674-SO-43, Version 4.0,
SSC-PD-4011

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: latimdetect

9 December 2002

California Institute of Technology
SIRTF Science Center

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

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

THIS PAGE IS INTENTIONALLY LEFT BLANK

SIRTF Science Center

Subsystem Design Specification

Prepared by:

Frank Masci

Concurred by:

Bill Latter

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	April 13, 2000
2.0	Upgraded to read in pmask and/or dmask images and detect bright pixels (latent images) only at pixel locations with usable mask values.	September 5, 2000
3.0	(i) Introduced a generalized latency model specified by model coefficients for a particular instrument. (ii) Changed output persistence-time images from 8-bit/pixel to 16-bit/pixel format to facilitate a larger (realistic) range of latent persistence times. (iii) IPAC-table output file is now optional.	November 9, 2000
4.0	Complete re-design of algorithm: This version computes latent fluxes arising from bright pixels in an input "history" science image at a "future" target image time and applies noise thresholding to flag latents in the target image using a bit-mask.	December 9, 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.

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

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 LATIMDETECT program reads pixel data from a science “history” FITS image at time T_h and a target science *noise image* at time T_t (where $T_h < T_t$) and computes the expected latent flux image at time T_t resulting from pixel strengths at T_h . A generic, user-specifiable two-parameter latency model is used. Thresholding of latent pixels in the target science *noise image* is then performed and as main output, all “above-noise” latent pixels are flagged in a bit mask. Optional outputs are a 32-bit latent-strength FITS image, a corresponding latent-uncertainty FITS image and an ascii table of latent pixel strengths and locations in IPAC table format. LATIMDETECT is written in standard ANSI/ISO C.

2.1. LATIMDETECT Requirements

LATIMDETECT 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/namelist parameters passed by the start up script and use them to run the program.

B.) Read in as input standard FITS files, mask-images.

C.) Produce as primary output an updated (DCE) d-mask or (BCD) b-mask bit-image reporting “above-noise” latent pixels in the science target image. Also, optionally output a latent strength and uncertainty image and a diagnostic table.

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 LATIMDETECT 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/output)

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

2.3. Version History

2.3.1. Version 1.0

Initial version created on April 13, 2000.

2.3.2. Version 2.0

Upgraded on September 5, 2000.

2.3.3 Version 3.0

Upgraded on November 9, 2000.

2.3.4. Version 4.0

Upgraded on December 9, 2002. This version contains a complete re-write of the algorithm (see review summary on page iv).

2.4. Liens

No major liens have been identified.

3. Input

3.1. LATIMDETECT Input

LATIMDETECT 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. LATIMDETECT NAMELIST Input

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

Namelist variable	**Description**	**Dim.**	**Type**	**Units**	**Default**
FITS_Hist_Image_Filename	Required input “history” image FITS filename	256	C	-	Null
FITS_Hist_Unc_Image_Filename	Optional input uncertainty “history” image FITS filename	256	C	-	Null
FITS_Hist_Image_PMask_Filename	Optional PMask image filename	256	C	-	Null
FITS_Hist_Image_DMask_Filename	Optional DMask image filename corresponding to “history” image	256	C	-	Null
FITS_Target_Image_DMask_Filename	Required DMask image filename corresponding to “target” image	256	C	-	Null

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

FITS_Target_Unc_Image_Filename	Required input uncertainty “target” science image filename	256	C	-	Null
FITS_Out1_Filename	Optional output latent strength image FITS filename	256	C	-	Null
FITS_Out2_Filename	Optional output uncertainty latent strength image FITS filename	256	C	-	Null
Data_Out_Filename	Optional output IPAC-table filename for locations and strengths of latent pixels.	256	C	-	Null
Log_Filename	Output log filename	256	C	-	stdout
Ancillary_File_Path	Pathname where supporting source (include) files are installed.	256	C	-	./
LatentSignalToNoiseRatio	Optional latent signal-to-noise ratio for latent thresholding	1	R*4	-	1.0
Acoeff_Latency_Model	A parameter of latency model	1	R*4	-	2.0
Bcoeff_Latency_Model	B parameter of latency model	1	R*4	-	1.0
PMaskFatal	Unusable PMask bits in PMask image	1	I*2	-	0
DMaskFatal	Unusable DMask bits in “history” DMask image	1	I*2	-	0
Data_Plane	FITS data planes considered as input: 1=All, 2=First, or 3=Last	1	I*2	-	1

