674-SO-43, Version 1.0,
SSC-PT-4006

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: FIFTRANS

24 November 2004

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:

Dave Shupe

Approved by:

Mehrdad Moshir

Concurred by:

Suzanne Dodd

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	November 24, 2004

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.

3 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 FIFTRANS program reads as input a "master" mosaic Fiducial Image Frame (FIF), a sub-mosaic FIF (e.g., for a scan-leg contained therein), a "Cartesian shifts" file describing translational and rotational offsets of the sub-FIF and a list of input FITS images comprising the sub-FIF. The program then computes translational and rotational offsets for the input list of images in the master FIF. The output is a table in IPAC format to be read by the *pointingrefine* module downstream (see SDS SSC-PD-4065).

The main use of FIFTRANS in SSC operations is to produce a refinement offset table for BCDs comprising a MIPS scan-leg. The scan-leg mosaics have had their pointings previously refined in an upstream of the *pointingrefine* program. FIFTRANS transfers the refined pointing solution from the scan-leg frames to the BCD frames.

2.1. FIFTRANS Requirements

FIFTRANS is initiated by a startup script under the control of the pipeline executive and does its required functions for a given starting and ending SCLK time; 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 Fiducial Frame Tables (FIFs), FITS image list and a sub-FIF Cartesian shifts file.

C.) Produce as primary output a table in IPAC format containing image pointing refinement offsets and uncertainty information.

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 FIFTRANS program of the AOT PRODUCTS Subsystems.

A.) The SOS Downlink Requirements Document

B.) The SOS Requirements Document

C.) The SOS Downlink Software Development Guidelines

D.) The *pointingrefine* Subsystem Design Specification:

674-SO-43 SSC-PD-4065

E.) The following Software Interface Specifications (SIS)

SOSDL-SIS-PD-3017 (Fiducial Image Frame (FIF) table format)

2.3. Version History

2.3.1. Version 1.0

Initial version created on June 20, 2004

2.4. Liens

No liens have been identified.

3. Input

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

3.1. FIFTRANS Input

FIFTRANS takes all of its input from the command line which is set up by the startup script. This is controlled by the pipeline executive or executed standalone. The command-line parameters that can be defined for FIFTRANS are listed in Table 1. Command-line option flags are defined in Table 2.

Namelist variable	Description	Dim.	Type	Units	Default
NONE	Required input FITS image list filename comprising sub-FIF.	256	Char	-	Null
NONE	Required input Cartesian shifts filename for sub-FIF mosaic.	256	Char	-	Null
NONE	Required input “master” mosaic FIF filename.	256	Char	-	Null
NONE	Required input sub-mosaic (sub-) FIF filename	256	Char	-	Null
NONE	Required line number in Carteisan Shifts file corresponding to sub-FIF.	1	I*2	-	Null
NONE	Optional ancillary path containing supporting files	256	Char	-	./
NONE	Optional output log filename	256	Char	-	stdout

Table 1: Input Parameters for fiftrans

Command-line option	Value Name
-i	Input image list filename

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

-s	CartesianShifts filename
-f1	Master mosaic FIF filename
-f2	Sub-mosaic FIF filename
-n	Line number in CartesianShifts
-a	Ancillary file path
-l	Output log file
-v	Verbosity switch

Table 2. Command-line options

4. Processing

4.1. FIFTRANS Processing

FIFTRANS begins processing by writing its name and version number to standard output, checks that the required command-line parameters were passed to it and that the required environment variables were set. If this condition is not true, it writes a message stating which parameters are missing, recommends a look at this document, and terminates by issuing an appropriate exit code to the pipeline executive; otherwise it proceeds as follows.

If an error occurs during processing, 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 to standard output.

4.2 FIFTRANS Processing Phases

FIFTRANS operates in seven phases: initialization, read input FIF and Cartesian shift tables, storage of WCS information of input FITS images and FIFs, transfer pointings of input FITS images and sub-FIF to master mosaic FIF x,y coordinates, compute input image shifts and uncertainties, output results and termination. This processing level is depicted in Figure 1.

4.1.1. FIFTRANS Initialization

FIFTRANS 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.) The command-line inputs (see Tables 1 and 2) are read and checked for correct data range, consistency, etc. If any errors are encountered, a message is printed, and execution aborts.

