674-SO-43, Version 1.0,
SSC-PT-4002

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: MULTIBANDREFINE

26 August 2002

National Aeronautics and
Space Administration

California Institute of Technology
SIRTF Science Center

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:

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	August 26, 2002

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

2 Table of Contents

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

List of Figures

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

List of Tables

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

1. Introduction

1.1. Purpose and Scope

The Subsystem Design Specification is a document that describes the basic requirements, assumptions, definitions, software-design details and necessary interfaces for each subsystem. The document will be used to trace the incremental development of each subsystem and also to allow trace-back of levied requirements; this document should have sufficient detail to allow future modification or maintenance of the software by developers other than the original developers. This document is an evolving document as changes may occur in the course of science instrument hardware design and maturity of operational procedures. This document is not intended to repeat sections or chapters from other Project documents; when appropriate, references to proper sections of primary reference documents will be made.

1.2. Document Organization

This document is organized along the major themes of Requirements; Assumptions; Operational Concept; Functional Descriptions; Functional Dependencies; Input; Output; Other S/S Interfaces; Algorithm Descriptions (when applicable); and Major Liens.

The material contained in this document 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.

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

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 MULTIBANDREFINE software uses output computed by the POINTINGREFINE software to refine the pointings of a set of images corresponding to separate wavelength bands acquired simultaneously in a single exposure. An example is the IRAC instrument on board SIRTF consisting of four focal plane arrays which acquire images of the sky ate four wavelengths. The software is generic enough that anywhere from one to a maximum number of four band-specific FITS image lists can be specified on input.

Input to the MULTIBANDREFINE software are Cartesian offsets in the mosaic fiducial image frame computed by the POINTINGREFINE software for one or two lists of band-specific images. If two (band-specific) offset files are specified, average offsets (with inverse-variance weighting) are computed and the results used to refine the pointings and orientations of images in all band-specific image lists specified on input. Only absolute refinement of images (in the ICRS) is supported. Input images *do not* have to be in any specific order in the input band-specific lists, *neither* do the lists have to be of uniform length. Sorting is done via the EXPID FITS keyword which is expected to be unique across all bands of a single sky exposure.

The software uses routines from the standard World Coordinate System Library (WCS) (Doug Mink, 2001, SAO) for pixel to sky coordinate conversions. All standard types of WCS map-projections are supported. The primary outputs from MULTIBANDREFINE are new FITS header keywords giving the refined pointing and twist angle; optionally, tables in IPAC format listing refined pointings and uncertainty information for each band-specific image list, and, a QA file storing diagnostic information. MULTIBANDREFINE is written in ANSI/ISO C.

2.1. MULTIBANDREFINE Requirements

MULTIBANDREFINE 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 (up to four) lists of standard FITS images, lists of Cartesian offset files (up to two) and fiducial frame table parameters.

C.) Produce as primary output: updated pointing information in the FITS headers and IPAC tables for each band-specific image list.

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 either to standard output or a log file.

2.2. Applicable Documents

The following documents are relevant to the MULTIBANDREFINE 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 POINTINGREFINE SDS Document, latest version:

sds-s6.3-pointingrefine.doc

2.3. Version History

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

2.3.1. Version 1.0

Initial version created on August 26, 2002.

2.4. Liens

One (albeit minor) lien has been identified:

- The uncertainty in *refined twist angle* is assumed to be equal to the uncertainty in rotational offset computed in the reference image frame. This approximation is expected to hold only if one is far enough away from the poles (say $|\delta| < 50°$ to be exact). Close to the poles, uncertainties in RA are expected to be strongly correlated with uncertainties in twist angle since a small shift in RA near a pole implies a large change in twist angle. A robust computation of the twist angle uncertainty will require full error-propagation of Equation (9).

3. Input

3.1. MULTIBANDREFINE Input

MULTIBANDREFINE 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. MULTIBANDREFINE NAMELIST Input

MULTIBANDREFINE reads the NAMELIST file whose name is passed to it by start-up script. The name of the NAMELIST is MULTIBANDREFIN. 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_ch1	Required filename containing list of band-1 FITS-images.	256	C	(redundant)	Null
FITS_Image_List_Filename_ch2	Optional filename containing list of band-2 FITS-images. Required if "-s2" specified.	256	C	(redundant)	Null
FITS_Image_List_Filename_ch3	Optional filename containing list of band-3 FITS-images.	256	C	(redundant)	Null
FITS_Image_List_Filename_ch4	Optional filename containing list of band-4 FITS-images.	256	C	(redundant)	Null
Fiducial_Frame_Table	Required filename in IPAC format specifying fiducial image frame WCS parameters.	256	C	(redundant)	Null
X_Y_Theta_Shifts_chn1	Required filename of list of Cartesian shifts in fiducial image frame for band-1 images in format output by "pointingrefine" program.	256	C	(redundant)	Null
X_Y_Theta_Shifts_chn2	Optional filename of list of Cartesian shifts in fiducial image frame for band-1 images in format output by "pointingrefine" program.	256	C	(redundant)	Null
Data_Out_Filename1	Optional output table filename of refined pointings and uncertainties for band-1 images.	256	C	(redundant)	Null

Data_Out_Filename2	Optional output table filename of refined pointings and uncertainties for band-2 images.	256	C	(redundant)	Null
Data_Out_Filename3	Optional output table filename of refined pointings and uncertainties for band-3 images.	256	C	(redundant)	Null
Data_Out_Filename4	Optional output table filename of refined pointings and uncertainties for band-4 images.	256	C	(redundant)	Null
Log_Filename	Optional output log filename	256	C	(redundant)	stdout
Ancillary_File_Path	Pathname where supporting source files are installed.	256	C	(redundant)	./ (current directory)