Table 1. Namelist file

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

```
&LATIMDETECTIN
 Comment = 'Generic namelist file for latimdetect.',
 Ancillary_File_Path = '../latimdetect_v4',
 FITS_Hist_Image_Filename = './testing/bcd_slope1.fits',
 FITS_Hist_Unc_Image_Filename = './testing/uncert_bcd_slope1.fits',
 FITS_Hist_Image_PMask_Filename = './testing/pmask.fits',
 FITS_Hist_Image_DMask_Filename = './testing/bmask_slope1.fits',
 FITS_Target_Image_DMask_Filename = './testing/bmask_slope2.fits',
 FITS_Target_Unc_Image_Filename = './testing/uncert_bcd_slope2.fits',
 FITS_Out1_Filename = './testing/latent_fluxes.fits',
 FITS_Out2_Filename = './testing/latent_fluxes_unc.fits',
 Data_Out_Filename = './testing/latent_fluxes.tbl',
 Log_Filename = 'stdout',
 Comment = 'Data_Plane: 1=All, 2=First, 3=Last',
 Data_Plane = 2,
 Comment = 'Acoeff_Latency_Model is equivalent to decay time constant:',
 Acoeff_Latency_Model = 5.0,
 Bcoeff_Latency_Model = 1.0,
 LatentSignalToNoiseRatio = 3.0,
 PMaskFatal = 16896,
 DMaskFatal = 25088,
&END
```

3.1.2. LATIMDETECT 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. LATIMDETECT FITS Input

LATIMDETECT 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	Optional Namelist_Filename
-i	FITS_Hist_Image_Filename
-iu1	FITS_Hist_Unc_Image_Filename
-ip	FITS_Hist_Image_PMask_Filename
-id1	FITS_Hist_Image_DMask_Filename
-id2	FITS_Target_Image_DMask_Filename
-iu2	FITS_Target_Unc_Image_Filename
-o1	FITS_Out1_Filename
-o2	FITS_Out2_Filename
-o3	Data_Out_Filename
-l	Log_Filename
-a	Ancillary_File_Path
-S	LatentSignalToNoiseRatio
-A	Acoeff_Latency_Model
-B	Bcoeff_Latency_Model

-fp	PMaskFatal
-fd	DMaskFatal
-p	Data_Plane
-v (verbose switch)	-
-vv (super-verbose switch)	-
-d (debug switch)	-

Table 2. Command-line options

4. Processing

4.1. LATIMDETECT Processing

LATIMDETECT 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.

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

4.2 LATIMDETECT Processing Phases

LATIMDETECT operates in seven phases: initialization, data input, latent strength computation, latent uncertainty computation, latent-thresholding and flagging, results output, and termination. This processing level is depicted in Figure 1.

4.1.1. LATIMDETECT Initialization

LATIMDETECT 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.

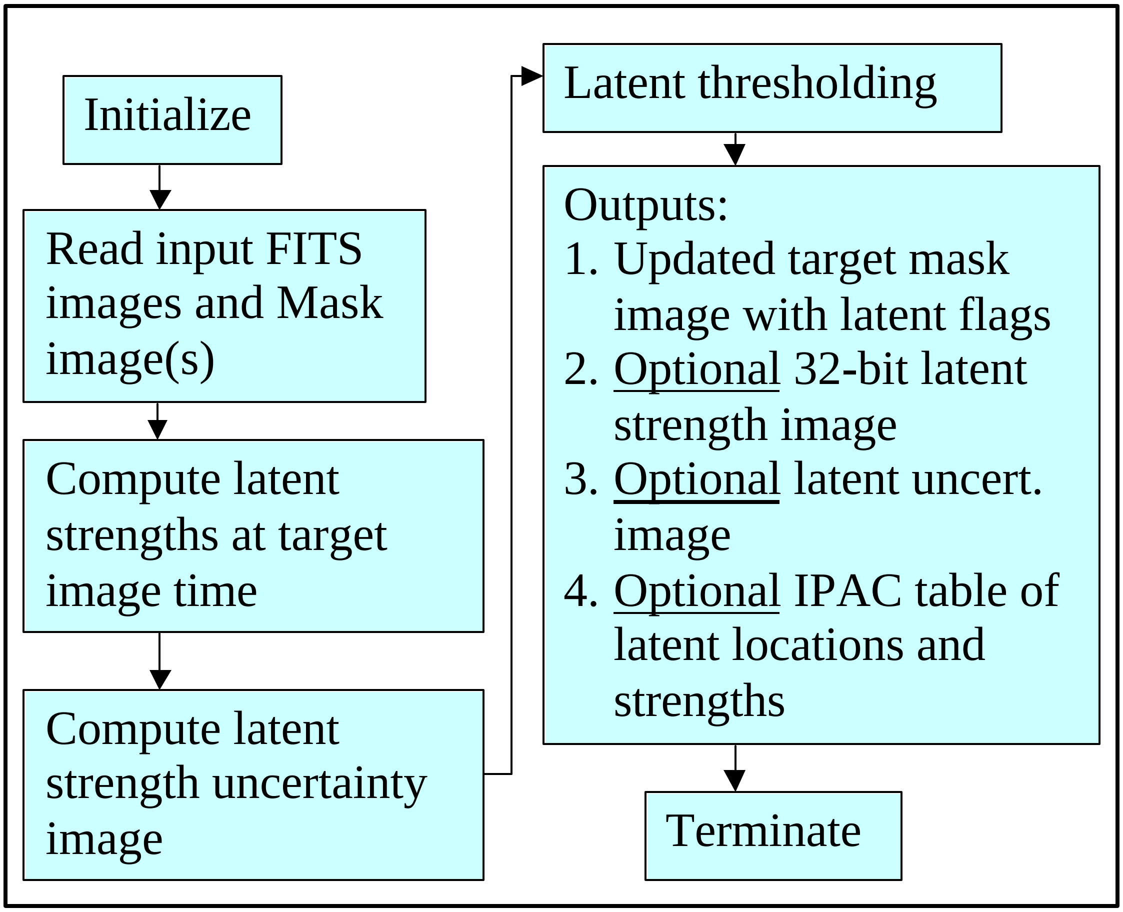
Initialize
Read input FITS images and Mask image(s)
Compute latent strengths at target image time
Compute latent strength uncertainty image
Latent thresholding
Outputs:
1. Updated target mask image with latent flags
2. Optional 32-bit latent strength image
3. Optional latent uncert. image
4. Optional IPAC table of latent locations and strengths
Terminate

