674-SO-43, Version 2.0,
SSC-PD-4057

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: LATIMFLAG

27 January 2003

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 Reach 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	September 11, 2001
2.0	The computational algorithm has been optimized in this version. Finite sums of exponentials have been replaced by their analytic counterparts, reducing the computation time considerably.	January 27, 2003

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 represents 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

LATIMFLAG reads data from a list of standard FITS images and applies a recursive algorithm to track and flag pixels containing a latent flux above the background noise due to bright sources in previous images. The software is generic enough for application to different instrumental and data acquisition methods used on SIRTF. The parameters for the latency model are read from an input table. The software only reports latents at non-fatal pixel locations as specified by p-mask and d-mask bit settings. An accompanying list of noise images is read to perform the latent-strength thresholding.

For each image in the input list, the primary outputs of LATIMFLAG are: an update of a relevant bit-word in a corresponding bit-mask image (the d-mask) to indicate that latents exist above the local noise, and optionally, a corresponding 32-bit latent strength image. The bit-word is only set if latents are found with strength above the noise in any pixels of an image. LATIMFLAG is written in ANSI/ISO C.

2.1. LATIMFLAG Requirements

LATIMFLAG 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 list of standard FITS images, accompanying list of noise images, mask images and model parameters.

C.) Produce as primary output an updated d-mask image if latent pixels are found and optionally, a 32-bit latent strength image.

D.) Provide exit codes to the pipeline executive and also provide 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 LATIMFLAG 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 input)

SOSDL-SIS-PD-3001 (p- and d-mask inputs)

2.3. Version History

2.3.1. Version 1.0

Initial version created on September 11, 2001.

2.3.2. Version 2.0

This version contains an optimization of the computational algorithm. Finite sums of exponentials have been replaced by their analytic counterparts, reducing the computation time considerably.

2.4. Liens

Further improvement in the execution speed of LATIMFLAG may be needed. A potential speed-up could be achieved by pre-filtering input images to search for the “brightest” pixels likely to leave latents. This may involve thresholding the input data with a user-specified signal-to-noise ratio prior to tracking latents.

3. Input

3.1. LATIMFLAG Input

LATIMFLAG 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. LATIMFLAG NAMELIST Input

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

Namelist variable	Description	Dim.	Type	Units	Default
FITS_Image_List_Filename	Required filename containing list of FITS-images.	512	C	Any, later converted.	Null
FITS_Image_Noise_List_Filename	Required filename containing list of corresponding noise images.	512	C	Same as above	Null
FITS_Image_PMask_Filename	Optional p-mask FITS-image	512	C	-	Null
FITS_Image_DMask_List_Filename	Optional filename containing list of d-mask FITS-images	512	C	-	Null

Latent_Parameter_Table	Required latent model parameter table	512	C	-	Null
PMaskFatal	Optional fatal PMask data bits	1	I*2	-	0
DMaskFatal	Optional fatal DMask data bits	1	I*2	-	0
Exp_Time_Keyword	Required exposure-time FITS keyword	161	C	-	Required
Frm_Time_Keyword	Required frame-time FITS keyword	161	C	-	Required
Beg_Frm_Keyword	Required begin-frame time or start observation time keyword	161	C	-	Required
Num_Samples_Keyword	Number of Fowler samples keyword, only set if Sampling_Flag = 2 below.	161	C	-	Null
time_read	Time interval between sample reads (Required).	1	R*4	seconds	Required
time_delay	Optional time delay from Beg_Frm time to first sample read.	1	R*4	seconds	0
time_settle	Optional settling time from target acquisition to Beg_Frm time.	1	R*4	seconds	0
Output_Flag	Create output latent strength image: 1 = Yes, 2 = No.	1	I*1	-	2
Sampling_Flag	Optional sampling mode: 1 = Normal ramp mode. 2 = Fowler sampling mode.	1	I*1	-	1

Per_Sec_Flag	Optional units type: 1 = input data in plain DN or electrons. 2 = data in units of sec^{-1}.	1	I*1	-	2
Conversion_Factor	Optional conversion factor to convert input data units to electrons.	1	R*4	$\frac{electrons}{inp\_data}$	1
SatLevel	Saturation threshold level	1	R*4	*electrons*	10^6
SNratio	Required signal-to-noise ratio for latent flagging.	1	R*4	-	3
Log_Filename	Optional output log filename	512	C	-	stdout
Ancillary_File_Path	Pathname where supporting source files are installed.	512	C	-	./ (current directory)