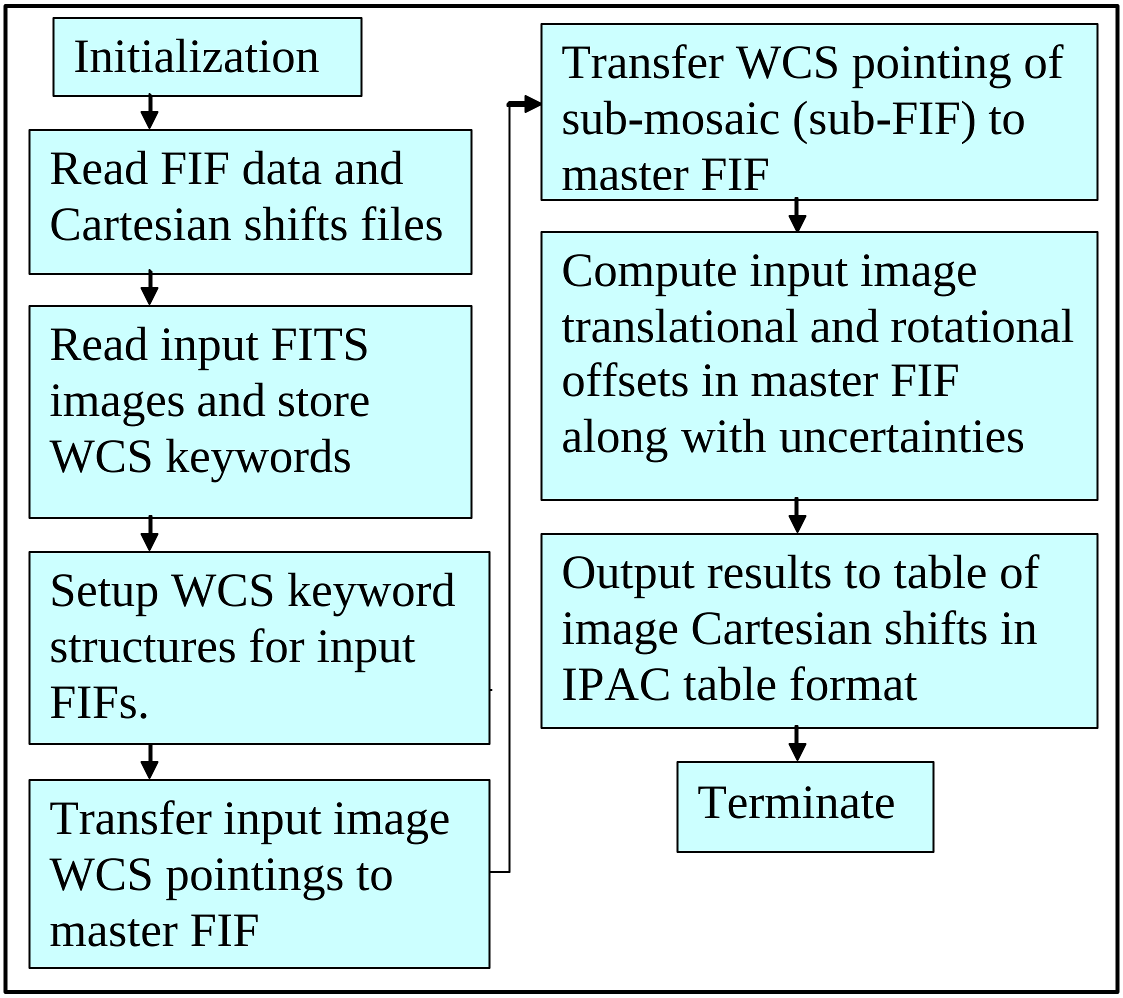

Figure 1. FIFTRANS data and processing flow

4.1.2. Input FIF and Cartesian Tables

The input master FIF and sub-FIF are read using the *tblio* IPAC table library. An example of an input FIF is as follows.

```
\char comment = Output from fiducial_image_frame, version 1.2
\char Date-Time = Nov 08, 2004, 18:19:45
\real CRVAL1 = 259.3088
\real CRVAL2 = 59.5402
```

```
\real CRPIX1 = 75.50
\real CRPIX2 = 642.50
\real CROTA2 = -94.96859
\real CDELT1 = -6.92764E-04
\real CDELT2 = 7.21916E-04
\int  NAXIS1 = 150
\int  NAXIS2 = 1284
\char PROJTYPE = TAN
\real EXTENT_X = 0.103915
\real EXTENT_Y = 0.926940
```