Figure 1. LATIMDETECT data and processing flow

4.1.2. FITS-Image Input

The specified input images are read and stored in memory. This includes all data planes in the history and target noise image (depending on the –p option setting). The optional PMask and/or DMask images are read in as well, if specified.

4.1.3. Latent Strength Computation

For a given pixel in the history science image (namelist: FITS_Hist_Image_Filename) and target science uncertainty image (namelist: FITS_Target_Unc_Image_Filename) whose acquisition time difference is t, the expected latent pixel strength (above the target image background) is computed from the following two-parameter model:

$$F_{TOT}(t) = F_{bck} + \left[F_{TOT}(0) - F_{bck}\right]\exp\left\{-\left(\frac{t}{A}\right)^{1/B}\right\}, \qquad (1)$$

where F_{bck} is the median background flux in the science target image, $F_{TOT}(0)$ the initial pixel flux in the history image and A, B the latency model parameters (namelist: Acoeff_Latency_Model and Bcoeff_Latency_Model). The A parameter plays the role of an "e-folding latent decay time" and the B parameter controls the "tilt" of the latency decay curve. This model is generic enough that it can be used to characterize a range of detector behaviors.

F_{bck} is computed using the QATOOL module from prior processing in the BCD pipeline and its value is read from the "MEDIAN" FITS header keyword in the target science uncertainty image.

The time variable t in Equation (1) represents the acquisition time separation between the history and target images. A latent will usually start to decay at the end of the history image exposure and measured at the end of the target image read-out, however, we assume that the history and target images have the same integration time so that t is simply the difference in observation start times of these images (represented by the SCLK_OBS FITS header keyword):

$$t \equiv \text{SCLK\_OBS(Target)} - \text{SCLK\_OBS(History)} \;\; > 0\ . \qquad (2)$$

Given a history, target image pair, $F_{TOT}(t)$ is computed for every pixel in the array given that the following is satisfied: $F_{TOT}(0) > F_{bck} > 0$, otherwise the latent flux is set to zero.

4.1.4. Latent Uncertainty Computation

Optionally, a latent strength uncertainty image can be computed to accompany the latent strength image if both input history uncertainty and output uncertainty image filenames are specified prior to execution. The assumption here is that the main contributor to the latent uncertainty is photon noise in the target image background and history image input fluxes. No systematic uncertainties which account for derivations in the model parameters A, B or time difference t is assumed. From Equation (1), straightforward error propagation leads to:

$$s^{2}[F_{TOT}(t)] = (1-\Phi)^{2} s^{2}(F_{bck}) + \Phi^{2} s^{2}[F_{TOT}(0)], \qquad (3)$$

where

$$\Phi = \exp\left\{-\left(\frac{t}{A}\right)^{1/B}\right\}$$ (constant for given latency model and history/target image pair).