Table 1. Namelist file

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

```
&LATIMIN
 Comment = 'Generic namelist file for latimflag, default values.',
 Ancillary_File_Path = '../latimflag_v1',
 FITS_Image_List_Filename = './testing/latimflag_images_test.list',
 FITS_Image_Noise_List_Filename = './testing/latimflag_noise_images.list',
 FITS_Image_DMask_List_Filename = './testing/latimflag_dmask_images.list',
 FITS_Image_PMask_Filename = './testing/pmask_ch1.fits',
 Latent_Parameter_Table = './testing/IRAC_latent_parameters.tbl',
 Exp_Time_Keyword = 'EXPTIME',
 Frm_Time_Keyword = 'FRAMETIM',
 Beg_Frm_Keyword = 'UTCS-OBS',
 Comment = 'Number of Fowler samples, only if Sampling_Flag=2 below:',
 Num_Samples_Keyword = 'AFOWLNUM',
 Comment = 'Time or clock period between sample reads:',
 time_read = 0.2,
 Comment = 'time-delay after Beg_Frm to first sample',
 time_delay = 0.08,
```

```
  Comment = 'settling time from source acquisition until exposure begins:',
  time_settle = 1.3,
  Comment = 'Create output latent strength image: 1 = Yes, 2 = No',
  Output_Flag = 2,
  Comment = '1 = Normal sampling ramp mode, 2 = Fowler sampling mode',
  Sampling_Flag = 2,
  Comment = '1 = in plain electron or DN units, 2 = in per_sec units',
  Per_Sec_Flag = 1,
  Comment = 'Conversion factor for above units:',
  Conversion_Factor = 1,
  Comment = 'Saturation level in electrons:',
  SatLevel = 250000,
  Comment = 'S/N ratio for latent flagging:',
  SNratio = 1,
  Log_Filename = 'stdout',
  PMaskFatal = 1024,
  DMaskFatal = 2,
 &END
```

3.1.2. LATIMFLAG 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. LATIMFLAG FITS Input

LATIMFLAG 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_List_Filename

-in	FITS_Image_Noise_List_Filename
-id	FITS_Image_DMask_List_Filename
-it	Latent_Parameter_Table
-ip	FITS_Image_PMask_Filename
-l	Log_Filename
-a	Ancillary_File_Path
-mp	PMaskFatal
-md	DMaskFatal
-k1	Exp_Time_Keyword
-k2	Frm_Time_Keyword
-k3	Beg_Frm_Keyword
-k4	Num_Samples_Keyword
-tr	time_read
-td	time_delay
-ts	time_settle
-o	Output_Flag
-u	SatLevel
-rn	SNratio
-s	Sampling_Flag

-c	Per_Sec_Flag
-j	Conversion_Factor
-v (verbose switch)	-
-vv (super-verbose switch)	-
-d (debug switch)	-

Table 2. Command-line options

4. Processing

4.1. LATIMFLAG Processing

LATIMFLAG begins processing by writing its name and version number to standard output (verbose mode only), 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 LATIMFLAG Processing Phases

LATIMFLAG operates in eleven phases: initialization, FITS data and noise image list inputs, (p-,d-) mask-image inputs if specified, latent model parameter input, start observation time-ordering, units conversion, recursive computation of latent "trap" image, computation of latent strength and thresholding, results output (d-mask update and optional latent strength image), input image FITS header update, and termination. This processing level is depicted in Figure 1.

4.1.1. LATIMFLAG Initialization

LATIMFLAG 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.