Table 1. Namelist file

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

```
  &MULTIBANDREFIN
  Comment = 'Generic namelist file for multibandrefine',
  Ancillary_File_Path =           '../multibandrefine_v1',
  FITS_Image_List_Filename_chn1 = './testing/fitsimglist_ch1.txt',
  FITS_Image_List_Filename_chn2 = './testing/fitsimglist_ch2.txt',
  FITS_Image_List_Filename_chn3 = './testing/fitsimglist_ch3.txt',
  FITS_Image_List_Filename_chn4 = './testing/fitsimglist_ch4.txt',
  X_Y_Theta_Shifts_chn1 =         './testing/shifts_ch1.dat',
  X_Y_Theta_Shifts_chn2 =         './testing/shifts_ch2.dat',
  Fiducial_Frame_Table =          './testing/fiducial_800.tbl',
  Data_Out_Filename_chn1 =        './testing/refined1.tbl',
  Data_Out_Filename_chn2 =        './testing/refined2.tbl',
  Data_Out_Filename_chn3 =        './testing/refined3.tbl',
  Data_Out_Filename_chn4 =        './testing/refined4.tbl',
  Log_Filename =                  'stdout',
 &END
```

3.1.2. MULTIBANDREFINE 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. MULTIBANDREFINE FITS Input

MULTIBANDREFINE 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, and fits_close_file.

Command-line option	**Variable**
-n (namelist filename)	-
-f1	FITS_Image_List_Filename_chn1
-f2	FITS_Image_List_Filename_chn2
-f3	FITS_Image_List_Filename_chn3
-f4	FITS_Image_List_Filename_chn4
-fi	Fiducial_Frame_Table
-s1	X_Y_Theta_Shifts_chn1
-s2	X_Y_Theta_Shifts_chn2
-o1	Data_Out_Filename_chn1

-o2	Data_Out_Filename_chn2
-o3	Data_Out_Filename_chn3
-o4	Data_Out_Filename_chn4
-l	Log_Filename
-a	Ancillary_File_Path
-qa (QA switch – generates "QAlogfile.txt"	-
-v (verbose switch)	-
-vv (super-verbose switch)	-
-d (debug switch)	-

Table 2. Command-line options

4. Processing

4.1. MULTIBANDREFINE Processing

MULTIBANDREFINE 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, 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 all input parameters, date and time, processing time, and a termination-status code are written to a log file (specified by the –l command-line option).

4.2. MULTIBANDREFINE Processing Phases

MULTIBANDREFINE operates in nine phases: initialization, read in and sort input offset values and input image lists, compute weighted average of band-1 and band-2 offsets, assign final offset values to all band-specific image lists, transform observed-image pointings to fiducial image frame, use Cartesian offsets to refine celestial pointings and twists, compute uncertainties in refined pointings, output results generation, and termination. This processing level is depicted in Figure 1.

4.2.1. MULTIBANDREFINE Initialization

MULTIBANDREFINE 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.