$s^{2}(F_{bck}) =$ variance in target science image, over all image pixels as computed by the QATOOL module from prior processing and read from the “TRIMEDSD” FITS header keyword. **Note**: this is expected to be an overestimate since all bright (above-background) pixels are included in it’s computation.

$s^{2}[F_{TOT}(0)] =$ variance in history image pixel, as read from the input history uncertainty image from prior processing.

4.1.5. Latent Thresholding and Flagging

The main output from LATIMDETECT is flagging of latent containing pixels in a bit-mask image. In this case, the output DMask corresponding to the target science image (namelist variable: FITS_Target_Image_DMask_Filename) is updated with a specific 16-bit integer. This bit is set if the final latent flux (Equation 1) satisfies:

$$F_{TOT}(t) > LatentSignalToNoiseRatio * s_{\text{target}} \qquad (4)$$

and

$$F_{TOT}(t) > F_{bck} \ ,$$

where LatentSignalToNoiseRatio is a user-specified namelist/command-line parameter, and s_{target} is the noise of a given pixel in the science target uncertainty image (namelist variable: FITS_Target_Unc_Image_Filename). The bit to set in the target mask image is specified by the DMASKLATBIT variable in the *latimdetect.h* include file. It’s value is currently set at 32 (i.e. bit 5).

4.1.6. FITS-Image and IPAC-Table Output

All processing statistics are written to standard output and a log file. The main output is an updated target D-Mask image showing the locations of latent-containing pixels. Optionally, the latent strength image, corresponding uncertainty image and a table reporting latent pixel strengths and locations can be generated. Section 6.0 gives more details on the information included in the outputs.

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 6.1.3.

5. Algorithm Specifics

5.1. Fatal and Masked Pixels

The simple algorithm employed in this software has been adequately described in the previous section. As a detail, fatal (unwanted pixel) values in the input history and target science images are detected from the three input mask images and "and'd" against the fatal template values (specified by PMaskFatal and DMaskFatal). These are handled by replacing them with NaN values during processing.

5.2. Diagnostic Messages

The number of above-background latent pixels in the target image is written to standard output and also written to a log file. Furthermore, a warning is printed to standard output if the fraction of all array pixels covered in latents exceeds some pre-specified threshold fraction. This fraction is specified by the LATFRAC variable in the *latimdetect.h* include file and is currently set at 0.5.

5.3. Requirements

A. LATIMDETECT requires that the "SCLK_OBS" keyword be present in the FITS header of the input history and target uncertainty images. Furthermore, the background and uncertainty keywords: "MEDIAN" and "TRIMEDSD" are required in the FITS header of the target uncertainty science image.

B. It is expected that the start observation time (SCLK_OBS value) of the history image precedes that of the target image. If this is not satisfied, the program will abort with a message sent to standard output.

6. Outputs

6.1. LATIMDETECT Output

LATIMDETECT is capable of generating the following outputs:

A.) Standard-output processing and status messages.

B.) An updated 16-bit integer mask image reporting above-background latent pixel locations.

C.) Optional 32-bit floating-point FITS image representation of latent pixel strengths and corresponding uncertainty image in input image units.

D.) An optional IPAC-table file containing latent pixel locations and strengths, along with ancillary information (see Section 6.1.2).

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

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

6.1.1 LATIMDETECT FITS Output

LATIMDETECT 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 LATIMDETECT FITS IPAC-Table Output

LATIMDETECT optionally outputs an IPAC-table containing the following information and results: program name and version number, date and time of calculation, input-image filenames, fatal P and D mask bits if specified by user, latent signal-to-noise ratio, latency model coefficients, latent pixel locations and strengths in input image units. Below is a snippet from such a table.

```
\character comment = Output from LATIMDETECT, version  4.00
\character Date-Time = Wed Nov 27 14:25:16 2002
\character inputHistoryFITSfile = ./testing/bcd_slope1.fits
\character inputHistoryUncertFITSfile = ./testing/uncert_bcd_slope1.fits
\character inputTargetUncertFITSfile = ./testing/uncert_bcd_slope2.fits
\character inputPMaskfile = ./testing/pmask.fits
\integer UnusablePMaskBits = 16896
\character input_History_DMaskfile = ./testing/bmask_slope1.fits
\integer UnusableDMaskBits = 25088
\character input_Target_DMaskfile = ./testing/bmask_slope1.fits
\real Model_Coeff_A = 5.000000
\real Model_Coeff_B = 1.000000
\real Latent Signal-to-Noise Ratio = 3.000000
|Index |Plane |Column|Row   |Strength|
|i     |i     |i     |i     |d       |
     1      1      1      1  1002.22
     2      1      2      1  1003.01
     3      1      3      1  1004.45
     4      1      4      1  1004.25
     5      1      5      1  1003.21
     6      1      6      1  1006.17
     7      1      7      1  1006.82
     8      1      8      1  1004.32
                ~
                ~
                ~
```