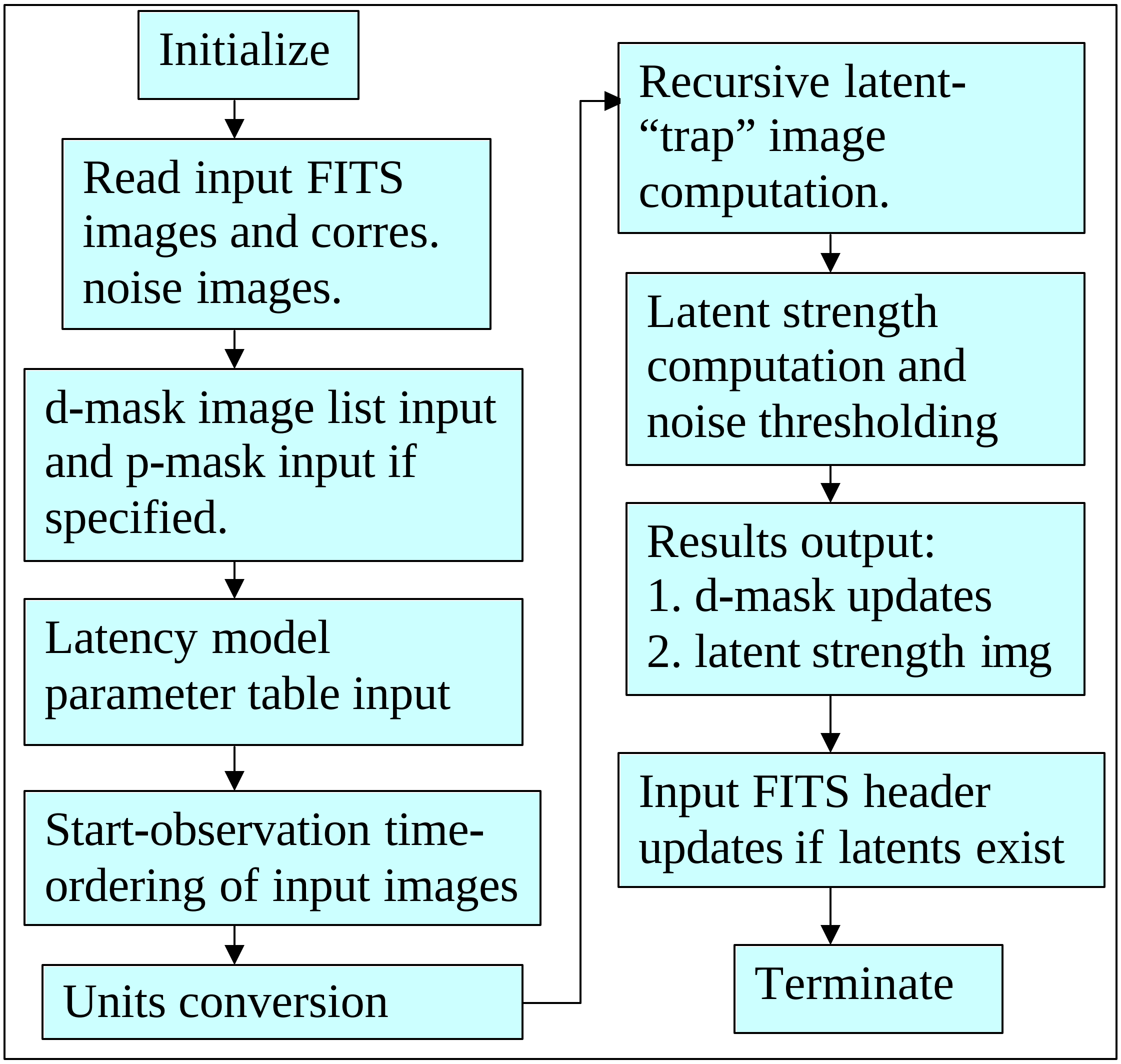

Figure 1. LATIMFLAG data and processing flow

4.1.2. FITS Image and Noise Image List Inputs

Each image in the input list of FITS images (namelist/command-line parameter: FITS_Image_List_Filename) is read recursively in the same loop where the main computation is performed. This is to ensure efficient use of memory which is freed immediately once the latent computation is complete for that image. All images must be composed of a single data plane and all must be unique in terms of their start-observation time. The images need not be listed in order of increasing start-observation time (see section 4.1.5).

In a similar manner, each noise image from the noise image list is mutually read with the science image. Each image in the noise image list must correspond in the same order to images in the science input list. The noise images are computed from the input science images before-hand using the SNESTIMATOR module. This module is used in a mode which computes the 1-σ interval (or standard deviation) in the pixel distribution within a window of specific size over an image. Outlier rejection must also be included to avoid bright sources biasing the result. The units of the noise images must be the same as the input science images.

4.1.3. Mask Image Inputs

LATIMFLAG optionally reads in p-mask (pixel) and d-mask (dce) images. There is one d-mask for each input DCE image which are listed in the same order in a separate file (namelist parameter: FITS_Image_DMask_List_Filename). These are read in simultaneously with the input science and noise images and used in the processing stage (see section 5.1).

The p-mask reports detector specific status information such as hot and dead pixels. The d-mask is “DCE” dependent and records pipeline processing information. An example is that we will want to exclude all pixels in input images affected by cosmic radiation hits since these do not produce latents (see section 4.1.7). To avoid these, a relavant d-mask bit is read which is set by previous execution of a rad-hit detection module on the input science images.

4.1.4. Latency Model Parameters

The parameters used to compute the latent fluence (in electrons) are read from a user-specified table in IPAC-table format (namelist parameter Latent_Parameter_Table). The parameters apply to a specific detector and are derived from prior laboratory characterization of latent-images. An example of an input parameter table for the IRAC channel 2 detector is shown below. The columns from left-to-right represent: the "trap" species index, the decay timescale for this species, the total number of available traps to be filled (the latents will decay by the release of these "traps") and the trap filling efficiency.

```
\char instrument = IRAC
\char computed_by = W. Glaccum
\char date_of_test = Thu Aug 30 11:30:00 2001
\int channel = 2
\int num_species = 5
\char Num_T_units = electrons
\char tau_units = seconds
| trap_id| tau_decay|   Num_T|  efficiency|
|i       |r         |i       |r           |
        1       0.50      400     4.00e-05
        2       7.50      400     6.00e-05
        3      35.00      400     1.50e-05
        4     180.00     3200     5.50e-07
        5    1000.00     3200     4.50e-07
```

4.1.5. Start-Observation Time Ordering

Each image read from the input list is ordered from the shortest observation time to the longest and the sorted list is stored in memory. The time-ordering is performed with respect to whatever is specified by the "Beg_Frm_Keyword" namelist parameter. Usually this keyword is the "UTCS-OBS" keyword expressed in seconds from the start date of January 1st, 2000. This keyword must be distinct in all the input science images.

4.1.6. Conversion of Units

By default, the algorithm implemented by the software assumes the units of the input science images are in *electrons per second* as specified by the namelist parameter Per_Sec_Flag = 2. However if this is not true, the user can specify that the input data are in plain DN or electrons by setting Per_Sec_Flag = 1. The software will then divide through by the exposure time during processing.