The Cartesian shifts file corresponding to all the sub-mosaic (e.g, scan-leg mosaics for MIPS) within the master FIF is also read in. In operations, this is produced upstream from an initial run of the pointing refinement program (*pointingrefine*). An example of an input Cartesian shifts file is as follows:

```
\character Cartesian_Tangent_Shifts_File
\character From_Pointing_Refinement_Program = "pointingrefine", Version  6.50
\character Creation_Date_Time = Mon Nov  8 19:14:46 2004
\character Input_Image_List = LegMosaicImgList.txt
\character Reference_Image (shifts are measured in) = image defined by fiducial frame
table (MosaicPntgRefOut/FIF.tbl)
|Img |Theta     |XT        |YT       |Err_Theta |ErrXT     |ErrYT     |NASTROM |
|int |real      |real      |real     |real      |real      |real      |int     |
|null|degree    |pixel     |pixel    |pixel     |pixel     |pixel     |null    |
   1  -0.040854  -0.075370   0.241726   0.051169   0.257887   0.212408   5
   2   0.095043   0.255239   0.086131   0.147272   0.380593   0.370002   3
   3   0.035289   0.326808   0.144751   0.044205   0.292910   0.270869   4
   4  -0.048266  -0.190643  -0.429042   0.098548   0.780356   0.532529   4
   5  -0.212639  -1.308910   0.641431   0.566823   4.486370   0.866661   2
   6  -0.064053   0.307830  -0.978883   0.080495   0.722530   0.462224   3
   7  -0.085376   0.498007   0.777938   0.252546   1.145473   0.495705   3
   8   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0
   9  -0.104673   0.342974   0.732437   0.243284   0.932361   0.455609   2
  10   0.000000   0.000000   0.000000   0.000000   0.000000   0.000000   0
```

4.1.3. Storage of WCS Information

WCS pointing information from all input FITS image headers are read using the *fitsrhead* routine from the *wcs* library (January 2001, Doug Mink, SAO). The WCS information is stored in an array of structures *wcsin*[i] (for each FITS image) using the *wcsinit*(*header*) routine. The WCS information for the input FIFs are also stored in WCS structures. For these, the *wcskinit* routine from the *wcs* library is used.

4.1.4. Transfer of WCS Info. to Master FIF

The WCS pointing (CRVAL1, CRVAL2 keywords) are transferred to *x, y* representations in the master mosaic FIF for all input FITS images and the input sub-FIF. The *wcs2pix* routine from the *wcs* library is used.

4.1.5 Image Offset Computations and Uncertainties

Translational and rotational offsets (and accompanying uncertainties) are computed in the master FIF for all input images using the input sub-FIF Cartesian offsets. Given coordinates of the rotational center for the sub-FIF mosaic (X_A, Y_A) in the master frame and translational/rotational offsets for the sub-FIF: (dX_A, dY_A, d?$_A$), the "new" master FIF coordinates of an input image (subscript "*Bn*") in the sub-FIF are given by the following transformation

$$\begin{pmatrix} X_{Bo} \\ Y_{Bo} \end{pmatrix} \rightarrow \begin{pmatrix} X_{Bn} \\ Y_{Bn} \end{pmatrix} = \begin{pmatrix} X_A \\ Y_A \end{pmatrix} + \begin{pmatrix} \cos dq_A & -\sin dq_A \\ \sin dq_A & \cos dq_A \end{pmatrix} \begin{pmatrix} X_{Bo} - X_A \\ Y_{Bo} - Y_A \end{pmatrix} + \begin{pmatrix} dX_A \\ dY_A \end{pmatrix}, \qquad (1)$$

where the coordinates with subscript "*Bo*" are the original positions of an input image in the master FIF, obtained from the WCS transformation described in Section 4.1.4. The angle δq_A is measured in the counter-clockwise sense, and the (dX_A, dY_A, d?$_A$) are read from the input Cartesian shifts file for the corresponding input sub-FIF. The translational offsets of the input images (relative to their original positons) are then given by:

$$\begin{pmatrix} dX_B \\ dY_B \end{pmatrix} = \begin{pmatrix} X_{Bn} - X_{Bo} \\ Y_{Bo} - Y_{Bo} \end{pmatrix}. \qquad (2)$$

Since the input images are assumed to be rigidly fixed in the sub-FIF, the best value we can assume for the rotational offsets are $\delta q_B = 0$.

Uncertainties in the offsets (dX_B, dY_B, d?$_B$) are computed using standard error propagation on equations (1) with the assumption δq_A ˜ 0. This assumption has been found to be valid for *Spitzer* data where the rotational offsets reflect observatory pointing/position angle uncertainties of typically 20 arcsec. Equations (1) can therefore be linearized in δq_A by assuming

$$dq_A \approx 0 \;\; \Rightarrow \;\; \sin dq_A \approx dq_A \text{ and } \cos dq_A \approx 1 \; . \qquad (3)$$

With this assumption, equations (1) and (2) reduce to:

$$dX_B = -(Y_{Bo} - Y_A)dq_A \ + dX_A$$
$$dY_B = \ (X_{Bo} - X_A)dq_A \ + dY_A \qquad (4)$$