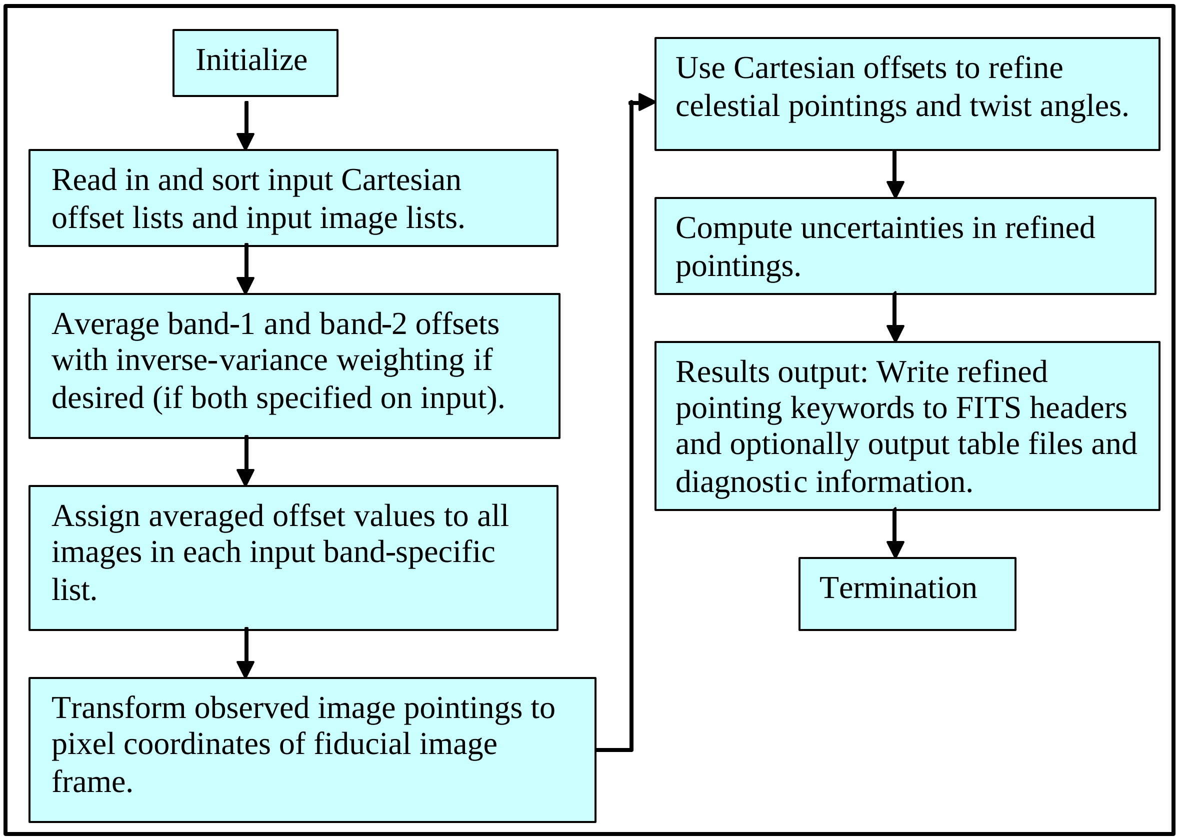

Figure 1. MULTIBANDREFINE data and processing flow

4.2.2. Read/Sort Input Offset and FITS Image Lists

Up to two Cartesian offset lists as generated by the "pointingrefine" software can be specified on input (namelist parameters X_Y_Theta_Shifts_chn1 and X_Y_Theta_Shifts_chn2). The offsets in these lists must correspond *in the same order* to images listed in the input FITS image lists for band-1 and band-2 respectively. One offset list (corresponding to the band-1 input image list) is the minimum input required. The input offset list which has the maximum number of entries is used to establish the "base" array prior to averaging the two-band offsets. This maximizes the amount of information to propagate downstream when refining images in other bands.

FITS images specified in all available input lists are read and values of their header keyword "EXPID" are stored in memory. If both offset lists are specified for band-1 and band-2 images, one dimensional arrays of EXPID values and matching offset values are made for the averaging stage.

4.2.3. Weighted Averaging of Band-1 and Band-2 Offsets

Averaging of band-1 and band-2 offsets can only be performed if both offset files are specified on input. Otherwise, only band-1 image offsets (required minimum input) will be used in the pointing refinement stage. Below is an example of an offset text file where the column names are NOT included in the pointingrefine output.

```
Img# |theta     |X_shift   |Y_shift   |Err_theta  |Err_X     |Err_Y     |NASTROM
     |deg       |ref_pix   |ref_pix   |deg        |ref_pix   |ref_pix
   1  -0.026478  -1.233473   0.850536   0.102698   0.156743   0.254244   1
   2   0.068312  -1.110399   0.153084   0.104464   0.127155   0.181946   0
   3  -0.094970   0.418239   0.330770   0.104017   0.132257   0.372433   3
   4   0.011755  -0.395867   0.897485   0.101616   0.146039   0.305199   0
   5  -0.023190   0.620723   0.590670   0.101570   0.125114   0.216150   0
   6   0.021597   0.141953   0.215126   0.090561   0.172765   0.165146   4
   7  -0.004009  -0.733558  -0.750462   0.103518   0.140041   0.348311   0
   8   0.033793   0.611526   0.861340   0.102333   0.116684   0.388635   3
```

Since the band-1 and band-2 image lists can be of different length (due to missing packets in the downlink process) with images not necessarily in the same order, averaging of offsets from each band-dependent offset list is performed by ensuring that the EXPID keyword values from the image lists match. When a matching band-1, band-2 pair is identified, an average offset (with inverse variance weighting) is computed. For any two offsets ΔX_1 , ΔX_2 (either theta, X_shift or Y_shift), the average is given by:

$$\left\langle \Delta X \right\rangle = \frac{w_1 \Delta X_1 + w_2 \Delta X_2}{w_1 + w_2} , \qquad (1)$$

where w_1 and w_2 are inverse-variance weights:

$$w_1 = \frac{1}{s^2_{\Delta X 1}} \quad , \quad w_2 = \frac{1}{s^2_{\Delta X 2}} \qquad (2)$$

The uncertainty in the average value (Equation 1) is given by:

$$s\left\langle \Delta X \right\rangle = \frac{1}{\sqrt{w_1 + w_2}} \tag{3}$$

It is important to note that if either offset ΔX_1 or ΔX_2 is *zero*, this means no solution existed from the "global-minimization" computation in the pointingrefine software. If this is found, then the *non-zero* offset value is stored in the "final" post-averaged offset array. If offsets from both bands are *zero*, then the value zero is stored in the final array.

4.2.4. Assigning Offsets to All Band-Dependent Images

Once a final array of (averaged) offset values and corresponding EXPID's to which they apply is established, each FITS image in every band-specific input list is assigned a Cartesian offset value according to its EXPID keyword value. Again, the image lists can be of different size and the image files in a different order so that recursive matching between EXPID values assigned to final offset values and actual image EXPID's is necessary.

4.2.5. Transformation of Observed Image Pointings

The observed RA, Dec values as specified by FITS keywords CRVAL1, CRVAL2 in the headers of all images (from any band-specific list) are transformed to pixel coordinates of the fiducial image frame. These will later be adjusted using the Cartesian offset values and re-transformed back to the sky to yield refined pointings (see below). An example of a fiducial image frame table (namelist parameter Fiducial_Frame_Table) as generated by the "fiducial_image_frame" software is shown below.

```
\char comment = Output from fiducial_image_frame, version 1.0
\char Date-Time = Jul 22, 2002, 10:20:30
\real CRVAL1 = 159.819
\real CRVAL2 = 59.429
\real CRPIX1 = 810.5
\real CRPIX2 = 810.5
\real CROTA2 = 0.0
\real CDELT1 = -3.370319900569E-04
\real CDELT2 = 3.370319900569E-04
\int  NAXIS1 = 1620
```

```
\int  NAXIS2 = 1620
\char PROJTYPE = TAN
\real EXTENT_Y = 1.0
\real EXTENT_Z = 1.0
```