If the input data are in plain DN, the user can set a multiplicative conversion factor via the namelist parameter Conversion_Factor to convert this to electrons (Conversion_Factor = 1 if data units are in electrons). The user must then set Per_Sec_Flag = 1 to ensure that this gets converted to electrons/sec during processing. Likewise, if the data is in units of DN/sec, DN/sampling-time, or Jansky (which by definition is also measured in units of $\sec^{-1}$), the user must set Per_Sec_Flag = 2, AND, specify the relevant conversion factor to convert this to electrons/sec. Therefore in generality, the conversion factor should be computed as follows:

$$\text{If Per\_Sec\_Flag} = 1,\ \text{Conversion\_Factor} = \frac{electron\ units}{input\ \ data\ units}.$$

$$\text{If Per\_Sec\_Flag} = 2,\ \text{Conversion\_Factor} = \frac{electron\ \ per\ \text{second}\ units}{input\ \ data\ in\ \sec^{-1}\ or\ \ (sampling\ time)^{-1}\ \ units}.$$

4.1.7. Latent "Trap" Image Computation

Starting with the first image in the input list, we compute what's called a "latent-trap" image at the end of it's exposure. This specifies the amount of trapped charge in every pixel. These charge traps eventually decay and appear as latents in subsequent images. The latent trap image is effectively an image of the *number* of filled traps, which we label $N_F(t)_i$. The subscript *i* refers to the "trap-species" or type of latent trap distinguished by a characteristic decay-time and trap filling efficiency. $N_F(t)_i$ is unique to every pixel in the image. The latent-trap image is propagated <u>forward in time</u> and updated with each consecutive image in the sequence to record any new latent charge traps created by "new" bright sources.

The latent-trap image is computed from a latent-decay model provided by W. Glaccum (see http://ssc-et.ipac.caltech.edu/Instruments/IRAC/Downlink/Latent_report.pdf) which was initially derived to report latents in SIRTF's IRAC detectors. It is now generic enough to be applied to a number of different instruments. The number of filled traps of type i in the general case when a pixel is illuminated by a constant flux S_0 is given by:

$$N_F(t)_i = N_F(t_A)_i e^{-r_i(1+f_i)(t-t_A)} + N_{T_i}\left(\frac{f_i}{1+f_i}\right)\left[1-e^{-r_i(1+f_i)(t-t_A)}\right], \qquad (1)$$

where:

$f_i = \frac{c_i S_0}{r_i}$, where c_i = trap filling efficiency, r_i = decay rate $(1/t_i)$, t_i = e-folding decay time scale.

$N_F(t_A)_i$ = Number of filled traps at $t = t_A$.

N_{T_i} = Total number of available traps.

t_A = Time when flux (S_0) is turned on, i.e. when detector first acquires a bright source.

$(t_A = t_B - t_{settle}$, where t_B = begin frame time; t_{settle} = telescope settle time).

The time variable t in Eqn.(1) refers to the end of the previous image exposure so that the number of filled traps at $t=0$ for an image in a time-ordered sequence (i.e. $N_F(t_A)_i$), becomes exactly that accumulated at the end of the previous exposure. This formula is therefore used "recursively" from one image to the next in time-order and applied to every pixel. The flux S_0 is in units of electrons per second (converted internally by the software), time is in seconds and $N_F(t)_i$ is in electron (charge) units.

The parameters (N_{Ti} , c_i, t_i) are model dependent parameters specific to the detector. These are read from a table in IPAC format specified by namelist parameter Latent_Parameter_Table (see section 4.1.4).

As a further detail, it is important to note that latent charge is cumulative and is not completely removed following a frame reset. Cosmic radiation hits are known not to cause latents. These are masked out by reading the output from a cosmic rad-hit detection module.

For saturated pixels, the latency is indeterminate. Pixels driven to saturation are identified by thresholding the input data using a user-specified saturation level in units of *electrons* (namelist parameter SatLevel). If the counts exceed this value, we assume the worst possible case where all charge traps are filled (with $N_F(t) = N_T$ for that pixel). It has been observed in the laboratory that traps saturate long after the wells do, so this assumption is a reasonably conservative one.

4.1.8. Latent-Strength Image and Thresholding

The actual latent fluence or strength (in electrons) at the end of a given frame exposure depends on the sampling mode used to acquire photons and read out the detector. The two sampling modes handled by the current software are Fowler-sampling (specific to IRAC), and, normal ramp mode (specific to MIPS-24µm). These are considered in turn below.

4.1.8.1. Fowler Sampling Mode

This data taking mode consists of acquiring a number of "pedestal" and "signal" sample read pairs which we define as N_s (the Fowler number). The total counts are computed from the mean difference of all such pairs ($DN_j = signal_j - pedestal_j$):

$$S = \frac{1}{N_s}\sum_{j}^{N_s} DN_j \qquad (2)$$

For a latent trap species i (see above), the effective latent fluence for that species at the end of an exposure (in *electrons*) can be computed analogously from

$$L_i = \frac{1}{N_s}\sum_{j}^{N_s} FL_i(j) \qquad (3)$$

where

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

$$FL_i(j) = \int_{t_1}^{t_2} r_i N_F(t)_i \, dt \qquad (4)$$

$$\begin{aligned}
t_1 &= t_{j-1} + t_B + t_d \qquad (5)\\
t_2 &= t_{j-1} + t_B + t_d + T_{\exp}\\
t_{j-1} &= (j-1)t_c\\
j &= 1, 2, 3, \ldots N_s\\
t_B &= \text{Begin frame time (Namelist param: Beg_Frm_Keyword).}\\
t_d &= \text{Time delay from Beg_Frm to first sample read (Namelist param: time_delay).}\\
T_{\exp} &= \text{Image exposure time (namelist param: Exp_Time_Keyword).}\\
t_c &= \text{Time interval between sample reads (Namelist param: time_read).}
\end{aligned}$$

