Noise_Out_Filename	Output 2-plane uncertainty FITS-image.	161	C	-	Null
Log_Filename	Output log filename	161	C	-	Stdout
Ancillary_File_Path	Pathname where supporting source files are installed.	161	C	-	./ (current directory)

Ignore_Frames1	Number of initial planes to ignore if image cube is the first in a sequence.	1	I*2	-	0
Ignore_Frames2	Number of initial planes to ignore if image cube has order > 1 in image sequence.	1	I*2	-	0
T_Integration	Sampling integration time, only used if T_INT keyword is not in image header	1	R*4	sec	0.524288
DCE_Number	DCE number in sequence, only used if DCENUM keyword is not in header	1	I*2	-	1

Table 1. Namelist file

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

```
 &SURSIMSLOPEIN
 Comment = 'Generic namelist file for sursimslope, default values.',
 Ancillary_File_Path = '../sursimslopeR4_v1',
 FITS_Image_Filename = './testing/dntoflux_bcd.fits',
 FITS_Noise_Image_Filename = './testing/dntoflux_uncert_bcd.fits',
 FITS_Out_Filename = './testing/sursimslope_bcd.fits',
 FITS_Noise_Out_Filename = './testing/sursimslope_uncert_bcd.fits',
 Log_Filename = 'stdout',
 Comment = 'Number of initial frames to ignore:',
 Ignore_Frames1 = 1,
 Ignore_Frames2 = 1,
 Comment = 'sampling integration time:',
 T_Integration = 0.524288,
 Comment = 'DCE Number in sequence',
 DCE_Number = 1,
&END
```

3.1.2. SURSIMSLOPER4 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. SURSIMSLOPER4 FITS Input

SURSIMSLOPER4 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	**Value**
-n	Namelist_Filename
-i1	FITS_Image_Filename
-i2	FITS_Noise_Image_Filename
-o1	FITS_Out_Filename
-o2	FITS_Noise_Out_Filename
-l	Log_Filename
-a	Ancillary_File_Path
-p1	Ignore_Frames1
-p2	Ignore_Frames2
-t	T_Integration
-c	DCE_Number

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

Table 2. Command-line options

4. Processing

4.2 SURSIMSLOPER4 Processing

SURSIMSLOPER4 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.

4.3 SURSIMSLOPER4 Processing Phases

SURSIMSLOPER4 operates in seven phases: initialization, data input, linear least square slope computation, difference image computation, uncertainty computation, results output, and termination. This processing level is depicted in Figure 1.

4.3.1. SURSIMSLOPER4 Initialization

SURSIMSLOPER4 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.