6.1.3 LATIMDETECT Log-File Output

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

```
Program LATIMDETECT, Version 4.0
Namelist File  = latimdetect_README.nl
```

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

```
Input History Image = ./testing/dntoflux_bcd_slope1.fits
Input Uncertainty of Target Image = ./testing/dntoflux_uncert_bcd_slope2.fits
Input PMask Image = ./testing/pmask.fits
Unusable PMask Bits = 16896
Input DMask History Image = ./testing/bmask_slope1.fits
Unusable DMask Bits = 25088
Input DMask Target Image = ./testing/bmask_slope2.fits
Latency Model Coeff A = 5.000000
Latency Model Coeff B = 1.000000
Latent Signal-to-Noise Ratio = 3.000000
Data-Plane Flag = 2
Ancillary Data-File Path = ./
Verbose flag = 0
Super-verbose flag = 0
Debug flag = 0
Number of latent suspects in target image (plane 1) from above history image =
11124
Fraction of latent pixels (above noise/bckgnd) on array = 0.678955
Performed latent-pixel detection.
Program latimdetect: Status Message: 0x0000
Normal exit from Function 0x0000: LOG_WRITER
latimdetect Probs Warning: NaN's were produced in the results.
A total of 1708 NaN's were produced in the results.
Processing time: 0.070000 seconds
Current date/time: Thu Dec  5 17:08:12 2002
Program LATIMDETECT, version 4.0, terminated.
```

6.2. LATIMDETECT Output Tutorial

An online tutorial which lists all command-line options can be generated by typing “latimdetect” on the command line with no arguments. This tutorial will indicate which parameters are required and those which are optional. Optional parameters are assigned the default values defined in Table 1. If any of the required parameters are not specified, or are unacceptably out of range, the program will abort with a message sent to standard output. Below is the latimdetect tutorial screen.

```
Program LATIMDETECT, Version 4.0

Usage: latimdetect
   -n   <input_namelist_fname> (Optional)
   -i   <input_hist_image_fname> (Required)
   -iu1 <input_hist_uncert_image_fname> (Optional; specify with -o2 below)
   -ip  <input_hist_pmask_image_fname> (Optional)
   -id1 <input_hist_dmask_image_fname> (Optional)
```

```
-id2 <input_target_dmask_image_fname> (Required)
-iu2 <input_target_uncert_image_fname> (Required)
-o1  <output_latent_image_fname> (Optional)
-o2  <output_latent_uncert_image_fname> (Optional; specify with -iu1 above)
-o3  <output_latent_table_fname> (Optional)
-l   <log_fname> (Default is stdout)
-fp  <unusable_pmask_bits> (Optional, Default = 0)
-fd  <unusable_dmask_bits> (Optional, Default = 0)
-p   <data_plane_to_process> (1=All, 2=First, 3=Last; Default = 1)
-A   <A_coefficient_of_model> (Optional, Default = 2.0)
-B   <B_coefficient_of_model> (Optional, Default = 1.0)
-S   <Latent_Signal_To_Noise_Ratio> (Optional, Default = 1.0)
-a   <ancillary_file_path> (Optional, Default = ./)
-d   (prints debug statements)
-v   (verbose output)
-vv  (superverbose output)
```

7. Testing

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

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

1. Executed LATIMDETECT 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 LATIMDETECT with input MIPS-24 laboratory stimulator images.

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

4. Executed LATIMDETECT on a non-square, large (COSMIC) image.

5. Executed LATIMDETECT with various combinations of latency model parameters and checked that latent strengths were correctly computed and latents flagged appropriately.

8. Usage Examples

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

```
LATIMDETECT –n latimdetect.nl -v | & tee out.log
```

Without using a namelist file:

```
LATIMDETECT -i input_hist_img.fits –iu1 input_hist_uncert.fits
–ip pmask.fits –id1 dmask_hist.fits –id2 dmask_targ.fits –iu2
input_targ_uncert.fits –o1 latent_strengths.fits –o2
latent_uncerts.fits –o3 latent_strengths.tbl –fp 16896 –fd
25088 –p 2 –A 5 –B 1 –S 3 –v –a ancpath
```

9. Glossary

BCD	Basic Calibrated Data
DCE	Data Collection Event
DN	Data Number
IOC	In-Orbit Checkout
SDS	Subsystem Design Specification
SIS	Software Interface Specification
SOS	Software Operations System
TBD	To Be Determined
TBR	To Be Resolved

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