Evaluating the integral in Equation (4) with $N_F(t)_i$ given by Equation (1), we have:

$$FL_i(j) = \left(\frac{g_i - 1}{g_i}\right) N_{T_i} r_i T_{\exp} + \frac{\exp[-g_i r_i (j-1) t_c]}{g_i} \left[1 - \exp(-g_i r_i T_{\exp})\right] \left[N_F(t_A)_i - \left(\frac{g_i - 1}{g_i}\right) N_{T_i} \right] \times \exp[-g_i r_i (t_B + t_d - t_A)] \qquad (6)$$

Inserting Equation (6) into Equation (3) and simplifying, the latent strength from a single species can be written:

$$L_i = \left(\frac{g_i - 1}{g_i}\right) N_{T_i} r_i T_{\exp} + \frac{C}{g_i} \left[1 - \exp(-g_i r_i T_{\exp})\right] \left[N_F(t_A)_i - \left(\frac{g_i - 1}{g_i}\right) N_{T_i} \right] \exp[-g_i r_i (t_B + t_d - t_A)]$$

where

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

$$C = \frac{1}{N_s}\left[\frac{1-\exp(-N_s g_i r_i t_c)}{1-\exp(-g_i r_i t_c)}\right],$$

$$g_i = 1 + f_i$$

The total latent strength in *electrons* due to all trap types (species) *i* is then computed from the final sum:

$$F(latent) = \sum_i L_i \qquad (7)$$

4.1.8.2. Normal Ramp Sampling

This data taking mode consists of building a signal from a number non-destrunctive (cumulative) sample reads which we also define as N_s. Instead of computing a total count, this mode returns a "ramp" slope fitted to the samples. This is performed on board using a linear-least squares algorithm where the slope (in units of *DN/sec*) can be explicitly written:

$$S = \sum_j^{N_s} (f_1 - f_2 t_j) DN_j, \qquad (8)$$

where f_1 and f_2 are functions of "summations" over the sampling time interval t_j :

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

In this mode, the software does not directly read N_s from the image headers via a namelist parameter (i.e. Num_Samples_Keyword). Instead, it computes this internally from the exposure time (T_{exp} - Exp_Time_Keyword) and sample read-time interval (t_c - time_read):

$$N_s = \frac{T_{exp}}{t_c}$$

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

For a latent trap species i (see above), the effective latent fluence for that species at the end of an exposure (in *electrons/sec*) can be computed analogously from

$$L_i = \sum_{j}^{N_s} (f_1 - f_2 t_j) FL_i(j) \tag{9}$$

where

$$FL_i(j) = \int_{t_1}^{t_2} r_i N_F(t)_i \, dt \tag{10}$$

$$t_1 = t_{j-1} + t_B + t_d \tag{11}$$

$$t_2 = t_{j-1} + t_B + t_d + t_c$$

$$t_{j-1} = (j-1)t_c$$

$$j = 1, 2, 3, \ldots N_s$$

$$t_B = \text{Begin frame time (Namelist param: Beg\_Frm\_Keyword).}$$

$$t_d = \text{Time delay from Beg\_Frm to first sample read (Namelist param: time\_delay).}$$

$$t_c = \text{Time interval between sample reads (Namelist param: time\_read).}$$

Evaluating the integral in Equation (10) with $N_F(t)_i$ given by Equation (1), we have:

$$FL_i(j) = \left(\frac{g_i - 1}{g_i}\right) N_{T_i} r_i t_c + \frac{\exp[-g_i r_i (j-1) t_c]}{g_i} \left[1 - \exp(-g_i r_i t_c)\right] \left[N_F(t_A)_i - \left(\frac{g_i - 1}{g_i}\right) N_{T_i} \right] \times \exp[-g_i r_i (t_B + t_d - t_A)] \tag{12}$$

where

$$g_i = 1 + f_i$$

Accounting for all sample reads $j = 1, 2, 3\ldots N_S$ by evaluating Eqn. (9), the total latent strength in *electrons/sec* due to all trap types (species) i is computed from the final sum:

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

$$F(latent) = \sum_i L_i \tag{13}$$

4.1.8.3. <u>Thresholding</u>

The latent strength at the end of every image exposure is computed from either Eqns (3, 6, 7) or Eqns (9, 12, 13) as the latent-trap image is recursively propagated along the time-ordered image sequence by use of Eqn (1). All pixels in the current image whose latency leads to a total count (in electrons) that is greater than some factor of the local background noise will be flagged in the accompanying d-mask. In other words if

$$F(latent) > (S/N)s_L,$$

where (S/N) is a user-specified signal-to-noise ratio (namelist parameter: SNratio) and σ_L is the local noise computed from the distribution of pixel intensities within a window of specific size centered on the pixel in question.