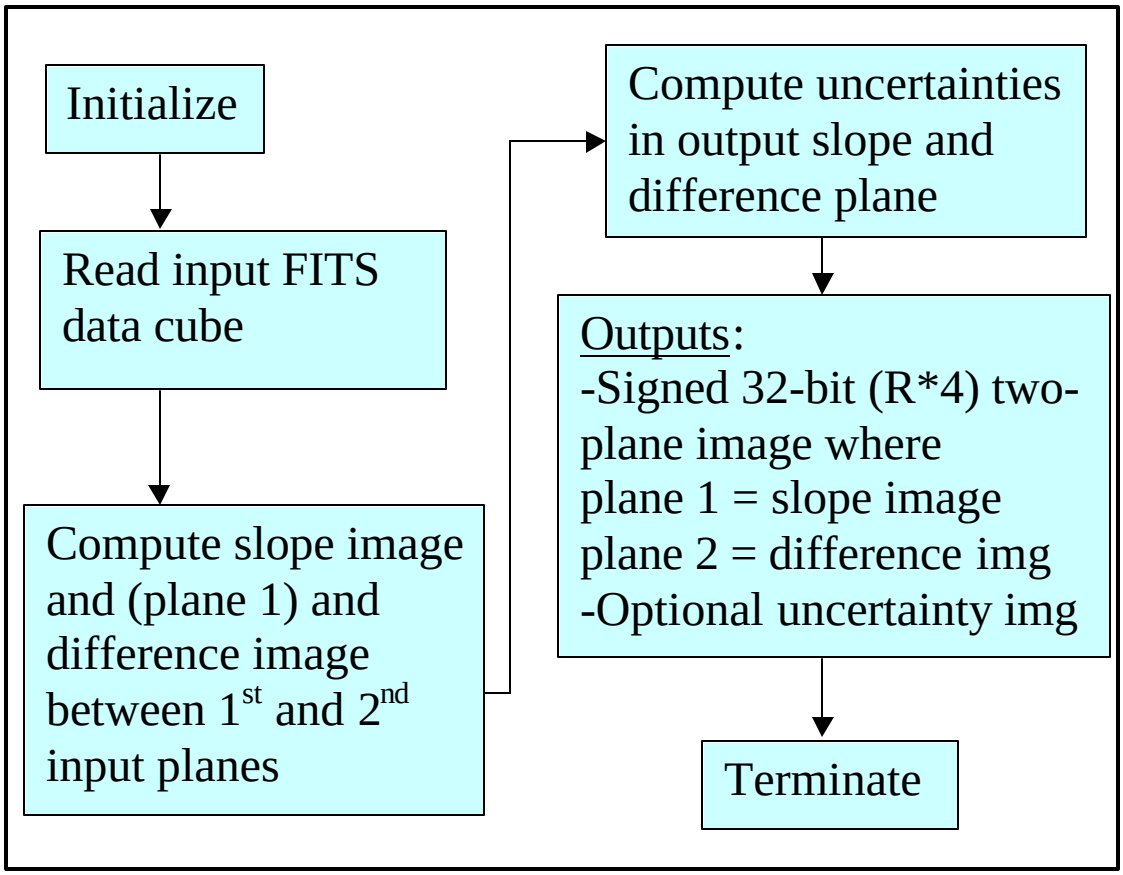

Figure 1. SURSIMSLOPER4 data and processing flow

4.3.2 FITS-Image Input

The input data and corresponding uncertainty image are read and stored in memory. This includes all data planes in the input FITS cubes, however, one can choose to ignore a fixed initial number of planes depending on the –p1 or –p2 option setting. Processing can therefore commence at a user-specified plane number in the input FITS cube.

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

4.3.3. Slope-Image Plane 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

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

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

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

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

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

These last three parameters are further discussed below.

The slope (Equation 2) is computed for every pixel with data from all planes in the input cube to create a “slope image”. However, the user can also specify which plane to process first via the Ignore_Frames1 or Ignore_Frames2 parameter. All plane numbers prior and up to this value will not be included in computation of the slope image. Pixel values in the slope plane image are in units of DN per second.

An additional parameter used in the algorithm is the namelist/command-line parameter DCE_Number. This is a consequence of the data acquisition method and is used to determine the first valid data plane in the input cube to commence processing. DCE_Number is initially searched for in the FITS header via the keyword DCENUM, however, it can be overidden by its equivalent namelist/command-line parameter. 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 data plane 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. See above for an explanation of this parameter.

The "N_{end}" parameter in the above equation 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:

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

4.3.4. Difference Image Plane Computation

The difference image plane is computed by subtracting plane number *START* (see above logic) from plane number *START* + 1 in the input FITS cube. The pixel values are then re-scaled in terms of DN per second.

4.3.5. Uncertainties

We compute the variance in the slope by accounting for correlations 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. We further assume that an individual sample y_i is drawn

from a distribution that is approximately Gaussian and that its error can be computed by Poisson methods. The general error propagation equation (for slope m) obtained by expanding the variance formula to second order can be written:

$$s^2(m) = \sum_{i=N_{start}}^{N_{end}} \left(\frac{\partial m}{\partial y_i}\right)^2 s^2(y_i) + 2\sum_{i=N_{start}}^{N_{end}} \sum_{j>i}^{N_m} \left(\frac{\partial m}{\partial y_i}\right)\left(\frac{\partial m}{\partial y_j}\right) s^2(y_i, y_j), \tag{4}$$

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 (4) 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:

$$s^2(m) = \mathrm{var}(m) + \mathrm{cov}(m). \tag{5}$$

The variance in slope as defined by Equation (4) with the slope defined by Equation (2) can be written:

$$\mathrm{var}(m) = \sum_{i=N_{start}}^{N_{end}} \left[f_1 - f_2 t_i\right]^2 s^2(y_i), \tag{6}$$

The covariance term (second term in Equation 4) can be computed by assuming there exists perfect correlation between any two samples in the ramp. In other words, if the counts in a pixel in sample-plane s is increased, then the counts in samples $>s$ will increase in direct proportion since the sample reads are cumulative. This is equivalent to saying that between any two samples y_i and y_j , the correlation coefficient r is 1:

$$r = \left(\frac{s^2(y_i, y_j)}{s(y_i)s(y_j)}\right) = 1 \quad \Rightarrow s^2(y_i, y_j) = s(y_i)s(y_j). \tag{7}$$