From equations (4), uncertainties in the translational offsets of an input image in the master FIF are given by:

$$s\,(dX_B) = \sqrt{(Y_{Bo} - Y_A)^2 s^2(dq_A) \ + s^2(dX_A)}$$
$$s\,(dY_B) = \sqrt{(X_{Bo} - X_A)^2 s^2(dq_A) \ + s^2(dY_A)}\,, \qquad (5)$$

where the uncertainties s(dX_A), s(dY_A) and s(d?$_A$) are read from the input Cartesian shifts file for the corresponding input sub-FIF. The accuracy to which we can estimate the rotational offset of an input image is as good as that of the sub-FIF itself, therefore we assume:

$$s\,(dq_B) = s\,(dq_A)\,.$$

4.1.6 Output Results

Translational and rotational offsets for all input images are written to a table file in IPAC format. This is in the same format as the input example given in Section 4.1.2 for the sub-FIF. The table filename is constructed by appending the appropriate line number in the input Cartesian shifts file (specified by command-line option "-n") to the input Cartesian Shifts filename itself (option "-s"). i.e.,

Output filename = <CartesianShiftsFilename>_n

4.1.7 Termination

Summary output is written both to standard output and to the header of the output IPAC table. The program also issues an appropriate exit code to the system to be picked up by the pipeline executive.

5. Tutorial

If the FIFTRANS program is executed on the command-line with no arguments, the following tutorial will be printed to standard output.

```
Program fiftrans, Version 1.0

Usage: fiftrans
   -i  <inp_image_list_fname>              (Required)
   -s  <inp_leg_Cartesianshifts_fname>     (Required)
   -f1 <inp_mosaic_fif_fname>              (Required)
   -f2 <inp_leg_fif_fname>                 (Required)
   -n  <line(leg)_num_in_Cartesianshifts>  (Required)
   -a  <ancillary_file_path>               (Optional, Default = ./)
   -l  <log_fname>                         (Optional, Default = 'stdout')
   -v  (verbose output)

   Output will be Cartesianshifts file for input BCDs, named
   <inp_leg_Cartesianshifts_fname>_n, in same order as input file list.
```

6. Output

FIFTRANS is capable of generating the following output:

A.) Standard-output processing and status messages.

B.) An output table file in IPAC format containing diagnostic information in the header and results of the computations.

All FIFTRANS disk output is written to the pathname that is specified with the input Cartesian shifts filename in the command-line (option “-s”).

7. Testing

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

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

1. Executed FIFTRANS with inputs read from and output written to directories different from where the program was run.
2. Executed FIFTRANS using an input image list consisting of 5000 entries.
3. Executed FIFTRANS with different input values and filenames to ensure error checking and reporting worked as intended.

8. Usage Example

Executed with input files in a subdirectory called "./testing", with the verbose flag turned on ("-v" option) and with shifts read from the first line ("-n 1") of the input sub-FIF Cartesian shifts file specified by the "-s" option:

```
fiftrans -i ./testing/ImageList_0.txt -s
./testing/cartesianShifts.tbl
         -f1 ./testing/FIFmosaic.tbl -f2 ./testing/FIF_0.tbl -n 1 –v
```

This will write results to an output file called: "`./testing/cartesianShifts.tbl_1`". With no output log filename specifed on input, the log is written to standard output (by default). The following is an example of the log output from the above run:

```
Program fiftrans, Version 1.0
Input image list file = ./testing/ImageList_0.txt
Cartesian shifts table for leg mosaics = ./testing/cartesianShifts.tbl
Mosaic FIF table name = ./testing/FIFmosaic.tbl
Leg-Mosaic FIF table name = ./testing/FIF_0.tbl
Output image Cartesianshifts filename = ./testing/cartesianShifts.tbl_1
Line number (leg shifts) used = 1
Ancillary Data-File Path = .
Verbose flag = 0
Performed leg-mosaic to image shift translation in ./testing/FIFmosaic.tbl fiducial
frame.
Program fiftrans: Status Message: 0x0000
Normal exit from Function 0x0000: LOG_WRITER
Processing time: 1.260000 seconds
Current date/time: Wed Nov 24 17:05:32 2004
Program fiftrans, version 1.0, terminated successfully.
```

9. Glossary

BPHF	Boresight Pointing History File
DCE	Data Collection Event
DN	Data Number
FIF	Fiducial Image Frame
IOC	In-Orbit Checkout
IPAC	Infrared Processing and Analysis Center
MIPS	Multi-band Imaging Photometer for *Spitzer*
PH	Pointing History
SCET	Spacecraft Ephemeris Time
SCLK	Spacecraft Clock
SDM	Science Data Management team at SSC
SDS	Subsystem Design Specification
SIS	Software Interface Specification
SODB	Science Operations Database
SSC	*Spitzer* Science Center
TBD	To Be Determined
TBR	To Be Resolved
WCS	World Coordinate System