It is worth mentioning that no latents can be recorded in the very first image of the sequence since there are no previous images (in our input list) from which to compute the latent-trap image. Below is a schematic summarizing the main steps used to flag and compute the strength of latents in a time-ordered sequence of N images with start observation-times $T_1 \rightarrow T_N$.

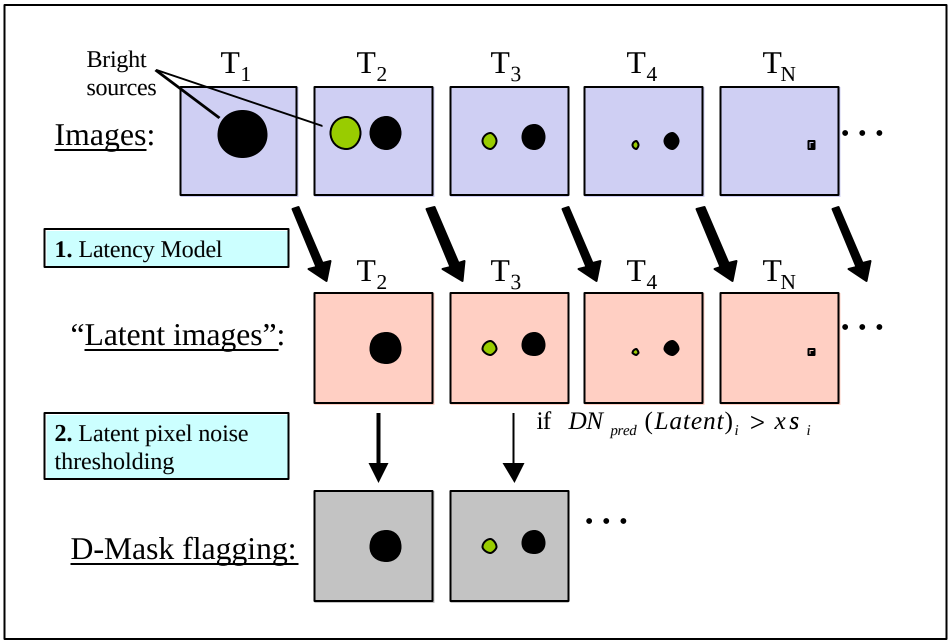

Figure 2. Computational proccessing steps

4.1.9. Results Output

The primary (default) output is an update of the d-mask corresponding to the image where a latent pixel was detected above the noise. Currently, bit number 5 (value 32) is set to indicate this. This bit-word is defined and set by the variable LATENTBIT in the include file latimflag.h.

Furthermore, the user can optionally output 32-bit latent strength images in units of either *electrons* (in Fowler-sampling mode), or, *electrons/sec* (in normal sampling mode). The latent strength images are given names "infile_latent.fits" where infile represents the input image name. The latent strength images are written to directories where the input science images reside.

4.1.10. Updating of Input FITS Headers

In addition to updating the d-mask for each image to indicate that a pixel was found with latent strength above the local noise, the program also updates the FITS header of the input image to indicate the filename of the latent-strength image produced IF the user chooses to create one. The following keyword string is appended at the end of the input FITS header:

```
LATIMFIL = ‘infile_latent.fits’ / Latent Image Filename
```

Where “infile” refers to the input image name.

4.1.11. 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 are handled by replacing them by NaNs in the input images. Likewise, un-usable pixels defined in the d-mask (e.g. rad hits) are also replaced by NaNs. Accordingly, all pixels that are read as NaNs from the input images are carried through as NaNs to the output (optional) latent strength images.

The maximum number of images that LATIMFLAG can handle is currently 5000. This is set by the parameter MAX_NUMBER_IMAGES in the include file latimflag.h.

5.2. Assumptions and Requirements

A. LATIMFLAG assumes that each image in the input list has a *single plane*, in other words with the standard FITS keyword values: NAXIS = 2 or NAXIS3 = 1. Every image must have the same dimensions. i.e. the NAXIS1 and NAXIS2 values must all be the same.

B. Each FITS image in the input list (namelist parameter: FITS_Image_List_Filename) is listed one per line and does not have to be in any specific order. Time-ordering is done internally. Each image must be distinct in its start-observation time.

C. The accompanying input list of d-mask and noise images must directly correspond, in the same order to images listed in the input list (from B).

D. The namelist variable "Num_Samples_Key" is only used by the software if Sampling_Flag = 2. In other words, if set to Fowler sampling mode. In normal ramp mode, (Sampling_Flag = 1), the number of samples in the ramp is computed from the ratio Exposure_Time/time_read (where time_read is the namelist variable representing the time between sample reads).

E. LATIMFLAG must be applied to images which have been dark subtracted, scanned for cosmic radiation hits (and these reported in corresponding d-masks), and, have NOT been linearized. It is also desirable to have the input science images and corresponding noise images (see G) in units of either *electrons* (in which case, Per_Sec_Flag = 1), OR, *electrons per second* (Per_Sec_Flag = 2). In these units, Conversion_Factor = 1. The software is only capable of converting data units using a *single*, (*pixel-independent*) conversion factor through the namelist parameter Conversion_Factor (or, a single gain if the input data is in DN). See section 4.1.6 for a general definition of Conversion_Factor.

