674-SO-43, Version 6.7,
SSC-PD-4065

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: POINTINGREFINE

1 December 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	May 1, 2002
2.0	• For frame-to-absolute source matching, do not declare a match between two sources if more than one ambiguous match exists in the nominal source radius. This will make source matching between frame extractions and absolutes more reliable since no flux-matching constraint is imposed. • Included option of using RA, Dec, delta_RA, delta_Dec from source extraction tables directly. Defaults to reading *x, y* extractions and uncertainties. • Produce as additional diagnostic output, a text file of offsets in the mosaic tangent (reference) frame.	August 16, 2002
3.0	• Pre-determine array sizes to allocate memory for storage of positions and uncertainties of source matches from all possible overlapping images by computing a-priori the number of overlapping image-pairs expected in image list. Prior to this, the maximum possible number of correlated image-pairs was used for memory allocation. • Write diagnostic message to standard output for amount of memory being allocated in previous step. • Speed up source matching algorithm by initially sorting source positions (*in declination*) into ascending numerical order from all input extraction tables (and absolute source list). • Furthermore, made source matching more efficient and robust by declaring a frame-to-frame or a frame-to-absolute match *only if a single* match exists within the nominal search radius. • Set MAX_NUMBER_IMAGES = 3001 in *pointingrefine.h*	December 30, 2002
4.0	• Use measured pointing uncertainties as a-priori weights in globally minimized cost function. • Added two new name-list/command-line parameters to handle this:	March 13, 2003

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

	"Use_Apriori_Unc" and "Apriori_Syst_Unc".	
4.1	• Changed default random pointing uncertainty keywords to use from FITS headers in *pointingrefine.h* to SIGRA, SIGDEC and SIGPA (all in arcsec). • Included new namelist/command-line parameter to specify minimum random pointing uncertainty below which uncertainties may be unrealistic. If so, the values are set to this minimum value. The namelist parameter is "Min_Tolerable_Apriori_Uncert".	April 3, 2003
4.2	• Increased execution speed considerably. This was due to first, replacing "pow" functions by explicit multiplications and second (most importantly), removing unnecessary looping over correlated sources unrelated to my image in question when computing matrix elements.	May 5, 2003
4.5	• Implemented refinement of CD-matrix elements. These are only refined and written to output headers if they exist on input. • Included a header in "IPAC table format" for the absolute source table.	May 29, 2003
5.0	Derive observed crota2 from PA keyword in headers if present, default to WCS value if PA is not present, also write the refined CT2RFND value in addition to refined PA to all headers.	September 9, 2003
5.2	• Added option to only do absolute pointing refinement (i.e. only use frame-to-absolute matches in minimization sum). • Output cos(dec)*RA_Residual to stdout and QA log file instead of just difference RA_Residual = RA_obs – RA_refined. • Don't compute fabs of RA_Residual since need sign for CD matrix refinement. • Added debug info. (dump4 and dump5 log outputs) for correlated source separations before and after refinement.	September 22, 2003
5.3	Fixed bug when output uncertainties where being computed as NaNs and also made twist angle refinement more robust.	September 24, 2003
5.4	• Added option to only do XY translational refinement via flag Only_Refine_XY_Translations. When this is set can also allow refinement using one souce match between frames. • Placed MIN_MATCH_PER_IMAGE_PAIR parameter in .h file to use when	October 9, 2003

	Only_Refine_XY_Translations=0	
5.5	• Removed case where one single point source per image was allowed but kept ability to perform no rotational refinement. • Added option to read in specific reference image filename for relative refinement from namelist/command-line	October 10, 2003
5.6	Added option Min_Source_Matches_Per_Pair with default set to 2, can set to 1 if have one pnt src per image and two images in input list	October 11, 2003
5.7	Added keywords XC_RESID, YC_RESID which represent refinement correction in CRPIX1,CRPIX2 in the image pixel frame	October 14, 2003
5.8	Updates for Linux (Intel) compatibility.	October 16, 2003
5.9	Added IPAC-table style header to output Cartesian shifts file	October 22, 2003
6.0	Write information on average source-match separation before and after refinement to output QA logfile	October 26, 2003
6.1	• Added new QA diagnostics: AveRA_Change_ABS, AveDec_Change_ABS and AveCROTA2_Change_ABS to QAlogfile.txt file. These represent absolute changes in the pointing and twist angle residuals. • Use CRDER1, CRDER2 and UNCRTPA from input image headers (values in degrees; defined in pointingrefine.h) instead of old keywords SIGRA, SIGDEC and SIGPA to compute prior weights for global minimization.	December 11, 2003
6.2	• Compute χ^2 minimum value and number of degrees of freedom for overall cost-function and write to QAlogfile.txt file. • Made transformation of source extraction error ellipses from input frames to tangent reference frame more robust.	December 30, 2003
6.3	• Added capability to perform frame extraction-to-absolute astrometric source matching. Two new namelist/command-line parameters were added to support this: "Max_Flux_Diff_Absolutes" and "Abs_Flux_Scale_Factor". • Set MAX_NUMBER_IMAGES = 5001 in *pointingrefine.h*	March 23, 2004
6.4	Changed namelist parameter "Max_Num_Sources_Per_Image" to represent the Nth brightest sources (in flux) to select from each extraction table if the actual number of entries in a table exceeds this number. This guards against the danger	March 26, 2004

	of reading the first Nth entries which are sorted in position on the array.	
6.5	• Compute covariance between refined RA and Dec and in reference pixel position per image and write to FITS headers. Also write REFFLAG keyword to headers indicating if pointing was actually refined. • Produce output table containing refined pointing information for use by SSC super-boresight software according to SOSDL-SIS-PT-3005.	June 29, 2004
6.6	Made value of refined position-angle keyword (PA_RFND) in output product table "refinedPointing.tbl" negative (i.e., = -CROTA2 value), in accord with values in FITS header.	November 24, 2004
6.7	Added column "CovXY" which is the covariance between the X and Y offsets in the Fiducial Image Frame (FIF) to the output Cartesian shifts table.	December 1, 2004

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.

Deleted: Page Break

List of Figures

List of Tables

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

1. Introduction

1.1. Purpose and Scope

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

1.2. Document Organization

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

The material contained in this document represents the current understanding of the capabilities of the major SIRTF systems. Areas that require further analysis are noted by TBD (To Be Determined) or TBR (To Be Resolved). TBD indicates missing data that are not yet available. TBR indicates preliminary data that are not firmly established and are subject to change.

1.3. Relationship to Other Documents

The requirements on the operation of SIRTF flow down from the Science Requirements Document (674-SN-100) and the Facility Requirements Document (674-FE-100). The Science Operations System is governed by the SOS Requirements Document (674-SO-100). The current document is also cognizant of the requirements that appear in the Observatory Performance and Interface Control Document (674-SEIT-100) as well as the Flight Ground Interface Control Document (674-FE-101). This document is also affected by the FOS/SOS Interface Control Document (674-FE-102) that governs interfaces between the Flight Operations System and the Science Operations System. Related Software Interface Specifications (SIS) will be as indicated in Section 2.2 of this document.

1.4. Change Procedure

This document is a level 4 document according to the SIRTF Project Documentation Plan (674-FE-103). Changes to this document after approval require the approval of the SOS Change Board (TBD). The process for change control is described in the SOS Configuration Management Plan.

2. Overview

The soon-to-be-launched Space Infrared Telescope Facility (SIRTF) shall produce image data with an a-posteriori pointing knowledge of 1.4" (1 sigma radial) with a goal of 1.2" in the ICRS frame. In order to perform robust image co-addition, mosaic generation, extraction and position determination of sources to faint levels, the pointing will need to be refined to better than a few-tenths of an arc-second. Prior to refinement, images are processed for instrument-artifact removal and image-pointing data corrected for systematic boresight-array misalignment.

Input to the position refinement software “POINTINGREFINE” are point sources extracted from a mosaic of overlapping images. The software will use this information to find a "global minimization" of all relative offsets amongst all overlapping images. This is a novel method utilizing a generic linear sparse matrix solver. The pointings and orientations of SIRTF images can be refined in either a "relative" sense where pointings become fixed relative to a single image of a mosaic, or, in an "absolute" sense (in the celestial frame) if absolute point source information is known. Our goal is to produce science products with sub-arc-second pointing accuracy.

The software reads in as input a list of FITS images, corresponding list of source extraction tables (as generated by the “sourcestimate” software) and optionally, a list of absolute source pointings and fiducial frame table information. The software uses routines from the standard World Coordinate System Library (WCS) (Doug Mink, 2001, SAO) for pixel to sky coordinate conversions and the UMFPACK library for solving unsymmetric sparse linear matrix equations (T. A. Davis, 2002). All standard types WCS map-projections are supported. The primary output from POINTINGREFINE are *new* FITS header keywords giving the refined pointing/twist angle and observed-refined pointing residuals. Optionally, a table in IPAC format listing refined pointings and uncertainty information can be generated. POINTINGREFINE is written in ANSI/ISO C.

2.1. POINTINGREFINE Requirements

POINTINGREFINE is initiated by a startup script under the control of the pipeline executive and does its required functions for a given DCE image or pre-processed DCE image; this involves performing the following tasks.

A.) Retrieve the command line parameters passed by the start up script and use them to run the program.

B.) Read in as input a list of standard FITS images, a corresponding list of source extraction tables in IPAC format, optional absolute source list and fiducial frame table parameters and various processing parameters.

C.) Produce as primary output an IPAC table and new pointing information in FITS headers.

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 POINTINGREFINE 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 Super-boresight refinement SIS: SOSDL-SIS-PT-3005

2.3. Version History

2.3.1. Version 1.0

Initial version created on May 1, 2002.

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

2.3.2. Version 2.0

This version includes the following updates:

- For frame-to-absolute source matching, do not declare a match between two sources if more than one ambiguous match exists in the nominal source radius. This will make source matching between frame extractions and absolutes more reliable since no flux-matching constraint is imposed.

- Included option of using RA, Dec, delta_RA, delta_Dec from source extraction tables directly. Defaults to reading *x, y* extractions and uncertainties.

- Produce as additional diagnostic output, a text file of offsets in the mosaic tangent (reference) frame.

2.3.3. Version 3.0

This version includes the following updates:

- Pre-determine array sizes to allocate memory for storage of positions and uncertainties of source matches from all possible overlapping images by computing a-priori the number of overlapping image-pairs expected in image list. Prior to this, the maximum possible number of correlated image-pairs was used for memory allocation.

- Write diagnostic message to standard output for amount of memory being allocated in previous step.

- Speed up source matching algorithm by initially sorting source positions (*in declination*) into ascending numerical order from all input extraction tables (and absolute source list).

- Furthermore, made source matching more efficient and robust by declaring a frame-to-frame or a frame-to-absolute match *only if a single* match exists within the nominal search radius.

- Set MAX_NUMBER_IMAGES = 3001 in *pointingrefine.h*

2.3.4. Version 4.0

This version makes use of measured pointing uncertainties as a-priori weights in globally minimized cost function. Also, added two new namelist/command-line parameters to handle this: “Use_Apriori_Unc” and “Apriori_Syst_Unc”.

2.3.5. Version 4.1

Changed default random pointing uncertainty keywords to use from FITS headers in *pointingrefine.h* to SIGRA, SIGDEC and SIGPA (all in arcsec). Included new namelist/command-line parameter to specify minimum random pointing uncertainty below which uncertainties may be unrealistic. If so, the values are set to this minimum value. The namelist parameter is “Min_Tolerable_Apriori_Uncert”.

2.3.6. Version 4.2

Increased execution speed considerably. This was due to first, replacing “pow” functions by explicit multiplications and second (most importantly), removing unnecessary looping over correlated sources unrelated to my image in question when computing matrix elements.

2.3.7. Version 4.5

Implemented refinement of CD-matrix elements. These are only refined and written to output headers if they exist on input. Also included a header in “IPAC table format” for the absolute source table.

2.3.8. Version 5.0

Derive observed crota2 from PA keyword in headers if present, default to WCS value if PA is not present, also write the refined CT2RFND value in addition to refined PA to all headers.