4.2.6. Refinement of Celestial Pointings

Given the final (band-1–band-2 averaged) offsets of an image m (δq^m, δX^m, δY^m), we correct the tangent points (usually image centers in reference image coordinates – i.e. $x^m{}_c$, $y^m{}_c$ in figure 2 below) corresponding to CRVAL1 and CRVAL2 (RA, DEC). This can be done using the original Cartesian transformation equations (Equations 5 and 6 in the "pointingrefine" SDS):

$$x_i^m \rightarrow \tilde{x}_i^n = x_i^m - (y_i^m - y_c^m)dq^m + dX^m \qquad (4)$$

$$y_i^m \rightarrow \tilde{y}_i^n = y_i^m + (x_i^m - x_c^m)dq^m + dY^m \quad , \qquad (5)$$

Since the rotation is about the centers, these transformations reduce to:

$$x_c^m(\text{new}) = x_c^m(\text{old}) + dX^m \qquad (6)$$

$$y_c^m(\text{new}) = y_c^m(\text{old}) + dY^m \qquad (7)$$

Using the WCS parameters of the fiducial reference image frame, these can be transformed back to the sky to yield *refined* CRVAL1 and CRVAL2 coordinates.

Refinement of the sky twist angle (CROTA2 keyword value) due to rotational offsets (and translational offsets if one is close to a pole) is a little more complicated. To compute the refined twist angle, we use a second point in an image located at coordinates (CRPIX1, NAXIS2) – or anywhere along a line joining this point and the center (CRPIX1, CRPIX2). See Figure 2 below. This is chosen because the <u>*angle*</u> between a vector extending from the center to this second point (solid red line in image m below) and lines of constant RA on the sky defines the twist angle measured east from north (see Figure 3). The coordinates of this second point in the reference image frame are corrected in the same way as the image centers, but using equations (3) and (4) with δq. This "corrected" second point is also transformed to the sky. These two RA, DEC points in an image can be used to compute the sky twist angle using spherical trigonometry (see Figure 3). The derivation is given below.

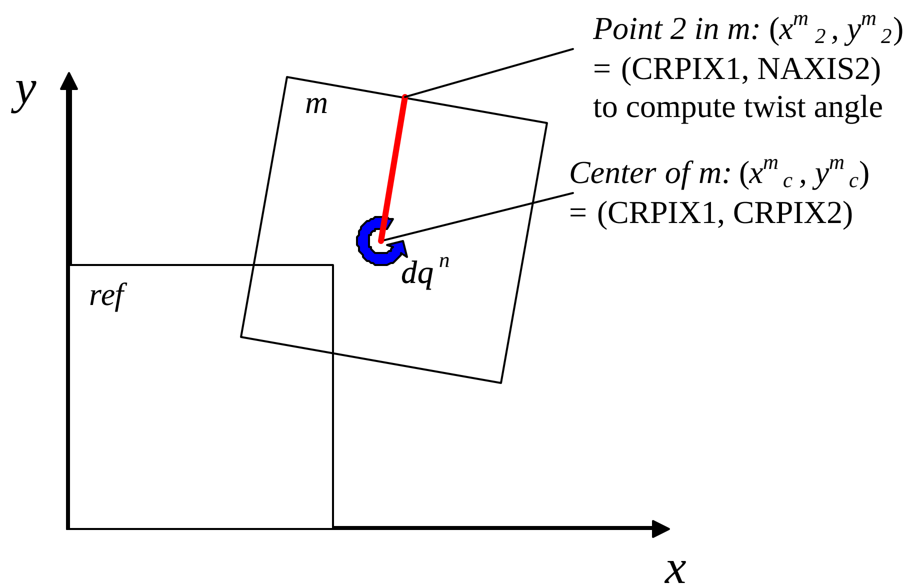

Figure 2. Schematic showing the two-points per image for computing the twist angle.

To compute the sky twist angle given these two (RA, DEC) points in an image, we shall make use of the schematic shown in Figure 4. Given points B and C on the sky (derived using the formalism above), the triangle ΔABC forms a spherical triangle with sides d_{AB} , d_{BC} and d_{AC} . The angle γ is our desired image twist angle (measured East from North or in the direction of increasing RA). Applying the "law of sines" to this spherical triangle leads to:

$$\frac{\sin g}{\sin\ d_{AB}} = \frac{\sin\left|a_c - a_2\right|}{\sin\ d_{BC}} \tag{8}$$

$$\Rightarrow\ g = \sin^{-1}\left[\frac{\sin\ d_{AB}}{\sin\ d_{BC}}\sin\left|a_c - a_2\right|\right]. \tag{9}$$

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

The distances d_{AB} and d_{BC} can be computed using the formula for the distance between two points along a great circle on a sphere:

$$d_{AB} = \cos^{-1}\left[\sin d_2\right] \qquad (10)$$

$$d_{BC} = \cos^{-1}\left[\sin d_2 \sin d_c + \cos d_2 \cos d_c \cos(a_2 - a_c)\right]. \qquad (11)$$

Care must be taken when computing γ from equation (9) since γ is usually defined to lie within $0 \le \gamma \le 360°$ and will need to be re-scaled for declinations $< \delta_c$ and whether $\alpha_2 < \alpha_c$ or $\alpha_2 > \alpha_c$.