F. Saturated pixels are identified by thresholding the input data with the use-specified namelist parameter "SatLevel". This value must be in plain *electron units*. This has been assigned with a (conservatively large) default value of 10^6 to ensure that all pixels are treated with the same algorithm (see Section 4.1.7) in case the user unintentionally omits this parameter from the namelist.

G. Each noise image must be derived from the science image by computing the width ($\pm 1\sigma$) of the distribution of pixel intensities within windows of specific size. This is computed robustly by the SNESTIMATOR module where it is recommended it be run in a mode which computes the standard-deviation within windows of size 1/8 the size of the original image on a side. Also, it is advised that an outlier rejection fraction of at least 0.1 be used to exclude bright sources. The noise images must be in the same units as the input images which is ensured by SNESTIMATOR.

H. The optional latent-strength images are written to directories where the input images reside, named "infile_latent.fits" where infile is the image name (without the .fits). The data units of these images can be either in plain *electrons* (in Fowler-sampling mode, Sampling_Flag = 2), or, *electrons/sec* (in normal sampling mode, Sampling_Flag = 1).

6. Output

6.1. LATIMFLAG Output

LATIMFLAG is capable of generating the following output:

A.) Standard-output processing and status messages.

B.) A new bit setting in the d-mask if a latent if found above the local noise in a pixel.

C.) An optional 32-bit FITS image of latent pixel strengths for each input image with filename "infile_latent.fits", where "infile" is the input FITS image name.

D.) A new keyword string (keyword: LATIMFIL) in the FITS header of the input image if a latent strength image is produced.

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

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

6.1.1 LATIMFLAG FITS Output

LATIMFLAG 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 LATIMFLAG 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

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

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

1. Tested LATIMFLAG on a simulated ensemble of six (256 $\times$ 256) IRAC images with relevant FITS headers updated to form a time-ordered list. The processing time was approximately 0.8 sec/image on a SUN-Blade 100 (500 MHz) workstation.

2. Tested LATIMFLAG on images containing different source intensities and varying lengths of time between the first and last image in a sequence to study the latent decay.

3. Executed LATIMFLAG 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.

4. Executed LATIMFLAG with input p- and d-mask images defined with different fatal bit-words, (e.g. cosmic radiation hits, hot/dead pixels) to test that they integrate properly with the output latent-strength images produced.

5. Executed LATIMFLAG exercising both the Fowler sampling mode option as applies to IRAC, and, normal ramp sampling as applies to the MIPS instrument.

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

7. Executed LATIMFLAG on non-square images.

8. Usage Examples

LATIMFLAG is a generic module that can be applied to detectors on SIRTF which use two different data acquisition (sampling) modes: Fowler sampling (IRAC), and, normal ramp mode consisting of a slope fit to a sequence of cumulative non-destructive sample reads (e.g. MIPS-24µm). The models used to predict the latent strength for these two modes were presented in section 4.1.7 and 4.1.8. All namelist parameters were defined in Table 1.

Below we show example namelist files (and corresponding command line specifications) that may be used as a first cut at flagging and predicting the latent strengths in image data acquired with IRAC (4-bands) and MIPS (primarily the 24µm band).

Both namelists are represented as files with extension ".nl" (e.g. latimflag.nl). To execute the software via a namelist with verbose (-v) output saved to a file "out.log", the command line is:

```
latimflag* –n latimflag.nl -v | & tee out.log
```

8.1. A Recipe for the Infrared Array Camera (IRAC)

Note: Text in blue parenthesized bold face are explanatory comments.

```
&LATIMIN
 Comment = 'Generic namelist file for latimflag, default values.',
 Ancillary_File_Path = './',
 FITS_Image_List_Filename = './latimflag_images.list',
 FITS_Image_Noise_List_Filename = './latimflag_noise_images.list',
 FITS_Image_DMask_List_Filename = './latimflag_dmask_images.list',
 FITS_Image_PMask_Filename = './pmask.fits',
```

```
  Latent_Parameter_Table = './IRAC_latent_parameters.tbl',
  Exp_Time_Keyword = 'EXPTIME',
  Frm_Time_Keyword = 'FRAMETIM',
  Beg_Frm_Keyword = 'UTCS-OBS',
  Comment = 'Number of Fowler samples, only if Sampling_Flag=2 below:',
  Num_Samples_Keyword = 'AFOWLNUM', (This keyword is expected for IRAC and
                                      must also set Sampling_Flag=2 below).
  Comment = 'Time or clock period between sample reads:',
  time_read = 0.2,
  Comment = 'time-delay after Beg_Frm to first sample',
  time_delay = 0.08,
  Comment = 'settling time from source acquisition until exposure begins:',
  time_settle = 1.5, (This should be checked, depends on settling of S/C
                      once a target is acquired).
  Comment = 'Create output latent strength image: 1 = Yes, 2 = No',
  Output_Flag = 1, (If yes, units of output data will be in electrons).
  Comment = '1 = Normal sampling ramp mode, 2 = Fowler sampling mode',
  Sampling_Flag = 2, (Ensure that Fowler sampling mode is selected).
  Comment = '1 = in plain electron or DN units, 2 = in per_sec units',
  Per_Sec_Flag = 1, (This may not be the case, but it is envisaged that
                     latimflag will be applied to IRAC data in plain electron
                     units).
  Comment = 'Conversion factor for above units:',
  Conversion_Factor = 1, (No conversion necessary if data is in plain electron
                          units).
  Comment = 'Saturation level in electrons:',
  SatLevel = 250000, (This is not a realistic value, must confirm with the
                      IST. This value must be in plain electron units).
  Comment = 'S/N ratio for latent flagging:',
  SNratio = 5,
  Log_Filename = 'stdout',
  PMaskFatal = 1024,
  DMaskFatal = 512, (Both PMaskFatal and DMaskFatal will have to be tuned).
 &END
```