2.3.9. Version 5.2

- Added option to only do absolute pointing refinement (i.e. only use frame-to-absolute matches in minimization sum).
- Output cos(dec)*RA_Residual to stdout and QA log file instead of just difference RA_Residual = RA_obs – RA_refined.
- Don't compute fabs of RA_Residual since need sign for CD matrix refinement.

- Added debug info. (dump4 and dump5 log outputs) for correlated source separations before and after refinement.

2.3.10. Version 5.3

Fixed bug when output uncertainties where being computed as NaNs and also made twist angle refinement more robust.

2.3.11. Version 5.4

- Added option to only do XY translational refinement via flag Only_Refine_XY_Translations. When this is set can also allow refinement using one souce match between frames.

- Placed MIN_MATCH_PER_IMAGE_PAIR parameter in .h file to use when Only_Refine_XY_Translations=0

2.3.12. Version 5.5

- Removed case where one single point source per image was allowed but kept ability to perform no rotational refinement.

- Added option to read in specific reference image filename for relative refinement from namelist/command-line

2.3.13. Version 5.6

Added option Min_Source_Matches_Per_Pair with default set to 2, can set to 1 if have one pnt src per image and two images in input list

2.3.14. Version 5.7

Added keywords XC_RESID, YC_RESID which represent refinement correction in CRPIX1,CRPIX2 in the image pixel frame

2.3.15. Version 5.8

Updates for Linux (Intel) compatibility.

2.3.16. Version 5.9

Added IPAC-table style header to output Cartesian shifts file

2.3.17. Version 6.0

Write information on average source-match separation before and after refinement to output QA logfile

2.3.18. Version 6.1

- Added new QA diagnostics: AveRA_Change_ABS, AveDec_Change_ABS and AveCROTA2_Change_ABS to QAlogfile.txt file. These represent absolute changes in the pointing and twist angle residuals.

- Use CRDER1, CRDER2 and UNCRTPA from input image headers (values in degrees; defined in pointingrefine.h) instead of old keywords SIGRA, SIGDEC and SIGPA to compute prior weights for global minimization.

2.3.19. Version 6.2

- Compute χ^2 minimum value and number of degrees of freedom for overall cost-function and write to QAlogfile.txt file.

- Made transformation of source extraction error ellipses from input frames to tangent reference frame more robust.

2.3.20. Version 6.3

- Added capability to perform frame extraction-to-absolute astrometric source matching. Two new namelist/command-line parameters were added to support this: “Max_Flux_Diff_Absolutes” and “Abs_Flux_Scale_Factor”.

- Set MAX_NUMBER_IMAGES = 5001 in *pointingrefine.h*

2.3.21. Version 6.4

Changed namelist parameter “Max_Num_Sources_Per_Image” to represent the Nth brightest sources (in flux) to select from each extraction table if the actual number of entries in a table exceeds this number. This guards against the danger of reading the first Nth entries which are sorted in position on the array.

2.3.22. Version 6.5

- Compute covariance between refined RA and Dec and in reference pixel position per image and write to FITS headers. Also write REFFLAG keyword to headers indicating if pointing was actually refined.
- Produce output table containing refined pointing information for use by SSC super-boresight software according to SOSDL-SIS-PT-3005.

2.4. Liens

Two minor liens have been identified:

1. 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 (18). This remains a lien of the current software.
2. If the input image list contains image clusters (sub-ensembles) which are disjoint from the nominal reference image frame (non-contiguous image-to-reference paths), then a singular matrix from the global minimization will result. If this occurs in “absolute refinement mode”, a second pass computation is performed and only those frames which contain absolute point sources are retained. These become the only images refined. In “relative refinement” mode, the software will abort with a message sent to standard output. A more robust scheme may have to be implemented which detects images that are contiguous with the reference image through frame-to-frame correlations if a “singular matrix” in the first pass is encountered, or, which treats the sub-ensembles in a self-consistent manner.

3. Input

3.1. POINTINGREFINE Input

POINTINGREFINE 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. POINTINGREFINE NAMELIST Input

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

Namelist variable	Description	Dim.	Type	Units	Default
FITS_Image_Reference_Filename	Specific FITS image to use as reference in **relative refinement mode**; Default = Use image with maximum # correlations from input list below	256	C	Null	See description column
FITS_Image_List_Filename	Required filename containing list of FITS-images.	256	C	Null	Null
Source_Table_List_Filename	Required filename containing list of source extraction tables in IPAC format.	256	C	Null	Null
Read_xy_Extractions	Optional: Use *x, y* pixel coordinates of extractions	1	I*1	Null	1

	from extraction tables. 1 = yes, 0 = no (use RA, Dec's).				
Fiducial_Frame_Table	Optional filename in IPAC format specified together with "Absolute_RA_DECs" (below). Specifies mosaic image (reference) frame.	256	C	Null	Null
Absolute_RA_DECs	Optional input text file of absolute point source pointings and uncertainties.	256	C	Null	Null
Data_Out_Filename	Optional output table filename of refined pointings and uncertainties.	256	C	Null	Null
Tangentshifts_Out_Filename	Optional output text filename of cartesian offsets in tangent reference image frame.	256	C	Null	Null
Max_Search_Radius	Optional search radius from each point source to perform position matching.	1	R*4	arc-second	5.0
Max_Flux_Diff	Optional largest flux difference tolerable for frame-to-frame flux matching.	1	R*4	Percent	5.0
Max_Flux_Diff_Absolutes	Optional largest flux difference tolerable for frame-to-absolute flux matching.	1	R*4	Percent	100.0
Abs_Flux_Scale_Factor	Scale factor for adjusting input absolute fluxes	1	R*4	-	1.0
Flux_Threshold	Optional flux threshold	1	R*4	Source flux	0

	above which to declare source matches.			units.	
Max_Num_Sources_Per_Image	The brightest Nth entries to read from each source extraction table.	1	I*2	Null	100
Use_Apriori_Unc	Optional: Use measured random uncertainties in RA, Dec if exist in FITS header. 1 = yes, 0 = no.	1	I*2	Null	1
Apriori_Syst_Unc	Optional constant systematic uncertainty to add onto RA, Dec random measurement uncertainties if they exist.	1	R*4	arc-second	0.0
Min_Tolerable_Apriori_Uncert	Optional minimum tolerable random uncertainty to cut off measured values if exist.	1	R*4	arc-second	0.0
Use_Only_Absolutes	Optional: Use _only_ frame-to-absolute matches when doing absolute refinement, 1=yes, 0=no.	1	I*2	Null	0
Only_Refine_XY_Translations	Only refine X,Y translations in reference frame, no rotational refinement; 1=yes, 0=no.	1	I*2	Null	0
Min_Source_Matches_Per_Pair	Minimum required number of source matches for declaring a correlated image pair (Usually >=2 for robustness; if have only one source per image then this must be 1 and no more than two images are allowed in	1	I*2	Null	2

	list); Default = 2				
Log_Filename	Optional output log filename	256	C	Null	stdout
Ancillary_File_Path	Pathname where supporting source files are installed.	256	C	Null	./ (current directory)

Table 1. Namelist file

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

```
 &POINTINGREFIN
  Comment = 'Generic namelist file for pointingrefine',
  Ancillary_File_Path =          '../pointingrefine_v7',
  Comment = 'Specific FITS image to use as reference in **relative refinement
mode**; Default = Use image with maximum # correlations from input list below',
  FITS_Image_Reference_Filename = '/stage/ssc-
pipe/fmasci/modules/pointingrefine_v7/testing/sws4322a00201CD.fits',
  FITS_Image_List_Filename =     './testing/pointingrefine_imagesCD.list',
  Source_Table_List_Filename =   './testing/pointingrefine_src_tbls.list',
  Comment = 'Read x,y positions from tables, 1=yes, 0=no (use RA,Dec instead),
Default = 1',
  Read_xy_Extractions =          1,
  Fiducial_Frame_Table =         './testing/fiducial.tbl',
  Absolute_RA_DECs =             './testing/absolute_ra_decs.list',
  Data_Out_Filename =            './testing/refined.tbl',
  Tangentshifts_Out_Filename =   './testing/tangentshifts.txt',
  Comment = '[arcsec], Default = 5.0',
  Max_Search_Radius =            5.0,
  Comment = 'For frame-to-frame flux matching: [percent], Default = 5',
  Max_Flux_Diff =                5.0,
  Comment = 'For frame-to-absolute flux matching: [percent], Default = 100',
  Max_Flux_Diff_Absolutes =      99.99,
  Comment = 'Scale absolute source fluxes by a factor for consistency with
extracted fluxes, Default = 1',
  Abs_Flux_Scale_Factor =        1.0,
  Comment = '[source extraction units], Default = 0',
  Flux_Threshold =               0,
  Comment = 'Use the following "brightest" entries from each extraction table,
Default = 100',
  Max_Num_Sources_Per_Image =    50,
  Comment = 'Use measured ptg uncertainties as a-priori weights if they exist in
FITS headers; 1=yes, 0=no; Default = 1',
  Use_Apriori_Unc =              1,
  Comment = '[arcsec], If input random ptg uncertainties exist, add following
constant systematic uncertainty, Default = 0',
```

```
  Apriori_Syst_Unc =             2.5,
  Comment = '[arcsec], If input random ptg uncertainties exist and are below
following value (either RA, Dec, or Twist), set them to this value, Default = 0',
  Min_Tolerable_Apriori_Uncert = 0.2,
  Comment = 'Use _only_ frame-to-absolute matches when doing absolute
refininement, 1=yes, 0=no; Default = 0',
  Use_Only_Absolutes =           0,
  Coment = 'Only refine X,Y translations in reference frame, no rotational
refinement; 1=yes, 0=no; Default = 0',
  Only_Refine_XY_Translations =  0,
  Comment = 'Minimum required number of source matches for declaring a correlated
image pair (Usually >=2 for robustness; if have only one source per image then
this must be 1 and no more than two images are allowed in list); Default = 2',
  Min_Source_Matches_Per_Pair =  2,
  Log_Filename =                 'stdout',
 &END
```

3.1.2. POINTINGREFINE 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. POINTINGREFINE FITS Input

POINTINGREFINE 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
-f2	Source_Table_List_Filename

-f3	Fiducial_Frame_Table
-f4	Absolute_RA_DECs
-fr	FITS_Image_Reference_Filename
-o	Data_Out_Filename
-ot	Tangentshifts_Out_Filename
-sr	Max_Search_Radius
-sd	Max_Flux_Diff
-sa	Max_Flux_Diff_Absolutes
-sb	Abs_Flux_Scale_Factor
-st	Flux_Threshold
-m	Max_Num_Sources_Per_Image
-t	Read_xy_Extractions
-b	Use_Apriori_Unc
-u	Apriori_Syst_Unc
-w	Min_Tolerable_Apriori_Uncert
-c	Use_Only_Absolutes
-e	Only_Refine_XY_Translations
-g	Min_Source_Matches_Per_Pair
-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. POINTINGREFINE Processing

POINTINGREFINE 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. POINTINGREFINE Processing Phases

POINTINGREFINE operates in thirteen phases: initialization, conversion of extracted point-source pixel positions to celestial coordinates if desired, determination of expected number of image-pair overlaps for memory allocation, point-source position and flux matching for all possible frame pairs, definition of reference/fiducial image frame, transformation of correlated source positions and uncertainties to reference-image frame, global minimization computation, compute offsets in mosaic

Cartesian plane, compute uncertainties in offsets, refinement of celestial pointings and twist angles, compute uncertainties in refined pointings, output results generation, and termination. This processing level is depicted in Figure 1.