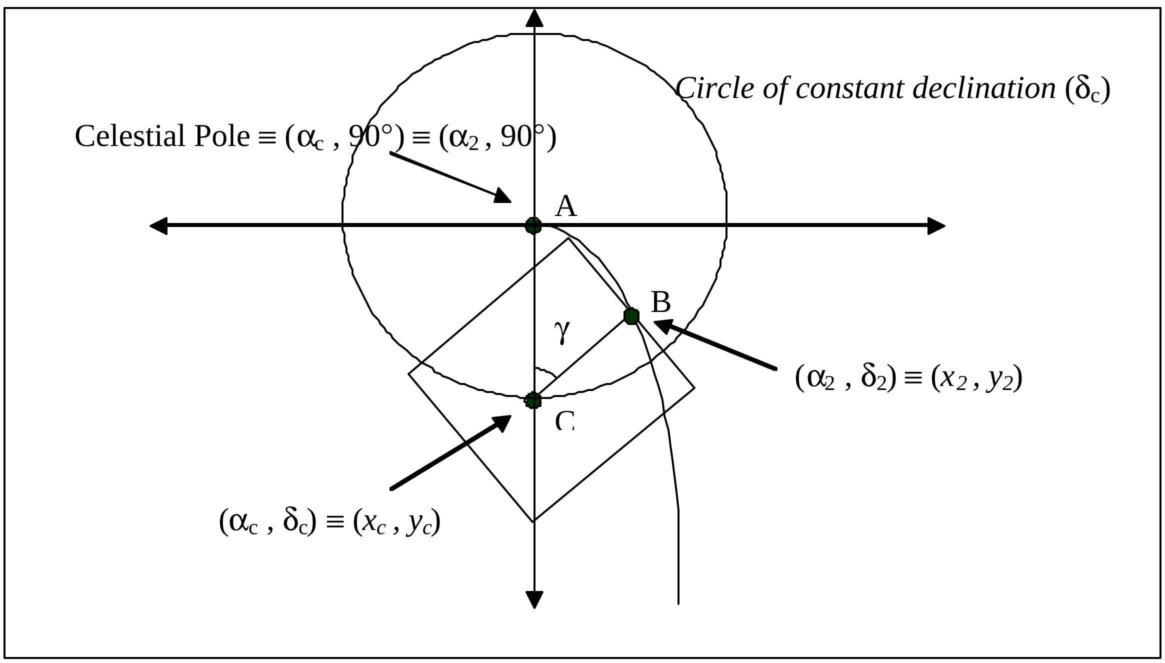

Figure 3. Schematic used in the derivation of the sky twist angle.

4.2.7. Uncertainties in Refined Pointings

The following is an algebraic representation of the process. Let (x_c, y_c) be the pixel coordinates of the "*refined*" pointing centers of an input image in the reference frame as computed from Equations (6) and (7). Then we transform the maximum and minimum extents to the sky:

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

$$\left.\begin{aligned} x_{\max} &= x_c + \Delta x_c \\ y_{\max} &= y_c + \Delta y_c \end{aligned}\right\} \xrightarrow{PIX2WCS} a_{\max}, d_{\max}$$

$$\left.\begin{aligned} x_{\min} &= x_c - \Delta x_c \\ y_{\min} &= y_c - \Delta y_c \end{aligned}\right\} \xrightarrow{PIX2WCS} a_{\min}, d_{\min} \tag{12}$$

Uncertainties in the refined pointing centers can then be computed as follows:

$$\Delta a_c = \frac{1}{2}\cos^{-1}\left[\sin^2 d_c + \cos^2 d_c \cos(a_{\max} - a_{\min})\right]$$

$$\Delta d_c = \frac{1}{2}\left|d_{\max} - d_{\min}\right| \ . \tag{13}$$

The uncertainty in *refined twist angle* is assumed to be equal to the uncertainty in rotational offset computed in the reference image frame. This approximation is expected to hold only if one is far enough away from the poles (say $|\delta| < 50°$ to be exact). Close to the poles, uncertainties in RA are expected to be strongly correlated with uncertainties in twist angle since a small shift in RA near a pole implies a large change in twist angle. A robust computation of the twist angle uncertainty will require full error-propagation of Equation (9). This remains a lien of the current software.

4.2.8. Outputs

There are four possible output products, three of which are optional. Items **(1)** and **(2)** below contain final results of the refinement. Items **(3)** and **(4)** contain ancillary and QA diagnostic information pertaining to processing.

(1). Input image headers are always updated with *new* refined pointing keywords and uncertainties. Every image header is updated with a new set of keywords regardless if a refined solution existed or not. For images where no refinement was possible, the input CRVAL1, CRVAL2, CROTA2 keywords are reproduced to maintain a consistent set of pointing keywords for use in downstream ensemble processing. The NASTROM keyword represents the number of absolute sources used in the refinement and is only present if the "pointingrefine" software was executed in "absolute-refinement" mode. The following is an example of the keywords written to each input header after both *pointingrefine* and *multibandrefine* have been run.

```
PRECOMPU= 'pointingrefine'      / Pre-computation software version 2.0
SOFTWARE= 'multibandrefine'     / Multi-band refinement (post-processing)
PTGVERSN=                    1. / Version number of multibandrefine program
```

```
RARFND  =     159.750508334032 / [deg] Refined RA pertaining to mosaic
DECRFND =     59.1925262022405 / [deg] Refined DEC pertaining to mosaic
CT2RFND =     359.955934030143 / [deg] Refined CROTA2 pertaining to mosaic
ERARFND =  5.2347187065671E-05 / [deg] Error in refined RA
EDECRFND= 6.12400442605576E-05 / [deg] Error in refined DEC
ECT2RFND=    0.269450110038946 / [deg] Error in refined CROTA2
NASTROM =                    3 / # Astrometric sources for absolute refinement
RARESID =     4.02709200560594 / [arcsec] Residual: Observed-Refined RA
DECRESID=     1.04484844627848 / [arcsec] Residual: Observed-Refined DEC
CT2RESID=      365.52606192954 / [arcsec] Residual: Observed-Refined CROTA2
```