If the user instead wishes to execute LATIMFLAG on the command line with no namelist file, the following may be used. All unspecified parameters assume their default values as indicated in Table 1.

```
latimflag* –i image_list.txt –it model_param.tbl –id
dmask_list.txt –in noise_list.txt –ip  pmask.fits –k1 EXPTIME
```

```
–k2 FRAMETIM –k3 UTCS-OBS –o 1 –k4 AFOWLNUM –tr 0.2 –td 0.08 –
ts 1.5 –rn 5 –s 2 –c 1 –j 1 –u 190000 –mp 1024 –md 256 –v
```

8.2. A Recipe for the Multi-band Imaging Photometer for SIRTF (MIPS)

Note: Text in blue parenthesized bold face are explanatory comments.

```
&LATIMIN
 Comment = 'Generic namelist file for latimflag, default values.',
 Ancillary_File_Path = './',
 FITS_Image_List_Filename = './latimflag_images.list',
 FITS_Image_Noise_List_Filename = './latimflag_noise_images.list',
 FITS_Image_DMask_List_Filename = './latimflag_dmask_images.list',
 FITS_Image_PMask_Filename = './pmask.fits',
 Latent_Parameter_Table = './MIPS24_latent_parameters.tbl',
 Exp_Time_Keyword = 'EXPTIME',
 Frm_Time_Keyword = 'FRAMETIM',
 Beg_Frm_Keyword = 'UTCS-OBS',
 Comment = 'Number of Fowler samples, only if Sampling_Flag=2 below:',
 Num_Samples_Keyword = 'AFOWLNUM', (This is redundant if Sampling_Flag=1).
 Comment = 'Time or clock period between sample reads:',
 time_read = 0.524288,
 Comment = 'time-delay after Beg_Frm to first sample',
 time_delay = 0, (A value of zero implies that the first sample integration
                  will occur immediately following the frame start time).
 Comment = 'settling time from source acquisition until exposure begins:',
 time_settle = 1.5, (This should be checked, depends on settling of S/C
                     once a target is acquired).
 Comment = 'Create output latent strength image: 1 = Yes, 2 = No',
 Output_Flag = 1, (If yes, units of output data will be in electrons/sec).
 Comment = '1 = Normal sampling ramp mode, 2 = Fowler sampling mode',
 Sampling_Flag = 1, (Ensure that normal ramp mode is selected).
 Comment = '1 = in plain electron or DN units, 2 = in per_sec units',
 Per_Sec_Flag = 2, (It is envisaged that MIPS-24 data will be expressed as a
                    slope in units of electrons/sec or electrons/read_time).
 Comment = 'Conversion factor for above units:',
 Conversion_Factor = 1, (It is likely that the data units will be in
                         electrons/read_time. If so, must convert to
                         electrons/sec. Please check).
 Comment = 'Saturation level:',
 SatLevel = 250000, (This is not a realistic value, must confirm with the
```

```
                    IST. This value must be in plain electron units).
  Comment = 'S/N ratio for latent flagging:',
  SNratio = 5,
  Log_Filename = 'stdout',
  PMaskFatal = 1024,
  DMaskFatal = 512, (Both PMaskFatal and DMaskFatal will have to be tuned).
 &END
```

If the user instead wishes to execute LATIMFLAG on the command line with no namelist file, the following may be used. All unspecified parameters assume their default values as indicated in Table 1.

```
latimflag* –i image_list.txt –it model_param.tbl –id
dmask_list.txt –in noise_list.txt –ip  pmask.fits –k1 EXPTIME
–k2 FRAMETIM –k3 UTCS-OBS –o 1 –tr 0.524288 –td 0 –ts 1.5 –rn
5 –s 1 –c 1 –j 1 –u 100000 –mp 1024 –md 256 –v
```

9. Glossary

DCE	Data Collection Event
DN	Data Number
IOC	In-Orbit Checkout
IRAC	Infra-Red Array Camera
IST	Instrument Support Team
MIPS	Multi-band Imaging Photometer for SIRTF
SDS	Subsystem Design Specification
SIRTF	Space Infrared Telescope Facility

SIS	Software Interface Specification
TBD	To Be Determined
TBR	To Be Resolved