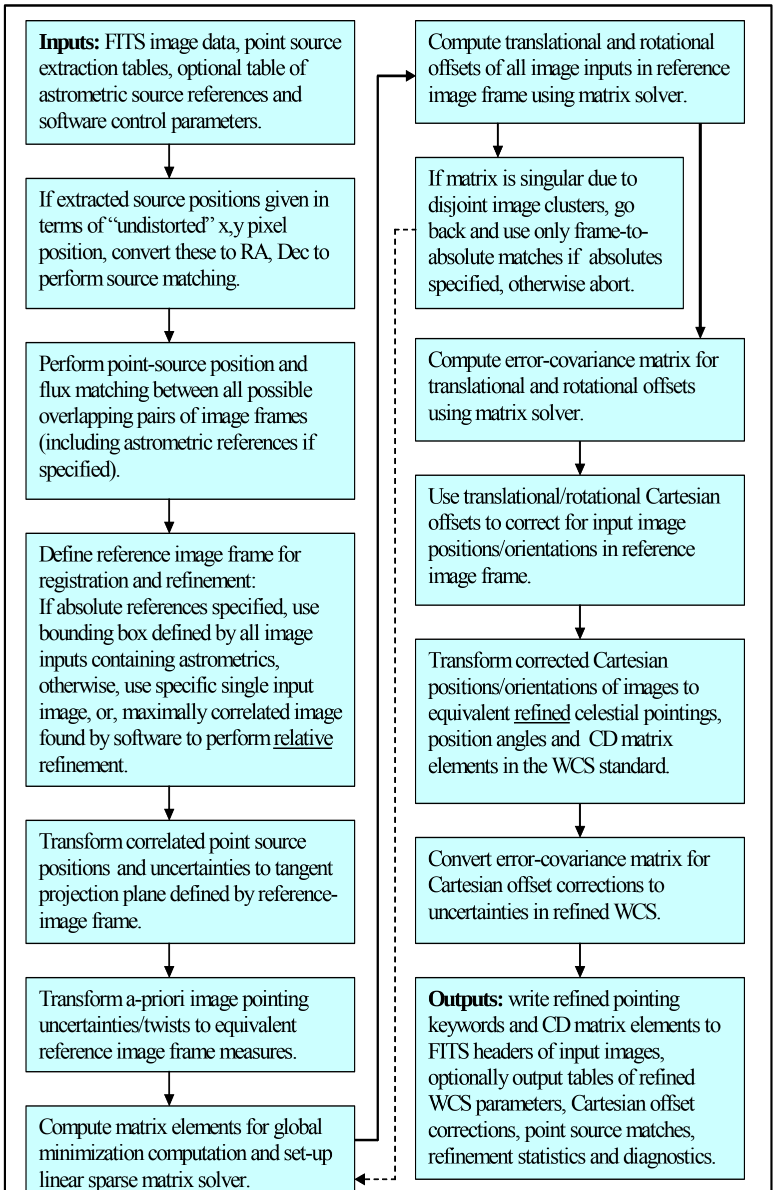
4.2.1. POINTINGREFINE Initialization

POINTINGREFINE 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.



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

Figure 1. POINTINGREFINE data and processing flow

4.2.2. Optional Pixel to RA, DEC Coordinate Conversion

If one wishes to use the the (x, y) pixel positions and corresponding uncertainties (*delta_x, delta_y*) of detected sources from the input source extraction tables (default mode) then a transformation to RA, DEC is performed. This is necessary in order to perform source matching in an "absolute" coordinate system. It is important to note that if the (x, y) pixel positions are used directly, then it is assumed that the source extractions were performed on images *corrected* for possible optical distortion. However, the source extraction tables output from the "sourcestimate" program contain (RA, Dec) positions and uncertainties (delta_RA, delta_Dec) already corrected for distortion. The distortion correction is included in the transformation. The use of (RA, Dec) directly will therefore be most optimal. The user can choose to read in celestial coordinates directly by specifying a "–t 0" on the command-line.

An example of an output table from the sourcestimate program is shown below. Only columns relevant to this software are shown.

```
\char comment = Output from SOURCESTIMATE, version  3.00
|x        |y       |delta_x    |delta_y    |RA          |Dec       |delta_RA  |delta_Dec  |flux
|r        |r       |r          |r          |r           |r         |r         |r          |r
 133.78    21.00    2.97e-01    2.16e-01     158.95311  59.14828  1.00e-04    7.28e-05   1.57e+01
 159.34    38.02    9.25e-02    2.92e-01     158.93620  59.15393  3.10e-05    9.86e-05   1.57e+01
 91.00     57.87    2.90e-01    1.85e-01     158.98100  59.16084  9.76e-05    6.27e-05   1.69e+01
               .
               .
               .
```

4.2.3. Expected Number of Image Pair Overlaps

We make memory allocation in the storage of correlated source positions as efficient as possible (Section 4.2.4) by guessing the total number of image-pairs expected to be overlapping in our image ensemble (input list). For a list of N images, there is a maximum of

$$N_{\text{maxpairs}} = \frac{1}{2} N(N-1)$$

distinct frame-pairs that can mutually overlap and hence could potentially contain correlated sources. This maximum occurs when all images are stacked more-or-less on top of each other. For a sparse mosaic or coadd, this number is smaller and thus reduces the required memory to allocate downstream.

We determine the number of images j in the input list overlapping (and hence potentially correlated) with any image i by first computing the distances between their centers and the center of our image of interest i on the sky: $d(RA_i, Dec_i \rightarrow RA_j, Dec_j)$. Taking circles of radii $R = \sqrt{2}\ L$ about each image center where L is the dimension of an image on a side (e.g. NAXIS1×CDELT1 [deg]), an image-pair (i, j) is guaranteed to overlap if the following is satisfied:

$$d(RA_i, Dec_i \rightarrow RA_j, Dec_j) < \sqrt{2}\, L\ .$$

This is repeated for all images i in the input list so that the total number of overlapping image-pairs "N_{pairs}" is found. We have assumed above that the dimension L is the same for each image in a pair, but in generality, the images could be non-square or have different pixel scales (CDELT's). To account for this, we take an ultra-conservative approach and use the largest possible L that can be derived in a pair of images so that L becomes image-pair dependent:

$$\begin{aligned} L = L_{ij} = {} & \max\left[\max(\mathrm{NAXIS1}_i, \mathrm{NAXIS2}_i),\ \max(\mathrm{NAXIS1}_j, \mathrm{NAXIS2}_j)\right] \times \\ & \max\left[\max(\mathrm{CDELT1}_i, \mathrm{CDELT2}_i),\ \max(\mathrm{CDELT1}_j, \mathrm{CDELT2}_j)\right], \end{aligned}$$

where the "max" function returns the maximum of the values enclosed in parenthesis.

4.2.4. Point Source Matching Between Frame Pairs

Prior to source matching, memory is allocated for storage of correlated source positions and uncertainties in every possible correlated image-pair in the input list. Given that the total number of image-pairs potentially containing correlated sources is N_{pairs} (as derived in Section 4.2.3), a maximum possible number of correlated sources per pair of N_{sources} (namelist parameter: Max_Num_Sources_Per_Image), and 8 double precision (64-bit floating point) numbers representing positions and uncertainties in an image-pair, the memory allocated is

$$\mathrm{Mem} = (6.4 \times 10^{-3}) N_{pairs} \left(\frac{N_{sources}}{100} \right) \mathrm{MB}$$

Note that this is not the *minimum* memory required to run the pointingrefine software (since memory is pre-allocated to arrays elsewhere) but it comes very close to it. This represents the most stringent memory requirement of any processing step in the software. For informational purposes, the amount of memory allocated in this step is written to standard output. If we consider the worst case scenario where all input images form a coadded stack so that the number of overlapping pairs is the maximum possible: $N_{\rm pairs} = \frac{1}{2} N_{imgs}(N_{imgs} - 1)$ where N_{imgs} is the number of input images, the required memory is:

$$\text{Mem} \approx 32\left(\frac{N_{imgs}}{100}\right)^2\left(\frac{N_{sources}}{100}\right)\text{MB} \quad \text{(\underline{worst case scenario})}.$$

Following memory allocation, source positions from all input extraction tables (including the absolute source list if specified) are sorted in the *declination* coordinate. The “quicksort” algorithm as described in *Numerical Recipes* (in C) is used to sort declinations in ascending numerical order. This preconditioning will speed up the source matching procedure by converting it to approximately an “N” process (instead of N^2) when matches are searched for between two source lists. Every possible image pair combination from the input list is searched for a “common” set of point sources in RA, DEC space. Both position and flux matching is performed. The position matching step attempts to find sources that fall within a nominal radius (namelist parameter: Max_Search_Radius).

Two types of matches are performed: frame-to-frame (between observed images) matching and frame-to-absolute source matching. For both frame-to-frame and frame-to-absolute matching, if more than one match is found within the search radius, no match is declared due to possible ambiguities. Only singly matched sources within the search radius are used. A caveat to this is that if the nominal search radius is “too large”, the more likely it is to miss bona-fide source matches to use in the refinement. It is therefore up to the user to fine tune the search radius so that the number of matches between image pairs is maximized and at the same time satisfies the inherent frame-to-frame pointing uncertainty.

As a rule of thumb, the Max_Search_Radius parameter should be set to allow for the *maximum* expected pointing uncertainty (if known apriori) and the *maximum* uncertainty in the positions (centroids) of point sources from frame extractions:

$$\text{Max\_Search\_Radius} \cong \left[\sigma^2_{centroid} + \sigma^2_{pointing}\right]^{1/2}.$$

In addition to position matching, sources (in both frame-to-frame and frame-to-absolute matching) are simultaneously matched in flux. To avoid use of an absolute flux scale, a flux-match is satisfied if any two fluxes fall within a maximum tolerable flux difference threshold (namelist parameters: Max_Flux_Diff and Max_Flux_Diff_Absolutes for frame-to-frame and frame-to-absolute matching respectively). In units of percent, this threshold is defined:

$$\text{Max\_Flux\_Diff} = 100\frac{|f_2 - f_1|}{f_2 + f_1},$$

where f_1 , f_2 are the respective fluxes of two sources. Due to possible wavelength differences and inconsistent flux units between extracted and absolute (astrometric) sources, the input absolute source fluxes can be re-scaled in the software via the namelist parameter: "Abs_Flux_Scale_Factor".

In addition to position and flux matching, a flux threshold (in source extractor units) can also be simultaneously applied (namelist/command-line parameter Flux_Threshold) above which to declare a match. If specified, this threshold is applied to both frame-to-frame and frame-to-absolute matches.

In order to declare two images as "correlated", the software also includes logic to ensure there exists at least two point sources common between the pair of images separated by a minimum distance of "CLOSEST_SOURCE_SEPN" pixels. The parameter CLOSEST_SOURCE_SEPN is defined in the include file "*pointingrefine.h*" and currently assumes the value of 5. This minimizes the uncertainty in the rotational offset between a pair of images, in particular when only two correlated sources are found. A minimum of two point sources is required to estimate a relative rotational offset.

4.2.5. Definition of Reference/Fiducial Image Frame

All correlated point sources between every possible image overlap are transformed to a rectilinear (x, y) *reference* frame chosen according to either of the following cases:

1. If NO absolute point source information (known catalog) is available (i.e. only relative pointing refinement is desired), the image in the input list which has the maximum number of overlaps with other images is chosen. This is the default mode of operation if no input absolute source list and fiducial frame table files are specified on input. This approach ensures that all neighboring images and consequently more images in the mosaic can be tied to this reference image when relative pointings are computed (see below). Alternatively, the user can specify a specific input image to use as the reference (namelist parameter: FITS_Image_Reference_Filename).

2. If absolute point source information is available (e.g. 2MASS catalog) then absolute pointing refinement is possible. A choice of reference frame here would be to define one which encompasses the full mosaic (i.e. the fiducial frame – derived beforehand). This can be treated as a "new" additional image to the input list of SIRTF-frames, containing the absolute point sources. When the SIRTF-images are refined "relative"

to this fiducial reference, they in reality become "absolutely" refined as we shall see below. In addition to performing absolute pointing refinement, the inclusion of absolute point sources also reduces the effect of an accumulation in "random" uncertainty in frame-to-frame offsets with distance from a single reference image.