(2). Optionally (if the "–o1", "–o2", "–o3" or "–o4" options are specified), tables in IPAC format for each image band listing FITS image names (with directory paths), refined RA, DEC, CROTA2 values and uncertainties can be generated. Ancillary information is included in the headers of these tables. An example is shown below.

```
\character Pointing_Refinement_Program = "multibandrefine", Version  1.00
\character Creation_Date_Time = Sun Aug 25 16:32:41 2002
\character Input_Image_List = ./testing/fitsimglist_ch1.txt
\character Input_Fiducial_Frame_Table = ./testing/fiducial_800.tbl
\character Input_Cartesian_Shifts_Filename (band 1) = ./testing/shifts_ch1.dat
\character Input_Cartesian_Shifts_Filename (band 2) = ./testing/shifts_ch2.dat
\character Reference_Image (pointings are relative to) = image defined by fiducial frame table
(./testing/fiducial_800.tbl)
\character RA  = Refined right ascension of CRPIX1,CRPIX2
\character DEC = Refined declination of CRPIX1,CRPIX2
\character CROTA2 = Refined twist angle measured East from North on sky
\character sigma_RA  = Uncertainty in refined right ascension
\character sigma_DEC = Uncertainty in refined declination
\character sigma_CROTA2 = Uncertainty in refined twist angle
\integer   Number_of_Frames = 6
|Index |Filename       |RA         |DEC        |CROTA2     |sigma_RA   |sigma_DEC
|sigma_CROTA2|
|int   |char           |real       |real       |real       |real       |real       |real
|
|null  |null           |degree     |degree     |degree     |degree     |degree     |degree
```

(3). Ancillary information on the fraction of input images actually refined, fractional differences between band-1 and band-2 Cartesian shifts (see Equation 14), QA indices for downstream analysis and residuals between input and refined pointings can be written to a generic log file "QAlogfile.txt" if the "–qa" switch is specified on the command-line (see example below). Additional information is also written to standard output by specifying the verbose "–v" switch. The fractional (%) difference between band-1 and band-2 Cartesian shifts is defined:

$$f = 100 \left| \frac{\Delta X_1 - \Delta X_2}{\Delta X_1 + \Delta X_2} \right| \tag{14}$$

```
AveFIF_Theta_Diff   0.00 ## Average % difference in band1-band2 Theta shifts
MaxFIF_Theta_Diff   0.00 ## Maximum % difference in band1-band2 Theta shifts
AveFIF_Xshift_Diff   0.00 ## Average % difference in band1-band2 X shifts
MaxFIF_Xshift_Diff   0.00 ## Maximum % difference in band1-band2 X shifts
AveFIF_Yshift_Diff   0.00 ## Average % difference in band1-band2 Y shifts
MaxFIF_Yshift_Diff   0.00 ## Maximum % difference in band1-band2 Y shifts

-------------------------Band_1------------------------
N.B. Image numbers below refer to order in input lists.

Number refined = 9
PercentRefinedImagesBand1 100.0 ## Percentage of input images refined (%)

AveNum_AbsSourcesBand1   0.67 ## Average number of absolute sources used over all refined
images (detected in bands 1 and 2)

"Fractional Differences" (%) in bnd1-bnd2 FIF Cartesian shifts:

Image # |      Theta    |    Xshift     |     Yshift
     1        0.000000       0.000000        0.000000
     2        0.000000       0.000000        0.000000
     3        0.000000       0.000000        0.000000
     4        0.000000       0.000000        0.000000
     5        0.000000       0.000000        0.000000
     6        0.000000       0.000000        0.000000
     7        0.000000       0.000000        0.000000
     8        0.000000       0.000000        0.000000
     9        0.000000       0.000000        0.000000


"Observed - Refined" residuals in units of arc-seconds.

Image # | Inp-Refnd RA | Inp-Refnd DEC | Inp-Refnd CROTA2 | #Abs.Sources used
     1       4.027092       1.044848      365.526062             0
     2       0.548773       0.743214      1295134.843358             0
     3       4.046361       0.898546      434.242301             0
     4       3.085945       0.188508      1295291.699240             0
     5       0.262006       0.142379      402.983207             0
     6       0.671323       0.872998       18.229302             6
     7       1.546364       0.713464      480.424901             0
     8       1.701993       1.343155      693.400917             0
```

```
     9       0.048061       1.175033      441.499577            0



-------------------------Band_2-------------------------
N.B. Image numbers below refer to order in input lists.

Number refined = 10
PercentRefinedImagesBand2 100.0 ## Percentage of input images refined (%)

AveNum_AbsSourcesBand2   0.60 ## Average number of absolute sources used over all refined
images (detected in bands 1 and 2)

"Fractional Differences" (%) in bnd1-bnd2 FIF Cartesian shifts:

Image # |      Theta    |    Xshift     |     Yshift
     1        0.000000       0.000000        0.000000
     2        0.000000       0.000000        0.000000
     3        0.000000       0.000000        0.000000
     4        0.000000       0.000000        0.000000
     5        0.000000       0.000000        0.000000
     6        0.000000       0.000000        0.000000
     7        0.000000       0.000000        0.000000
     8        0.000000       0.000000        0.000000
     9        0.000000       0.000000        0.000000
    10        0.000000       0.000000        0.000000


"Observed - Refined" residuals in units of arc-seconds.

Image # | Inp-Refnd RA | Inp-Refnd DEC | Inp-Refnd CROTA2 | #Abs.Sources used
     1       1.166749       0.099145      690.526303            0
     2       4.027092       1.044848      365.526062            0
     3       0.548773       0.743214      1295134.843358            0
     4       4.046361       0.898546      434.242301            0
     5       3.085945       0.188508      1295291.699240            0
     6       0.262006       0.142379      402.983207            0
     7       0.671323       0.872998       18.229302            6
     8       1.546364       0.713464      480.424901            0
     9       1.701993       1.343155      693.400917            0
    10       0.048061       1.175033      441.499577            0



-------------------------Band_3-------------------------
N.B. Image numbers below refer to order in input lists.

Number refined = 10
PercentRefinedImagesBand3 100.0 ## Percentage of input images refined (%)
```