Although it's important to note that since there may still be a component of *uncorrelated* photon noise between the sample reads, this assumption will lead to a (*conservative*) over-estimate of the error. The covariance in slope as defined by Equation (4) with the slope defined by Equation (2) can be written:

$$\mathrm{cov}(m) = 2\sum_{i=N_{start}}^{N_{end}} \sum_{j>i}^{N_m} (f_1 - f_2 t_i)(f_1 - f_2 t_j) s(y_i) s(y_j), \tag{8}$$

where we have made use of the relation in Equation (7).

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

The final slope uncertainty is computed from Equation (5) where $s(m) = \sqrt{s^2(m)}$ with var(m) and cov(m) defined by Equations (6) and (8) respectively.

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:

$$d = \frac{y_2 - y_1}{\Delta t}, \tag{9}$$

can be derived using the same formalism. Using Equation (9) in our error-propagation Equation (4) and assuming perfect correlation (as defined by the relation in Equation (7)), we have the simple result:

$$s(d) = \frac{s(y_2) - s(y_1)}{\Delta t}. \tag{15}$$

4.3.6. FITS-Image Output

Processing statistics are given in both the standard output and log file. The calculation results are given in a 32-bit/pixel two-plane FITS image file.

4.3.7. 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 pipeline executive. A detailed list of log file contents is given in Section 6.1.2.

5. Algorithm

5.1. Algorithm Description

The simple algorithm employed in this software has been adequately described in the previous section. As a detail, pixels with NaN values in the input planes to be processed are “preserved” and carried through to the output image planes. NaN values are skipped and not included in computation of the slope-image and difference-image for that pixel. The number of NaN pixels in the output image is written to stdout.

5.2 Algorithm-Implementation Details

If pixels with values at the 16-bit saturation limits [-32768, 32767] are encountered in any of the input planes, a warning can be printed to stdout indicating the pixel location and its corresponding plane number. This warning is only produced if the "super-verbose" command-line (-vv) option is set (see Table 2). Processing then continues as normal with inclusion of the saturated pixel.

Furthermore, if the computed pixel values in either the slope or difference image planes exceed the allowable (signed) 16-bit range [-32768, 32767], a warning is written to stdout indicating the pixel location, the value is forced to the saturation value 32767 and processing continues as normal. This warning is automatically produced regardless of any "verbose" command-line option settings on execution.

Two namelist/command-line parameters used in this algorithm are T_Integration and DCE_Number. SURSIMSLOPER4 first searches the FITS header for the keyword equivalents of these parameters: T_INT and DCENUM. If either of these are not present in the header then they are overridden with the namelist/command-line value. If unspecified in the namelist/command-line then default values are used.

6. Output

6.1. SURSIMSLOPER4 Output

SURSIMSLOPER4 is capable of generating the following output:

A.) Standard-output processing and status messages.

B.) A 32-bit real two-plane FITS image representation of pixels in terms of slopes (plane 1) and a difference between the first and second input planes (plane 2). Pixel values are in units of DN per second in both output planes.

C.) An optional corresponding 32-bit 2-plane uncertainty image.

D.) A log file containing processing statistics and status messages.

All SURSIMSLOPER4 disk output is written to the pathnames that are specified with the output filenames in the command-line or namelist inputs.

6.1.1 SURSIMSLOPER4 FITS Output

SURSIMSLOPER4 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 SURSIMSLOPER4 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

SURSIMSLOPER4 has been successfully unit-tested as a stand-alone program for a variety of different input cases. The tests were designed to check SURSIMSLOPER4 robustness and capability of generating corrected results.

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

1. Executed SURSIMSLOPER4 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.

2. Executed SURSIMSLOPER4 with input image cubes consisting of MIPS (RAW-mode) test data with and without saturated pixels.

3. Executed SURSIMSLOPER4 for all combinations of input parameters, in order to test that they function properly.

8. Usage Examples

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

```
SURSIMSLOPER4 –n sursimslopeR4.nl -v | & tee out.log
```

Without using a namelist file:

```
SURSIMSLOPER4 –i1 input.fits –i2 uncert_inp.fits -v -a ../ancpath –p1 1 –p2 1
–t 0.524288 –c 2 –o1 output.fits –o2 uncert_out.fits
```

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