An example format for the input fiducial frame table file (namelist parameter "Fiducial_Frame_Table") 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
```

The format for the absolute source list (namelist parameter "Absolute_RA_DECs") which must be specified with the above fiducial frame table if absolute refinement is desired is as follows. The absolute sources must reside within the fiducial-image frame. Currently, only four quantities are required in the absolute source list: in order from left to right: RA, Dec, σ(RA), σ(Dec), where uncertainties are 1-sigma:

```
\char  Absolute_source_list_for_fiducial_image_frame
\char  Query_catalog = '2MASS'
\char  Flux_density_band/wavelength = 'Ks'
|RA        |Dec       |UncRA   |UncDec  |Flux_Density |
|double    |double    |double  |double  |double       |
|deg       |deg       |deg     |deg     |microJansky  |
 82.332683  35.572376  0.00001  0.00001  100000
 82.381229  35.590476  0.00001  0.00001  200000
 82.477477  35.615665  0.00001  0.00001  300000
 82.279877  35.619709  0.00001  0.00001  400000
 82.369051  35.671982  0.00001  0.00001  500000
 81.905712  35.693850  0.00001  0.00001  600000
```

.
.
.

4.2.6. Transformation of Correlated Point Sources to Reference Image

The sky coordinates of all correlated (matched) point sources from all overlapping image pairs are transformed to the reference image frame using the standard WCS transformation. Associated positional uncertainties (either from frame extractions or the absolute input list) also need to be transformed to the reference (tangent) plane. Due to projection effects when one veers away from the tangent point (see Figure 3), the error ellipses representing these uncertainties need to be inflated accordingly. The following simple algorithm which uses the standard WCS library routines was devised.

Given positional uncertainties (Δx_{in}, Δy_{in}) of a source detected at position (x, y) in the frame of any input image (read from a source extraction table), the corresponding uncertainties in the tangent reference plane can be approximated by:

$$\Delta x_{ref} \approx \frac{CDELT1_{in}}{CDELT1_{ref}} \frac{1}{\cos(\theta_t)} \Delta x_{in} \qquad (1)$$

$$\Delta y_{ref} \approx \frac{CDELT2_{in}}{CDELT2_{ref}} \frac{1}{\cos(\theta_t)} \Delta y_{in} \qquad (2)$$

where the *CDELT1*, *CDELT2* are pixel scales in the x and y directions respectively at the relevant image centers and θ_t is the angle subtended by the separation of a source position and the tangent point in the reference image frame (see Figure 3).

The above expressions are approximations since first, no covariance information (or possible orientation of the error ellipse) between the Δx and Δy axes is accounted for and second, the error ellipse is assumed to be inflated isotropically in x and y with angular distance from the tangent point. This will allow us to be conservative and overestimate positional errors in the reference frame and sky. The second assumption is only valid for relatively small angular distances: θ_t<15°.

4.2.7 Global Minimization

For simplicity, shown below is a mosaic composed of three images, where one of the input images (labeled 1) defines the reference frame with coordinate axes (x, y). It is irrelevant whether this reference frame is the fiducial mosaic frame or a single input image. Our algorithm will be general in this regard. The filled and open circles are the same sources detected from each image of an overlapping pair and transformed into the reference frame of image 1. They are purposefully offset from each other to reflect the fact that the original images have random pointing uncertainties (including twist angle). These are the offsets we wish to compute.

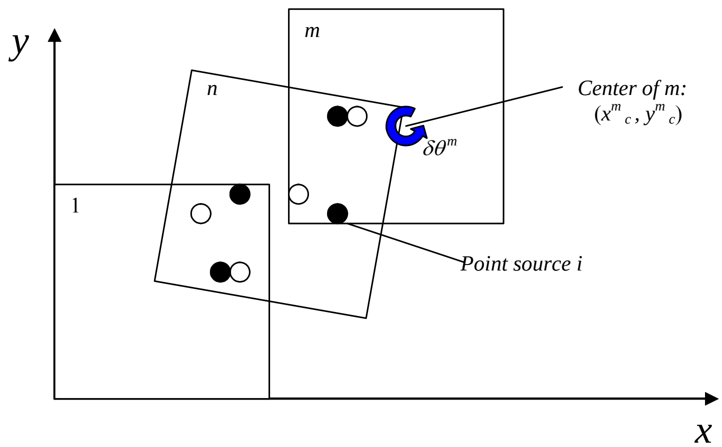


Figure 2. A simple three image mosaic

In general, the offset of some image *m* from another image *n* (see the above figure) can be represented by a rotation $\delta\theta^m$ and orthogonal translations δX^m and δY^m. In a rectilinear coordinate system, the coordinates of a point source *i* (in the frame of image 1) detected in an image (say *m*) can be transformed to its paired position in *n* as follows:

$$\begin{pmatrix} x_i^m \\ y_i^m \end{pmatrix} \rightarrow \begin{pmatrix} \tilde{x}_i^n \\ \tilde{y}_i^n \end{pmatrix} = \begin{pmatrix} x_c^m \\ y_c^m \end{pmatrix} + \begin{pmatrix} \cos\delta\theta^m & -\sin\delta\theta^m \\ \sin\delta\theta^m & \cos\delta\theta^m \end{pmatrix} \begin{pmatrix} x_i^m - x_c^m \\ y_i^m - y_c^m \end{pmatrix} + \begin{pmatrix} \delta X^m \\ \delta Y^m \end{pmatrix} \qquad (3)$$

where $(x^m{}_c, y^m{}_c)$ are coordinates of the center of image *m* and $\delta\theta^m$ is measured in the counterclockwise sense. The open circles overlapping between *m* and *n* represent point sources detected in image *m* so that a counterclockwise rotation of *m* will align the pairs of overlapping sources in the rotational sense only. Orthogonal translations in *x* and *y* are then needed to ensure complete alignment of sources.

The **main assumptions** here are:

1. The individual images are small enough on the sky that non-linear projection effects do not affect image-to-image offsets in the tangent plane of the reference image.

2. The mosaic extent is small enough as to avoid an exacerbation of "small-scale" non-linear projection effects at large distances from the mosaic's tangent point. A mosaic extent < 10° (maximum angular distance from the tangent point) is a good working measure. At angular distances $\theta_t \approx 10°$ from the mosaic tangent point, the relative displacement in the projected separation of two adjacent images (or two correlated sources therein) is ≈ 1.5% and varies as $\approx 1/\cos(\theta_t)$ (see Equations 1 and 2). Figure 2 shows the projection geometry.

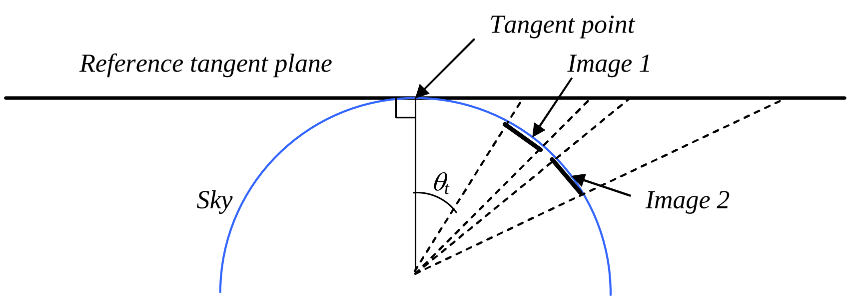


Figure 3. One-dimensional representation of projection geometry showing input images on sky and in mosaic (reference-image) tangent plane. Images 1 and 2 have the same physical size but different projected sizes.

Since uncertainties in the measured twist angle on the sky are expected to be small, the pair of equations defined by (1) can be linearized in $\delta\theta^m$ by assuming

$$\delta\theta^m \approx 0 \;\Rightarrow\; \sin\delta\theta^m \approx \delta\theta^m \text{ and } \cos\delta\theta^m \approx 1 \;. \tag{4}$$

Equation (3) can therefore be re-written as the pair of equations:

$$x_i^m \rightarrow \tilde{x}_i^n = x_i^m - (y_i^m - y_c^m)\delta\theta^m + \delta X^m \tag{5}$$
$$y_i^m \rightarrow \tilde{y}_i^n = y_i^m + (x_i^m - x_c^m)\delta\theta^m + \delta Y^m \;\;, \tag{6}$$

where $(\tilde{x}_i^n, \tilde{y}_i^n)$ represents a position "corrected" for relative image offset.

Let us define a cost function L, representing the sum of the squares of the "corrected" differences of all correlated point source positions in all possible correlated image pairs (*m, n*) in our mosaic:

$$L = \sum_{m,n}\sum_i \left\{ \frac{1}{\Delta x_i^{m,n}} \left[\tilde{x}_i^n - \tilde{x}_i^m\right]^2 + \frac{1}{\Delta y_i^{m,n}} \left[\tilde{y}_i^n - \tilde{y}_i^m\right]^2 \right\} + L_{ap} \tag{7}$$

where

$$\Delta x_i^{m,n} = \sigma^2(x_i^m) + \sigma^2(x_i^n)$$
$$\Delta y_i^{m,n} = \sigma^2(y_i^m) + \sigma^2(y_i^n)$$

and the σ^2 represent variances in extracted point source positions.

The additive term L_{ap} represents an "a-priori" weighting function which makes use of actual measured pointing uncertainties. This function is defined as:

$$L_{ap} = \sum_{m,n} \left\{ \frac{(\delta X^m)^2}{\sigma_{Xm}^2} + \frac{(\delta Y^m)^2}{\sigma_{Ym}^2} + \frac{(\delta\theta^m)^2}{\sigma_{\theta m}^2} + \frac{(\delta X^n)^2}{\sigma_{Xn}^2} + \frac{(\delta Y^n)^2}{\sigma_{Yn}^2} + \frac{(\delta\theta^n)^2}{\sigma_{\theta n}^2} \right\} \tag{8}$$

where the σ_{jm}^2, σ_{jn}^2 ($j = X, Y, \theta$) represent *measured* pointing variances (in the ICRS) transformed into the Cartesian reference image frame. Their computation is described in Appendix I.

The purpose of including L_{ap} is to avoid over-refining or biasing those images whose inherent measured pointing uncertainties are already small (within nominal bounds). In other words, for those images whose pointing uncertainties are known to be small a-priori will have a larger contribution to L_{ap} relative to that contributed by the correlated-source term (double sum in Equation 7). As a result, the solution will be biased towards minimizing L_{ap} and not the correlated-source term which could potentially degrade the final refined pointing (mathematically, in the limit $L \rightarrow L_{ap}$, the global minimum is satisfied by $\delta\theta \approx \delta X \approx \delta Y \approx 0$). Vice-versa, large pointing uncertainties will bias the solution towards the correlated source term where refinement using point-source correlation is obviously needed.

The outer sum in Equations (7) and (8) is over all correlated image pairs (m, n) including the reference image, while the inner sum in Equation (7) is over all correlated point sources i which belong in the overlap region of image pair (m, n).

Equation (7) can be re-written in terms of physical image offsets $\delta\theta$, δX and δY for each image by use of equations (5) and (6):

$$L = \sum_{m,n}\sum_{i}\left\{\frac{1}{\Delta x_i^{n,m}}\left[x_i^m - (y_i^m - y_c^m)\delta\theta^m + \delta X^m - x_i^n + (y_i^n - y_c^n)\delta\theta^n - \delta X^n\right]^2 + \right.$$

$$\left.\frac{1}{\Delta y_i^{n,m}}\left[y_i^m + (x_i^m - x_c^m)\delta\theta^m + \delta Y^m - y_i^n - (x_i^n - x_c^n)\delta\theta^n - \delta Y^n\right]^2\right\} +$$

$$\sum_{m,n}\left\{\frac{(\delta X^m)^2}{\sigma_{Xm}^2} + \frac{(\delta Y^m)^2}{\sigma_{Ym}^2} + \frac{(\delta\theta^m)^2}{\sigma_{\theta m}^2} + \frac{(\delta X^n)^2}{\sigma_{Xn}^2} + \frac{(\delta Y^n)^2}{\sigma_{Yn}^2} + \frac{(\delta\theta^n)^2}{\sigma_{\theta n}^2}\right\} \qquad (9)$$

Our aim is to minimize L with respect to all image offsets $\delta\theta^m$, δX^m and δY^m where m corresponds to an image which is correlated with any others in the mosaic (including the refererence image). By definition, the reference image has $\delta\theta = 0$, $\delta X = 0$ and $\delta Y = 0$. At the "global" minimum of L, the derivatives with respect to the three offsets vanish (for any image m):

$$\frac{\partial L}{\partial \delta\theta^m} = 0, \qquad \frac{\partial L}{\partial \delta X^m} = 0, \qquad \frac{\partial L}{\partial \delta Y^m} = 0 \qquad (10)$$

Evaluating the partial derivatives in (10) for image m (an arbitrary image in our mosaic), leads us to the following:

$$\frac{\partial L}{\partial \delta\theta^m} = \sum_n \sum_i \left\{ \frac{-2(y_i^m - y_c^m)}{\Delta x_i^{n,m}} \left[x_i^m - (y_i^m - y_c^m)\delta\theta^m + \delta X^m - x_i^n + (y_i^n - y_c^n)\delta\theta^n - \delta X^n \right] + \frac{2(x_i^m - x_c^m)}{\Delta y_i^{n,m}} \left[y_i^m + (x_i^m - x_c^m)\delta\theta^m + \delta Y^m - y_i^n - (x_i^n - x_c^n)\delta\theta^n - \delta Y^n \right] \right\} + \sum_n \frac{2\delta\theta^m}{\sigma_{\theta m}^2} \quad \text{(A)}$$

$$\frac{\partial L}{\partial \delta X^m} = \sum_n \sum_i \frac{2}{\Delta x_i^{n,m}} \left[x_i^m - (y_i^m - y_c^m)\delta\theta^m + \delta X^m - x_i^n + (y_i^n - y_c^n)\delta\theta^n - \delta X^n \right] + \sum_n \frac{2\delta X^m}{\sigma_{Xm}^2} \quad \text{(B)}$$

$$\frac{\partial L}{\partial \delta Y^m} = \sum_n \sum_i \frac{2}{\Delta y_i^{n,m}} \left[y_i^m + (x_i^m - x_c^m)\delta\theta^m + \delta Y^m - y_i^n - (x_i^n - x_c^n)\delta\theta^n - \delta Y^n \right] + \sum_n \frac{2\delta Y^m}{\sigma_{Ym}^2} \quad \text{(C)}$$

For all correlated images m in a mosaic, equations (A), (B) and (C) form a simultaneous set of equations. In general, for N_{corr} correlated images, we will have a set of $3(N_{corr} - 1)$ equations in $3(N_{corr} - 1)$ unknowns. The "−1" factor excludes the reference image where by definition $\delta\theta = 0$, $\delta X = 0$ and $\delta Y = 0$. As discussed above, if absolute source positions are known, these will form part of the "fiducial" frame encompassing the entire mosaic becoming the reference frame itself. This image can be treated in the normal way as if we had chosen a single input image as the "reference". Equation (9), as well as (A), (B) and (C) implicitly assume this reference image in the sum over all correlated pairs (m, n) when $n = reference$, for every image of interest m.

4.2.8 Solving for Offsets in the Mosaic Reference Image Frame

By setting equations (A), (B) and (C) to zero, we have a set of three general equations for a given image m. These need to be solved simultaneously for every correlated image m in the mosaic (excluding the reference image). One can isolate the coefficients of the offsets $\delta\theta^m$, δX^m, δY^m and

$\delta\theta^n$, δX^n, δY^n from equations (A), (B) and (C) which consist of sums over image pairs (*m, n*) and sources *i* correlated therein. The problem reduces to solving the following matrix equation for *X*:

$$A\,X = \Psi \tag{11}$$

where *A* is a $3(N_{corr} - 1) \times 3(N_{corr} - 1)$ coefficient matrix and Ψ is a column matrix containing constant terms characteristic of our mosaic. *X* is our "unknown" column matrix with $3(N_{corr} - 1)$ unknowns. Let us define the coefficients of $\delta\theta^m$, δX^m, δY^m and $\delta\theta^n$, δX^n, δY^n as well as constant terms in each of equations (A), (B) and (C) respectively by:

$$\begin{aligned}
&A_\theta^m\,,\ A_X^m\,,\ A_Y^m\,,\ A_\theta^n\,, A_X^n\,,\ A_Y^n,\ \ \text{constant term} = \Psi_A(m,n)\\
&B_\theta^m\,,\ B_X^m\,,\ B_\theta^n\,, B_X^n\,,\ \ \text{constant term} = \Psi_B(m,n)\\
&C_\theta^m\,,\ C_Y^m\,,\ C_\theta^n\,,\ C_Y^n\,,\ \ \text{constant term} = \Psi_C(m,n)
\end{aligned} \tag{12}$$

Using the conditions defined in (10), these coefficients and terms are explicitly given by:

$$A_\theta^m = \sum_n \sum_i \frac{(y_i^m - y_c^m)^2}{\Delta x_i^{n,m}} + \frac{(x_i^m - x_c^m)^2}{\Delta y_i^{n,m}} + \sum_n \frac{1}{\sigma_{\theta m}^2}$$

$$A_X^m = -\sum_n \sum_i \frac{(y_i^m - y_c^m)}{\Delta x_i^{n,m}}$$

$$A_Y^m = \sum_n \sum_i \frac{(x_i^m - x_c^m)}{\Delta y_i^{n,m}}$$