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

```
AveNum_AbsSourcesBand3   0.60 ## Average number of absolute sources used over all refined
images (detected in bands 1 and 2)

"Fractional Differences" (%) in bnd1-bnd2 FIF Cartesian shifts:

Image # |      Theta    |    Xshift     |     Yshift
     1        0.000000       0.000000        0.000000
     2        0.000000       0.000000        0.000000
     3        0.000000       0.000000        0.000000
     4        0.000000       0.000000        0.000000
     5        0.000000       0.000000        0.000000
     6        0.000000       0.000000        0.000000
     7        0.000000       0.000000        0.000000
     8        0.000000       0.000000        0.000000
     9        0.000000       0.000000        0.000000
    10        0.000000       0.000000        0.000000


"Observed - Refined" residuals in units of arc-seconds.

Image # | Inp-Refnd RA | Inp-Refnd DEC | Inp-Refnd CROTA2 | #Abs.Sources used
     1       1.166749       0.099145      690.526303             0
     2       4.027092       1.044848      365.526062             0
     3       0.548773       0.743214      1295134.843358             0
     4       4.046361       0.898546      434.242301             0
     5       3.085945       0.188508      1295291.699240             0
     6       0.262006       0.142379      402.983207             0
     7       0.671323       0.872998       18.229302             6
     8       1.546364       0.713464      480.424901             0
     9       1.701993       1.343155      693.400917             0
    10       0.048061       1.175033      441.499577             0



-------------------------Band_4------------------------
N.B. Image numbers below refer to order in input lists.

Number refined = 10
PercentRefinedImagesBand4 100.0 ## Percentage of input images refined (%)

AveNum_AbsSourcesBand4   0.60 ## Average number of absolute sources used over all refined
images (detected in bands 1 and 2)

"Fractional Differences" (%) in bnd1-bnd2 FIF Cartesian shifts:

Image # |      Theta    |    Xshift     |     Yshift
     1        0.000000       0.000000        0.000000
     2        0.000000       0.000000        0.000000
     3        0.000000       0.000000        0.000000
     4        0.000000       0.000000        0.000000
```

```
     5        0.000000       0.000000        0.000000
     6        0.000000       0.000000        0.000000
     7        0.000000       0.000000        0.000000
     8        0.000000       0.000000        0.000000
     9        0.000000       0.000000        0.000000
    10        0.000000       0.000000        0.000000


"Observed - Refined" residuals in units of arc-seconds.

Image # | Inp-Refnd RA | Inp-Refnd DEC | Inp-Refnd CROTA2 | #Abs.Sources used
     1       1.166749       0.099145      690.526303            0
     2       4.027092       1.044848      365.526062            0
     3       0.548773       0.743214      1295134.843358            0
     4       4.046361       0.898546      434.242301            0
     5       3.085945       0.188508      1295291.699240            0
     6       0.262006       0.142379      402.983207            0
     7       0.671323       0.872998       18.229302            6
     8       1.546364       0.713464      480.424901            0
     9       1.701993       1.343155      693.400917            0
    10       0.048061       1.175033      441.499577            0
```

(4). If the **"debug"** (−d) switch is specified, an intermediate product in the form of a text file named "multibandrefine_data.dump" is generated. This lists the final post-averaged Cartesian offset values to be applied to images across all bands. This is mainly used for diagnostic purposes.

4.2.9. 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 6.1.1.

5. Algorithm

5.1. Algorithm Specifics

The WCS software library supports 26 different map projections with which to perform coordinate transformations. Pointing keywords need to conform to the standard FITS conventions (e.g. CRVAL1, CRVAL2, CRPIX1, CRPIX2, CDELT1, CDELT2 and CROTA2 for the standard TAN projection with no distortion correction). In general, all celestial coordinates are measured in degrees with $0 \le RA \le 360°$, $-90° \le DEC \le 90°$ and $0 \le CROTA2 \le 360°$ (the position angle measured East from North or in the direction of increasing RA).

The maximum number of FITS image files allowed in any input list is currently 1000. This is specified by the "MAX_NUMBER_IMAGES" parameter in the *multibandrefine.h* include file.

If the executable "multibandrefine" is entered at the Unix prompt with no command-line arguments, the following on-line tutorial is generated:

```
Program multibandrefine, Version 1.0

Usage: multibandrefine
 -n  <inp_namelist_fname>          (Optional)
 -f1 <inp_image_list_fname_band1>  (Required)
 -f2 <inp_image_list_fname_band2>  (Required only if "-s2" specified)
 -f3 <inp_image_list_fname_band3>  (Optional)
 -f4 <inp_image_list_fname_band4>  (Optional)
 -fi <inp_fiducial_frame_table>    (Required)
 -s1 <cartesian_shifts_fname_band1>(Required)
 -s2 <cartesian_shifts_fname_band2>(Optional)
 -o1 <out_table_fname_band1>       (Optional, Input headers always updated)
 -o2 <out_table_fname_band2>       (Optional, Input headers always updated)
 -o3 <out_table_fname_band3>       (Optional, Input headers always updated)
 -o4 <out_table_fname_band4>       (Optional, Input headers always updated)
 -l  <log_fname>                   (Optional, Default = 'stdout')
 -a  <ancillary_file_path>         (Optional, Default = ./)
 -qa (prints QA information to file "QAlogfile.txt")
 -d  (prints debug statements to stdout and files)
 -v  (verbose output)
 -vv (superverbose output)
```

5.2. Assumptions and Requirements

A. MULTIBANDREFINE assumes that each FITS image in every input list has a *single* FITS *header* that defines a unique pointing for a single plane contained therein, in other words, with the standard FITS keyword values: NAXIS = 2 or NAXIS3 = 1. If this is not true, the program will abort with a message sent to standard output.