$$A_\theta^n = -\sum_i \frac{(y_i^m - y_c^m)(y_i^n - y_c^n)}{\Delta x_i^{n,m}} + \frac{(x_i^m - x_c^m)(x_i^n - x_c^n)}{\Delta y_i^{n,m}}$$

$$A_X^n = \sum_i \frac{(y_i^m - y_c^m)}{\Delta x_i^{n,m}}$$

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

$$A_Y^n = -\sum_i \frac{(x_i^m - x_c^m)}{\Delta y_i^{n,m}}$$

$$\Psi_A(m,n) = -\sum_n \sum_i \frac{(y_i^m - y_c^m)(x_i^n - x_i^m)}{\Delta x_i^{n,m}} + \frac{(x_i^m - x_c^m)(y_i^m - y_i^n)}{\Delta y_i^{n,m}}$$

$$B_\theta^m = -\sum_n \sum_i \frac{(y_i^m - y_c^m)}{\Delta x_i^{n,m}}$$

$$B_X^m = \sum_n \sum_i \frac{1}{\Delta x_i^{n,m}} + \sum_n \frac{1}{\sigma_{Xm}^2}$$

$$B_\theta^n = \sum_i \frac{(y_i^n - y_c^n)}{\Delta x_i^{n,m}}$$

$$B_X^n = -\sum_i \frac{1}{\Delta x_i^{n,m}}$$

$$\Psi_B(m,n) = -\sum_n \sum_i \frac{(x_i^m - x_i^n)}{\Delta x_i^{n,m}}$$

$$C_\theta^m = \sum_n \sum_i \frac{(x_i^m - x_c^m)}{\Delta y_i^{n,m}}$$

$$C_Y^m = \sum_n \sum_i \frac{1}{\Delta y_i^{n,m}} + \sum_n \frac{1}{\sigma_{Ym}^2}$$

$$C_\theta^n = -\sum_i \frac{(x_i^n - x_c^n)}{\Delta y_i^{n,m}}$$

$$C_Y^n = -\sum_i \frac{1}{\Delta y_i^{n,m}}$$

$$\Psi_C(m,n) = -\sum_n \sum_i \frac{(y_i^m - y_i^n)}{\Delta y_i^{n,m}}$$

To apply equations (A), (B) and (C) to our 3-image mosaic (see figure above), we will re-label image m with the new label m_1 and image n with label m_2. The reason for this is that these equations were derived in the general case and we will need to apply them to each image separately where $m = (m_1, m_2)$ and n is a dummy index ($\neq m$) used in the summation over correlated image pairs (including the reference image). Applying the set of equations A, B and C to images m_1 and m_2 independently, the matrix equation (11) can be represented as:

$$\begin{pmatrix} A_\theta^{m=m1} & A_X^{m=m1} & A_Y^{m=m1} & A_\theta^{n=m2} & A_X^{n=m2} & A_Y^{n=m2} \\ B_\theta^{m=m1} & B_X^{m=m1} & 0 & B_\theta^{n=m2} & B_X^{n=m2} & 0 \\ C_\theta^{m=m1} & 0 & C_Y^{m=m1} & C_\theta^{n=m2} & 0 & C_Y^{n=m2} \\ A_\theta^{n=m1} & A_X^{n=m1} & A_Y^{n=m1} & A_\theta^{m=m2} & A_X^{m=m2} & A_Y^{m=m2} \\ B_\theta^{n=m1} & B_X^{n=m1} & 0 & B_\theta^{m=m2} & B_X^{m=m2} & 0 \\ C_\theta^{n=m1} & 0 & C_Y^{n=m1} & C_\theta^{m=m2} & 0 & C_Y^{m=m2} \end{pmatrix} \begin{pmatrix} \delta\theta^{m1} \\ \delta X^{m1} \\ \delta Y^{m1} \\ \delta\theta^{m2} \\ \delta X^{m2} \\ \delta Y^{m2} \end{pmatrix} = \begin{pmatrix} \Psi_A(m_1,n) \\ \Psi_B(m_1,n) \\ \Psi_C(m_1,n) \\ \Psi_A(m_2,n) \\ \Psi_B(m_2,n) \\ \Psi_C(m_2,n) \end{pmatrix} \qquad (13)$$

The coefficient matrix A in equation (13) falls under the category of a sparse matrix due to the presence of a repeatable number of zero elements. The fraction of "zeros" will usually be > 22% and the minimum of ≈22% (as seen in the above example) occurs when every image in the mosaic is correlated with every other, such as in a stack. The level of "sparsity" in A will increase with non-zero elements in a block diagonal if one desires to tie and refine images to an absolute reference frame alone where n = *reference* and all coefficients with superscript n in A are zero. In general, the maximum and minimum possible number of *non-zero* elements in A is given by:

$$\begin{aligned} N(\text{min non-zeros}) &= 7(N_{corr} - 1) \\ N(\text{max non-zeros}) &= 7(N_{corr} - 1)^2, \end{aligned} \qquad (14)$$

where N_{corr} is the number of correlated images in the input list, including the reference image.

An assumption here is that images which are NOT correlated with any others in a mosaic will have their offsets explicitly set to zero: $\delta\theta^m = \delta X^m = \delta Y^m = 0$. Due to the lack of correlated point source positions, such images do not contribute to our cost function L. The best we can do is not refine their positions at all and assume their relative offsets are zero.

The matrix equation (13) is solved using the "UMFPACK" library, designed for solving unsymmetric sparse linear systems using direct sparse LU factorization (T. A. Davis, Version 4.0, April 11, 2002). It is written in ANSI/ISO C and relies on the Level-3 Basic Linear Algebra Subprograms (BLAS) for its performance. The library is portable to many versions of UNIX (Sun-solaris, Red-Hat Linux, IBM AIX, SGI IRIX and Compaq Alpha). The library also includes a scheme to correct solutions for possible accumulations in round-off error during the matrix decomposition stage (e.g. LU-factorization). The final solution is improved by solving for deviations from the real solution iteratively (see Numerical Recipes page 55 for a discussion of the method).

4.2.9 Offset Uncertainties in the Mosaic Reference Image Frame

We compute uncertainties and covariances between all image offsets by computing the inverse matrix (A^{-1}) which effectively represents the full error-covariance matrix. Variances in each offset are along the diagonal of A^{-1} and covariances are given by off-diagonal elements. We compute A^{-1} using the same sparse matrix solver on each "unknown" column of A^{-1} with the corresponding column in the identity matrix I on the right hand side. If X_c represents an unknown column of A^{-1} and I_c the same column in the identity matrix, then solving $AX_c = I_c$ repeatedly for every column in I will yield A^{-1} since $AA^{-1} = I$.

4.2.10 Refinement of Celestial Pointings

Once the offsets of every mutually correlated image m ($\delta\theta^m$, δX^m , δY^m) are computed, we correct the tangent points (usually image centers in reference image coordinates – i.e. $x^m{}_c$, $y^m{}_c$ in the figure below) corresponding to CRVAL1 and CRVAL2 (RA, DEC). This can be done using the original transformation equations (3) and (4). Since the rotation is about the centers, these transformations reduce to:

$$x_c^m(\text{new}) = x_c^m(\text{old}) + \delta X^m \qquad (15)$$

$$y_c^m(\text{new}) = y_c^m(\text{old}) + \delta Y^m \qquad (16)$$

Using the WCS parameters of the reference image (fiducial or otherwise), 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 3 below. This is chosen because the *angle* 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 4). 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 $\delta\theta$. 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 4). The derivation is given below.

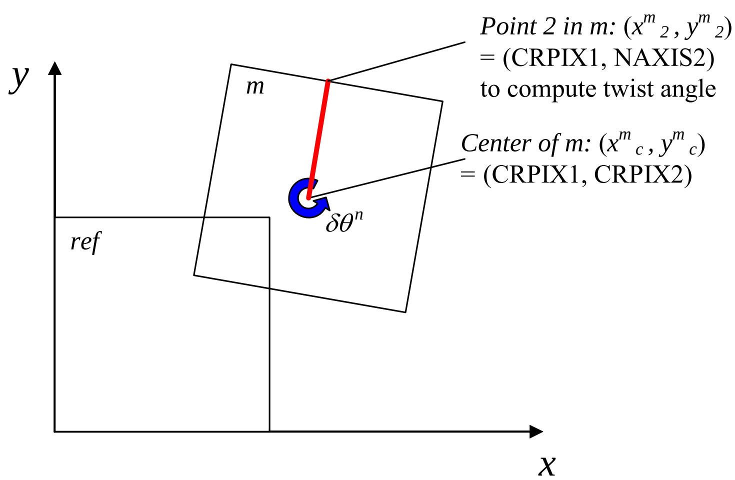


Figure 4: Schematic showing the two-points per image for computing the twist angle.

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:

$$\frac{\sin\gamma}{\sin d_{AB}} = \frac{\sin|\alpha_c - \alpha_2|}{\sin d_{BC}} \quad (17)$$

$$\Rightarrow \quad \gamma = \sin^{-1}\left[\frac{\sin d_{AB}}{\sin d_{BC}}\sin|\alpha_c - \alpha_2|\right]. \quad (18)$$

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:

$$d_{AB} = \cos^{-1}[\sin\delta_2] \quad (19)$$

$$d_{BC} = \cos^{-1}[\sin\delta_2 \sin\delta_c + \cos\delta_2 \cos\delta_c \cos(\alpha_2 - \alpha_c)]. \quad (20)$$

Care must be taken when computing γ from equation (18) 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 $\alpha_2 > \alpha_c$.

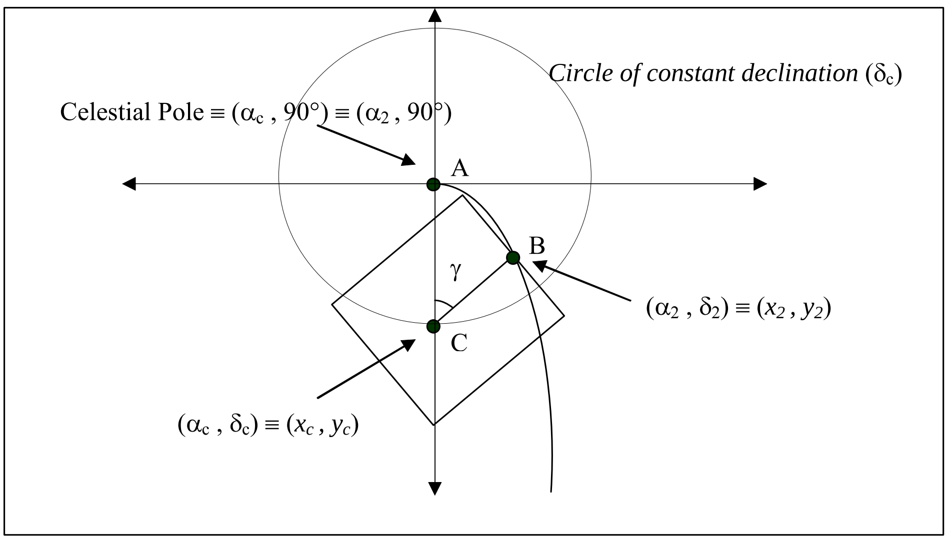


Figure 5. Schematic used in the derivation of the sky twist angle.

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

4.2.11 Uncertainties in Refined Pointings

We compute uncertainties in refined RA and Dec. by transforming selected components of the full error-covariance matrix from the reference image frame (Section 4.2.9) to the sky. This is accomplished via a similarity and rotational transformation. If $\Delta\alpha$ and $\Delta\delta$ are given offsets in RA and Dec. respectively, then these are related to equivalent offsets in the reference image pixel frame as follows:

$$\begin{pmatrix}\Delta\alpha \\ \Delta\delta\end{pmatrix} = \begin{pmatrix}\cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma\end{pmatrix}\begin{pmatrix}S_x\Delta x \\ S_y\Delta y\end{pmatrix} \qquad (21)$$

where S_x , S_y are reference image pixel scales ($\equiv$ CDELT1, CDELT2 respectively) and γ is the reference image twist angle (measured East from North; $\equiv$ CROTA2). Expanding Eqn. (21) into two equations, squaring each one and using the following relationships between (co)variances and expectation values: $\langle\Delta x^2\rangle = \sigma_x^2 + \langle\Delta x\rangle^2$; $\langle\Delta x\Delta y\rangle = \mathrm{cov}(x,y) + \langle\Delta x\rangle\langle\Delta y\rangle$, we arrive at the following expressions for (co)variances in RA and Dec. in terms of (co)variances in the pixel image frame:

$$\sigma_\alpha^2 = S_x^2\sigma_x^2\cos^2\gamma \;+\; S_y^2\sigma_y^2\sin^2\gamma \;-\; 2\cos\gamma\sin\gamma\,\mathrm{cov}(x,y)\left|S_xS_y\right|$$

$$\sigma_\delta^2 = S_y^2\sigma_y^2\cos^2\gamma \;+\; S_x^2\sigma_x^2\sin^2\gamma \;+\; 2\cos\gamma\sin\gamma\,\mathrm{cov}(x,y)\left|S_xS_y\right|$$

$$\mathrm{cov}(\alpha,\delta) = (\cos^2\gamma - \sin^2\gamma)\,\mathrm{cov}(x,y)\left|S_xS_y\right| \;+\; (\cos\gamma\sin\gamma)(S_x^2\sigma_x^2 - S_x^2\sigma_x^2)\ , \qquad (22)$$

where the last expression for the covariance is obtained by multiplying the two equations in Eqn (21) and taking expectation values of the cross products. Since $S_x \equiv$ CDELT1 is negative in the WCS convention (i.e., represents a left-handed coordinate system), we must multiply the covariance in Eqn (22) by –1. An equivalent measure of the covariance is the *co-standard deviation* (*csd*). This is defined in terms of the covariance as follows:

$$csd(\alpha,\delta) = sign[\mathrm{cov}(\alpha,\delta)]\sqrt{\left|\mathrm{cov}(\alpha,\delta)\right|}\ .$$

This quantity instead of the covariance is written to FITS headers (keyword CSRDRFND).

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 |δ| < 60°). Close to the poles, uncertainties in RA are 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 (18). This remains a lien of the current software.

We also compute the refinement correction in the reference pixel position (CRPIX1, CRPIX2) of each input image by transforming the refined RA, Dec to it's equivalent position in the pixel frame. Uncertainties associated with the refined reference pixel position are also computed. These are computed using the reverse of the transformation represented by Eqn (21):

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} S_x^{-1}\cos\gamma & S_x^{-1}\sin\gamma \\ S_y^{-1}\sin\gamma & S_y^{-1}\cos\gamma \end{pmatrix} \begin{pmatrix} \Delta\alpha \\ \Delta\delta \end{pmatrix} \qquad (21')$$

Taking expectation values of the squares and cross products of the equations represented by Eqn (21') leads us to the following expressions for (co)variances in the image pixel frames in terms of equivalent RA, Dec (co)variances.

$$\sigma_x^2 = \frac{1}{S_x^2}\left[\sigma_\alpha^2\cos^2\gamma \;+\; \sigma_\delta^2\sin^2\gamma \;+\; 2\cos\gamma\sin\gamma\,\mathrm{cov}(\alpha,\beta)\right]$$

$$\sigma_y^2 = \frac{1}{S_y^2}\left[\sigma_\delta^2\cos^2\gamma \;+\; \sigma_\alpha^2\sin^2\gamma \;-\; 2\cos\gamma\sin\gamma\,\mathrm{cov}(\alpha,\beta)\right]$$

$$\mathrm{cov}(\alpha,\delta) = \frac{1}{S_x S_y}\left[(\cos^2\gamma - \sin^2\gamma)\,\mathrm{cov}(\alpha,\delta) \;+\; (\cos\gamma\sin\gamma)(\sigma_\delta^2 - \sigma_\alpha^2)\right]. \qquad (22')$$

The sign of the product $S_x S_y$ in the denominator of Eqn (22') takes care of the sign of the overall covariance, i.e., if $S_x \equiv$ CDELT1 is negative which represents a left handed coordinate system. Just like for RA and Dec, we write the *co-standard deviation* (*csd*) to FITS headers (keyword CSDXCYC), defined as:

$$csd(x,y) = sign[\mathrm{cov}(x,y)]\sqrt{|\mathrm{cov}(x,y)|}\ .$$

4.2.12 Refinement of CD Matrix

The new FITS standard will replace the WCS keywords *CROTA2*, *CDELT1*, *CDELT2* by four "CD-matrix" keywords which in the absence of distortion and skew, is defined as follows:

$$\begin{pmatrix} CD_{11} & CD_{12} \\ CD_{21} & CD_{22} \end{pmatrix} = \begin{pmatrix} CDELT1.\cos(CROTA2) & -CDELT2.\sin(CROTA2) \\ CDELT1.\sin(CROTA2) & CDELT2.\cos(CROTA2) \end{pmatrix} \tag{23}$$

Since pointing refinement only shifts and rotates input images, these operations do not change the "skewness" or inherent distortion of the native coordinate system defined by the CD matrix. The only change to the CD matrix is contributed by a change in *CROTA2* to a new value "*CROTA_refined*". As outlined in the document: "CD-Matrix Implementation" (D. Makovoz; 05/02/03), this can be represented by an effective rotation to the CD matrix by angle γ where

$$\gamma = CROTA2\_refined - CROTA2 \tag{24}$$

and is defined as increasing in the counter-clockwise direction so that a coordinate (x_{new}, y_{new}) in a new (refined) coordinate system is given by:

$$\begin{pmatrix} x_{new} \\ y_{new} \end{pmatrix} = \begin{pmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} x_{old} \\ y_{old} \end{pmatrix} \tag{25}$$

The new (refined) CD-matrix under such a transformation is effectively given by:

$$\begin{aligned} CD_{refined} &= \begin{pmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} CD_{11} & CD_{12} \\ CD_{21} & CD_{22} \end{pmatrix} \\ &= \begin{pmatrix} \cos\gamma.CD_{11} + \sin\gamma.CD_{21} & \cos\gamma.CD_{12} + \sin\gamma.CD_{22} \\ \cos\gamma.CD_{21} - \sin\gamma.CD_{11} & \cos\gamma.CD_{22} - \sin\gamma.CD_{12} \end{pmatrix} \end{aligned} \tag{26}$$

Refined CD-matrix keywords are only written to an image header if their unrefined counterparts exist in the input header.

4.2.13 Outputs

There are six possible outputs from the software, five of which are optional. Items **(1)** and **(2)** below contain final results of the refinement. Items **(3), (4), (5) and (6)** 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 appended with a new set of keywords regardless if a refined solution was found or not. For images where no refinement was possible, the input CRVAL1, CRVAL2, CROTA2 keyword values are copied to their *new* refined counterparts 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 software is executed in "absolute-refinement" mode. The *new* refined CD-matrix keywords will only appear in the output header if their unrefined counterparts exist on input. The following is an example of the keywords written to each input header:

```
SOFTWARE= 'pointingrefine'     / Pointing refinement using pnt-src correlation
PTGVERSN=                  6.5 / Version number of pointingrefine program
RFNDFLAG=                    T / T if pointing was actually refined
RARFND  =     335.300477424704 / [deg] Refined RA
DECRFND =     63.2223413841496 / [deg] Refined DEC
CT2RFND =    -134.348482798526 / [deg] Refined CROTA2
PA_RFND =     134.348482798526 / [deg] Refined PA (= -CROTA2_refined)
ERARFND = 5.34765018912278E-06 / [deg] Error in refined RA
EDECRFND= 5.39444981203901E-06 / [deg] Error in refined DEC
EPA_RFND= 0.000805248989102741 / [deg] Error in refined PA or CROTA2
CSRDRFND= 3.87498210748186E-08 / [deg] Co-standard dev. in refined RA, DEC
NASTROM =                   40 / # Astrometric sources for absolute refinement
RARESID =   -0.240405370897228 / [arcsec] Residual: Observed-Refined RA
DECRESID=    0.508377061919418 / [arcsec] Residual: Observed-Refined DEC
PA_RESID=     0.83827648653596 / [arcsec] Residual: Observed-Refined PA
XC_RESID=    0.160152049269584 / [pix] Refinement correction in CRPIX1
YC_RESID=    -0.43191879675075 / [pix] Refinement correction in CRPIX2
EXCRESID=   0.0158423316097451 / [pix] Uncertainty in corr. for CRPIX1
EYCRESID=   0.0158409921786181 / [pix] Uncertainty in corr. for CRPIX2
CSDXCYC = -0.00147843602561188 / [pix] Co-standard dev. in ref. CRPIX1, CRPIX2
CD11RFND= 0.000236855933733227 / [deg/pix] Refined CD matrix element 1_1
CD12RFND= 0.000242434424919191 / [deg/pix] Refined CD matrix element 1_2
CD21RFND= 0.000242636737026603 / [deg/pix] Refined CD matrix element 2_1
CD22RFND= -0.000236982680365019 / [deg/pix] Refined CD matrix element 2_2
```

(2). Optionally (with the "–o" <*outfile*> option), a table in IPAC format listing FITS image names (with directory paths), refined RA, DEC, CROTA2 with uncertainties and co-standard deviations can be generated. This table contains additional keyword fields for use by the SSC's super-boresight refinement software. Ancillary information is included in the header of this table. The format and content of this table is based on the Software Interface Specification (SIS) SOSDL-SIS-PT-3005.

(3). Optionally (with the "–ot" <*outfile*> option), a text file listing the cartesian image offsets and uncertainties in the tangent reference image frame can be generated. An example is shown below.

```
\character Cartesian_Tangent_Shifts_File
\character From_Pointing_Refinement_Program = "pointingrefine", Version  6.00
\character Creation_Date_Time = Sun Dec  7 18:53:20 2003
\character Input_Image_List = ./testing/pointingrefine_imagesCD.list
\character Reference_Image (shifts are measured in) = image defined by fiducial frame
table (./testing/fiducial.tbl)
|Img |Theta     |XT        |YT       |Err_Theta |ErrXT     |ErrYT      |CovXY    |NASTROM
|
|int |real      |real      |real     |real      |real      |real       |real     |int
|
|null|degree    |pixel     |pixel    |degree    |pixel     |pixel      |pixel    |null
|
   1   0.030904   0.000007   0.001126   0.294616   0.005245   0.039297   1.9e-07   4
   2  -0.019685   0.000027   0.000658   0.304533   0.005245   0.039274   2.8e-07   3
   3  -0.033719  -0.000033   0.000268   0.297993   0.005245   0.039304   3.7e-07   0
   4   0.013485   0.000038  -0.004016   0.327261   0.007417   0.055322   4.6e-07   7
   5   0.007769   0.000037   0.000916   0.277941   0.005245   0.039286   5.5e-07   4
   6  -0.022380  -0.000023   0.001188   0.318355   0.005245   0.039347   6.4e-07   5
   7   0.077874   0.000006  -0.003052   0.357600   0.006424   0.048154   7.3e-07   5
   8   0.020536  -0.000012  -0.000190   0.234658   0.004283   0.032145   8.2e-07   4
   9   0.017612  -0.000040  -0.002542   0.337609   0.006424   0.048111   9.1e-07   7
```

(4). Ancillary information on which images could not be refined due to non-correlations, fraction of input images actually refined, average source separation in image frame before and after refinement, 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). This and additional information on all correlated image pairs is also written to standard output by specifying the verbose "–v" switch.

```
N.B. Image numbers below refer to order in input list.

pointingrefine_source_correlation: Number of images initially found correlated with
another image=9 (100.0% of total) (number ultimately refined if NO singular matrix
solution exists)

pointingrefine_compute_matrix: Fraction of non-zero elements in coefficient matrix
(measures the degree to which all images mutually overlap) = 49.0%

Average source-match separations (in reference frame pixels) before and after refinement:
AvgXYSepnBefRel 0.183506 ## Avg frame-to-frame sepn before absolute refinement
AvgXYSepnAftRel 0.171398 ## Avg frame-to-frame sepn after absolute refinement
ChiSquareMinRel 1.139250 ## Min. chi-square value for frame-to-frame matches
NumDegOfFreedomRel 73 ## Num. Degrees Freedom for frame-to-frame matches

AvgXYSepnBefAbs 0.244328 ## Avg frame-to-absolute sepn before absolute refinement
AvgXYSepnAftAbs 0.222374 ## Avg frame-to-absolute sepn after absolute refinement
ChiSquareMinAbs 1.342116 ## Min. chi-square value for frame-to-absolute matches
NumDegOfFreedomAbs 12 ## Num. Degrees Freedom for frame-to-absolute matches
```

```
AvgXYSepnBefTot 0.200571 ## Avg sepn betw'n _all_ matches before refinement
AvgXYSepnAftTot 0.185700 ## Avg sepn betw'n _all_ matches after refinement
ChiSquareMinTot 2.481365 ## Min. chi-square value for _all_ matches
NumDegOfFreedomTot 112 ## Num. Degrees Freedom for _all_ matches

Following values are in units of arc-seconds.

Image # | Inp-Refnd RA | Inp-Refnd DEC | Inp-Refnd CROTA2 | #Abs.Sources used
     1       0.039517      -0.125489      -162.101961            4
     2       0.142688      -0.062431      -110.048577            3
     3      -0.182572      -0.013206      -178.537078            0
     4       0.210271       0.393054      102.926218            7
     5       0.203518      -0.096975       -3.442259            4
     6      -0.126959      -0.119478      -126.846563            5
     7       0.025081       0.290696      -234.628361            5
     8      -0.067439       0.013425       19.759565            4
     9      -0.225542       0.258700      -44.094594            7

AveRA_Change 0.002062 ## Average residual in observed-refined RA (arcsec)
AveRA_Change_ABS 0.135954 ## Average residual in |observed-refined| RA (arcsec)
MaxRA_Change 0.225542 ## Maximum residual in |observed-refined| RA (arcsec)
AveDec_Change 0.059811 ## Average residual in observed-refined Dec (arcsec)
AveDec_Change_ABS 0.152606 ## Average residual in |observed-refined| Dec (arcsec)
MaxDec_Change 0.393054 ## Maximum residual in |observed-refined| Dec (arcsec)
AveCROTA2_Change -81.890401 ## Average residual in observed-refined CROTA2 (arcsec)
AveCROTA2_Change_ABS 109.153908 ## Average residual in |observed-refined| CROTA2 (arcsec)
MaxCROTA2_Change 234.628361 ## Maximum residual in |observed-refined| CROTA2 (arcsec)
PrcntRefinedImages 100.0 ## Percentage of input images refined (%)
AveNum_AbsSources   4.33 ## Average number of absolute sources used over all refined
images
```

(5). If the **"super-verbose"** (−vv) switch is specified, the following diagnostic information is written to standard output:

a. Extraction centroid errors of all correlated point sources in the reference image frame.

b. Coordinates of *unrefined* input pointings (CRVAL1, CRVAL2) in pixel coordinates of the reference image frame.

c. Matrix solver diagnostics.

(6). If the **"debug"** (−d) switch is specified, the following files (in italics) and corresponding diagnostics are generated:

a. "*pointingrefine_data.dump1*": If the software is run with the "–t 1" option (i.e. use *x,y* extractions only from input tables), this file will contain the input pixel coordinates of all sources from the source extraction tables and corresponding RA, DEC coordinates from the WCS transformation.

b. "*pointingrefine_data.dump2*": Contains a listing of all correlated point sources from all possible correlated image pairs (in input image pixel coordinates).

c. "*pointingrefine_data.dump3*": Coordinates of all correlated point sources on sky and in pixel coordinates of reference image frame.

d. "*pointingrefine_data.dump4*": Matched point-source separations in pixel frame of reference image between all image pairs before refinement. Column 1 = image pair (A:B) where A, B are image numbers in order from top of input image list; column 2 = X_separation; column 3 = Y_separation; column 4 = radial separation.

e. "*pointingrefine_data.dump5*": Matched point-source separations in pixel frame of reference image between all image pairs after refinement. (Same format as d. output).

f. "*pointingrefine_data.dump6*": Debug info. on twist angle refinement.

g. "*pointingrefine_matrix_coeffs.txt*": Lists all non-zero matrix elements and row locations (in Compressed Sparse Column or Harwell-Boeing format).

h. "*pointingrefine_colstr_array.txt*": Listing of row indices of first non-zero matrix elements for each column (in Compressed Sparse Column or Harwell-Boeing format).

i. "*pointingrefine_RHS_coeffs.txt*": Lists elements of right-hand-side vector B of the matrix equation $AX = B$.

j. "*covariance_matrix.txt*": Lists non-zero elements of the full error-covariance matrix for reference-frame offsets of all *correlated* images.

4.2.14 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 algorithm used by POINTINGREFINE has been adequately described in the previous section. As a detail, individual images which are NOT correlated with any others in the input list will have their offsets explicitly set to zero in the (mosaic) reference image plane: $\delta\theta^m = \delta X^m = \delta Y^m = 0$. Due to the lack of correlated point sources, such images cannot be refined and they are excluded from the "global minimization". However, if the input image list contains image clusters (sub-ensembles) which are disjoint from the nominal reference image frame (non-contiguous image-to-reference paths present), then a singular matrix from the global minimization will result. If this occurs in "absolute refinement mode", a second pass computation is performed and only those frames which contain absolute point sources are used. These become the only images refined. In "relative refinement" mode, the software will abort with a message sent to standard output. This remains a (albeit minor) lien of the current software.

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 user can specify the number of sources to use from each source-extraction table through the "Max_Num_Sources_Per_Image" parameter (command line option "-m") . The *brightest* "-m" sources are used from each extraction table if the total number of extractions exceeds the number specified by this parameter. The default is 100 sources per extraction table. The speed of the POINTINGREFINE software is found to critically depend on the number of sources used for each input image.

The user is also free to specify a constant systematic uncertainty to add to the measured random pointing uncertainties extracted from each FITS header for computing a-priori weights. This is given by the namelist/command-line parameter: "Apriori_Syst_Unc" (in arcsec) and is only used if a random measurement uncertainty (in RA, Dec, Twist) exists in a FITS header.

The following parameters are "hardcoded" in the include file *pointingrefine.h*. These should not be changed unless absolutely necessary: the maximum number of image files or table files allowed in the input lists: "MAX_NUMBER_IMAGES"; the "second-point" position lying along the vertical bisector joining pixel positions (CRPIX1,CRPIX2) and (CRPIX1,NAXIS2) in each image for computation of the refined twist-angle (see section 4.2.9): “TOPPOINT”; the closest permissible separation between any two distinct source matches between a frame pair (in order to be declared a “correlated image pair”): “CLOSEST_SOURCE_SEPN”; the number of iterations used by the linear sparse matrix solver for refining solutions due to accumulations in numerical round-off error: “NUM_ITER_SOLVE”, the three keywords corresponding to random measured uncertainties in R.A., Dec and Twist angle (CROTA2) and the position angle keyword “PA_KEYWORD”. Current settings are a follows:

```
#define MAX_NUMBER_IMAGES       5001
#define TOPPOINT                5  /* pixels */
#define CLOSEST_SOURCE_SEPN     5  /* pixels */
#define NUM_ITER_SOLVE          10 /* Number of iterations for sparse solver*/
#define RAERR_KEYWORD           "CRDER1"  /* Uncertainty keywrd for RA */
#define DECERR_KEYWORD          "CRDER2"  /* Uncertainty keywrd for Dec */
#define TWISTERR_KEYWORD        "UNCRTPA" /* Uncertainty keywrd for PA */
#define PA_KEYWORD              "PA"      /* Position-angle keywrd (degrees) */
```

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