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

B. Every image in any input list need NOT have the same pixel scale, i.e. different values for the standard CDELT header keywords are allowed. Also, they need NOT have the same dimensions as specified by NAXIS1 and NAXIS2. They can be non-square with NAXIS1 ≠ NAXIS2. The only requirement is that they pertain to the same wavelength band.

C. FITS images in any input list (namelist parameters: FITS_Image_List_Filename_chn[1-4]) are listed one per line and do not have to be in any specific order. The input band-specific image lists can also be of different size.

D. Repetition of FITS images in the same image list is not allowed!

E. Cartesian offsets in the offset lists for band-1 and band-2 *must correspond in the same order* to images listed in the respective band-1 and band-2 input FITS image lists. One offset list (with a corresponding band-1 input image list) is the minimum required input.

F. The maximum number of images allowed in the input list is currently 1000. This is defined by the MAX_NUMBER_IMAGES parameter in the include file *multibandrefine.h*.

G. Every image header is updated with a new set of (refined) keywords regardless if a refined solution existed or not. For images where no refinement was possible, the input CRVAL1, CRVAL2, CROTA2 keywords are reproduced to maintain a consistent set of pointing keywords for use in downstream ensemble processing.

6. Output

6.1. MULTIBANDREFINE Output

The output generated by MULTIBANDREFINE was outlined in depth in *section 4.2.8.*. To summarize, MULTIBANDREFINE is capable of generating the following output:

A.) Standard-output processing and status messages (if the verbose “-v” and/or super-verbose “-vv” switches are specified).

B.) New pointing keywords with refined values and uncertainties written to each input FITS header. These are updated with each subsequent execution of the multibandrefine software on the same input images.

C.) Optionally, band-dependent IPAC table files of refined pointing keyword values and uncertainties.

D.) Optionally, a QA log file listing fractional band-1-to-band-2 offset differences, *input – refined* pointing residuals in each coordinate and ancillary information.

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

F.) Optionally (if the debug switch "-d" is specified) intermediate diagnostic information on post-averaged band-1 and band-2 Cartesian offset values.

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

6.1.1 MULTIBANDREFINE 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. Below is an example of the log file output.

```
Program multibandrefine, Version 1.0
Input image list file for band 1 = ./testing/fitsimglist_ch1.txt
Input image list file for band 2 = ./testing/fitsimglist_ch2.txt
Input image list file for band 3 = ./testing/fitsimglist_ch3.txt
Input image list file for band 4 = ./testing/fitsimglist_ch4.txt
Input fiducial frame table = ./testing/fiducial_800.tbl
Input Cartesian Shifts file for band 1 = ./testing/shifts_ch1.dat
Input Cartesian Shifts file for band 2 = ./testing/shifts_ch2.dat
Output Table file (band 1 refined positions) = ./testing/refined1.tbl
Output Table file (band 2 refined positions) = ./testing/refined2.tbl
Output Table file (band 3 refined positions) = ./testing/refined3.tbl
Output Table file (band 4 refined positions) = ./testing/refined4.tbl
Ancillary Data-File Path = .
Verbose flag = 0
Super-verbose flag = 0
Debug flag = 0
QA flag = 1
```

```
QA log file = "QAlogfile.txt"
Performed multi-band ABSOLUTE pointing refinement.
Program multibandrefine: Status Message: 0x0000
Normal exit from Function 0x0000: LOG_WRITER
Processing time: 0.390000 seconds
Current date/time: Mon Aug 26 11:52:29 2002
Program multibandrefine, version 1.0, terminated successfully.
```

7. Testing

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

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

1. Tested MULTIBANDREFINE on simulated sets of IRAC images with 10-20 images for each of the four wavelength bands (listed in random order). These same images were first used in the "pointingrefine" program to generate Cartesian offset files.

2. Executed MULTIBANDREFINE with inputs read from and output written to directories different from where the program was run. Both namelist and command-line input mechanisms were exercised.

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

4. Executed MULTIBANDREFINE on four image lists of ≈1000 FITS images each to test for memory/speed limitations. On a SPARC-based CPU running at 500.4 MHz with 512 MB RAM, this took ≈ 190 seconds.

8. Usage Examples

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

```
multibandrefine –n multibandrefine.nl -v –vv | & tee out.log
```

Without using a namelist file but with output IPAC tables of refined pointing keyword values and QA log file generated:

```
multibandrefine -f1 fitsimglist_ch1.txt -f2
fitsimglist_ch2.txt -f3 fitsimglist_ch3.txt -f4
fitsimglist_ch4.txt -s1 shifts_ch1.dat -s2 shifts_ch2.dat –fi
fiducial.tbl -o1 refined1.tbl -o2 refined2.tbl -o3
refined3.tbl -o4 refined4.tbl –qa -v
```

9. Glossary

DCE	Data Collection Event
DN	Data Number
ICRS	International Celestial Reference System
IOC	In-Orbit Checkout
IRAC	Infrared Array Camera
IRAF	Image Reduction and Analysis Facility
QA	Quality Assurance
SDS	Subsystem Design Specification
SIS	Software Interface Specification
TBD	To Be Determined
TBR	To Be Resolved
WCS	World Coordinate System