```
Program pointingrefine, Version 6.4

Usage: pointingrefine
 -n  <inp_namelist_fname>        (Optional)
 -f1 <inp_image_list_fname>      (Required)
 -f2 <inp_table_list_fname>      (Required)
 -f3 <inp_fiducial_frame_table>  (Optional, Specified together with
                                  -f4 option; Default = Input frame with
                                  maximum no. of overlaps [=Reference Image])
 -f4 <inp_absolute_RA_DECs_fname>(Optional, Specified together with -f3 option)
 -fr <inp_FITS_image_ref_fname>  (Optional, Only for relative refinement mode;
                                  Default = Input frame with maximum no.
                                  of overlaps [=Reference Image])
 -o  <out_table_fname>           (Optional, Input headers always updated)
 -ot <out_tangentshifts_fname>   (Optional, Default = no file generated)
 -sr <Max_Search_Radius>         (Optional [arcsec], Default=5.0)
 -sd <Max_Flux_Diff>             (Optional [percent], Default=5.0; For
                                  frame-to-frame source matching)
 -sa <Max_Flux_Diff_Absolutes>   (Optional [percent], Default=100; For
                                  frame-to-absolute source matching)
 -sb <Abs_Flux_Scale_Factor>     (Optional, Scale input absolute
                                  fluxes with this factor; Default=1)
 -st <Flux_Threshold>            (Optional [source extractor units],
                                  Default=0)
```

```
-m  <Max_Num_Sources_Per_Image> (Optional, Use brightest "m" entries
                                 from each source extraction table if
                                 number exceeds "m"; Default=100)
-t  <Read_xy_Extractions?>      (Optional, Use x,y extractions directly?
                                 1=yes, 0=no (use RA,Dec instead), Default=1)
-b  <Use_Apriori_Unc?>          (Optional, Use measured ptg uncerts if exist
                                 in FITS headers; 1=yes, 0=no; Default=1)
-c  <Use_Only_Absolutes?>       (Optional, Use only absolute matches; need
                                 -f3 and -f4 above; 1=yes, 0=no; Default=0)
-e  <Only_Refine_XY_Translations>(Optional, Only refine orthogonal
                                  translations; 1=yes, 0=no; Default=0)
-g  <Min_Source_Matches_Per_Pair>(Optional, Min. required num. sources to
                                  match in an image pair; Default=2)
-u  <Apriori_Syst_Unc>          (Optional [arcsec], Add to ptg uncerts
                                 if they exist; Default=0)
-w  <Min_Tolerable_Uncert>      (Optional [arcsec], Minimum tolerable
                                 apriori random uncert; Default=0)
-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. POINTINGREFINE assumes that each FITS image in the 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.

B. Every image in the 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 the input list (namelist parameter: FITS_Image_List_Filename) are listed one per line and do not have to be in any specific order. Source extraction tables in the input list (namelist parameter: Source_Table_List_Filename) must have a one-to-one correspondence with respective images in the FITS image list.

D. It is recommended that the source extraction tables contain the brightest sources from each image in order to avoid large centroiding errors and hence optimize source-matching. If this is not possible, input source extraction fluxes can be filtered according to the flux threshold parameter: “Flux_Threshold”.

E. The maximum number of images allowed in the input list is currently 5000. This is defined by the MAX_NUMBER_IMAGES parameter in the include file *pointingrefine.h*.

F. Input image headers are always updated with *new* refined pointing keywords and uncertainties. Every image header is appended with a new set of keywords regardless if a refined solution was found or not. For images where no refinement was possible, the input CRVAL1, CRVAL2, CROTA2 keyword values are copied to their *new* refined counterparts to maintain a consistent set of pointing keywords for use in downstream ensemble processing.

G. To execute the POINTINGREFINE software in “absolute” refinement mode, *both* filename parameters: “Fiducial_Frame_Table” and “Absolute_RA_Decs” must be specified. The program will abort with a message written to standard output if only one of these is specified. For “relative” refinement” mode, these filename parameters must be omitted from the namelist or command-line. In this case, the refined pointings become relative to the input image which has the maximum number of paired-correlations with other images in the input list.

H. In “absolute” refinement mode, the fiducial frame table file (namelist parameter Fiducial_Frame_Table) and absolute source list file (namelist parameter Absolute_RA_Decs) must be made before hand. The absolute source list consists of all astrometrically known sources (e.g. from the 2MASS catalog) distributed within the bounding-box defined by the fiducial-frame-table parameters.

I. It assumed that extracted source positions, either in (x, y) pixel coordinates, or, in (RA, Dec) have been corrected for optical distortion effects. If RA, Dec are to be used from the source extraction tables directly (command-line option “-m 0”) as generated by the “sourcestimate” software, then these have automatically been corrected for optical distortion in the WCS transformation. If x, y are to be read directly, it must be ensured that the source extractions were performed on images already corrected for optical distortion.

J. The pointing (random) uncertainty keywords as defined by the string variables RAERR_KEYWORD, DECERR_KEYWORD and TWISTERR_KEYWORD for RA, Dec and CROTA2 respectively in *pointingrefine.h* must be present in each FITS header if one wants to use them as a-priori measurement weights in the global minimization calculation (section 4.2.7). If either of these keywords are not present or have a value of zero, the a-priori weights will be explicitly set to zero (or excluded from the computation). If these keywords exist in the FITS header, the user can opt whether to use these or not via the namelist/command-line parameter "Use_Apriori_Unc". Furthermore, the user can optionally specify an additional systematic uncertainty to add onto the random uncertainties via the namelist parameter "Apriori_Syst_Unc", and, a mimimum tolerable random uncertainty below which input random uncertainties are redeemed questionable (namelist parameter: "Min_Tolerable_Apriori_Uncert"). If below this specified value for *either RA, Dec or CROTA2*, the random uncertainty is set equal to this value in computations.

K. Refined CD-matrix keywords are only written to an output header if their unrefined counterparts exist on input.

L. Values of keywords specified by RAERR_KEYWORD, DECERR_KEYWORD and TWISTERR_KEYWORD in *pointingrefine.h* are all expected to be in units of degrees.

6. Output

6.1. POINTINGREFINE Output

The output generated by POINTINGREFINE was outlined in depth in *section 4.2.11*. To summarize, POINTINGREFINE 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 are written to each input FITS header. These are updated with each subsequent execution of the pointingrefine software on the same input images.

C.) Optionally, an IPAC table file of refined pointing keyword values and uncertainties and a text file listing Cartesian shifts and uncertainties in the (mosaic) reference image frame.

D.) Optionally, a QA log file listing *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) diagnostic information on intermediate steps during processing written to text files. See section 4.2.11 for details.

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

6.1.1 POINTINGREFINE 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.

```
Allocating at least 0.144000 MB of memory...

pointingrefine_source_correlation:  Number of images initially found correlated
with another image=9 (100.0% of total) (number ultimately refined if NO singular
matrix solution exists)

Program pointingrefine, Version 6.4
Namelist File  = pointingrefine.nl
Input image list file = ./testing/pointingrefine_imagesCD.list
Input table list file = ./testing/pointingrefine_src_tbls.list
Input fiducial frame table = ./testing/fiducial.tbl
Input Absolute RA, DEC source list = ./testing/absolute_ra_decs.list
Output Table file (refined positions) = ./testing/refined.tbl
Output Cartesian tangent shifts file = ./testing/tangentshifts.txt
Input Search Radius (arcsec) = 5.000000
Input Max. Flux Difference for frame-to-frame source matching (percent) =
5.000000
Input Max. Flux Difference for frame-to-absolute source matching (percent) =
99.990000
Input absolute flux scale factor = 1.000000
Input flux threshold for source matching = 0.000000
Used the following "brightest" entries from each extraction table: 50
Min. required number of sources to match between image pairs = 2
Read_xy_Extractions flag (0=no,1=yes) = 1
Use_Apriori_Unc flag (0=no,1=yes) = 1
Only_Refine_XY_Translations flag (0=no,1=yes) = 0
Use_Only_Absolutes flag (0=no,1=yes) = 0
Input a-priori systematic pointing uncertainty (arcsec) = 2.500000
Input minimum tolerable random pointing uncertainty (arcsec) = 0.200000
Ancillary Data-File Path = ../pointingrefine_v7
```

```
Verbose flag = 0
Super-verbose flag = 0
Debug flag = 0
QA flag = 0
Performed ABSOLUTE pointing refinement computation.
Program pointingrefine: Status Message: 0x0000
Normal exit from Function 0x0000: LOG_WRITER
Processing time: 0.420000 seconds
Current date/time: Fri Mar 26 17:53:37 2004
Program pointingrefine, version 6.4, terminated successfully.
```

7. Testing

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

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

1. Tested POINTINGREFINE on a list of 800 simulated IRAC images (~1.2″/pixel with sizes 5′×5′) using a truth source list provided by D. Shupe. Pointing keyword values were modified using a distribution of pointing uncertainties expected for SIRTF frames. The refined pointings (generated by the pointing refinement software) were compared with simulated, truth values and the refinement (or improvement) was found to be at least 80% in almost all images. The plot below shows the distribution of radial offsets between input raw images (with pointing uncertainties) and simulated ("truth" images) before and after pointing refinement. There were 564 absolute (truth) sources randomly distributed amongst the 800 input images (contained within the fiducial image frame). Each input frame had ≤ 50 extracted sources with centroid errors $|\Delta x|$ and $|\Delta y| \leq 0.5$ pixels.

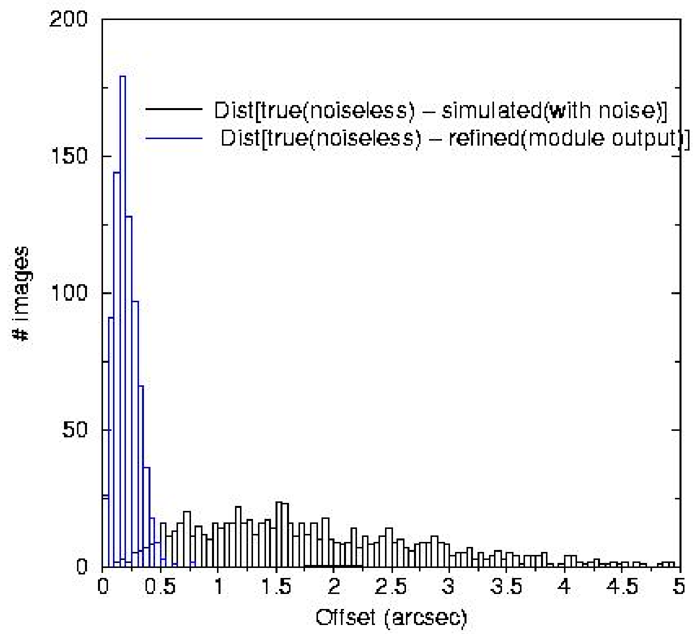


Figure 6. IRAC absolute refinement simulation (800 images).

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

4. Executed POINTINGREFINE on non-square images.

5. Executed POINTINGREFINE on lists of ≈1000 FITS images to test for memory/speed limitations. Both the number of images in the input list and number of sources read per image (MAX_NUM_ENTRIES below) were varied.

8. Usage Examples

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

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

Without using a namelist file in "relative" refinement mode with an output IPAC table of refined pointing keyword values and QA log file generated. A maximum of fifty sources are read from the input source extraction tables ("–m 50" option) and *x, y* pixel positions of extracted sources are used (default for –t option):

```
pointingrefine –f1 image_list.txt –f2 table_list.txt –o refined.tbl
–sr 5.0 –sd 5.0 –st 0 –m 50 –a /anc_path –qa –v
```

Without using a namelist file in "absolute" refinement mode with an output IPAC table of refined pointing keyword values and QA log file generated. A maximum of fifty sources are read from the input source extraction tables (–m option) with RA, Dec positions of extracted sources used ("–t 0" option), an a-priori systematic pointing uncertainty as specified by the "-u" option, and, a minimum tolerable random uncertainty specifed by "-w".

```
pointingrefine –f1 image_list.txt –f2 table_list.txt –f3
fiducial.tbl –f4 absolutes.txt –o refined.tbl –sr 5.0 –sd 5.0 –st 0
–m 50 –t 0 –u 1.0 –w 0.2 –a /anc_path –qa –v
```

If single-image absolute refinement is desired (i.e. only use frame-to-absolute matches), append a "–b 0 –c 1" to the previous command-line specification.

9. Glossary

DCE	Data Collection Event
DN	Data Number
FITS	Flexible Image Transport System
ICRS	International Celestial Reference System

IOC	In-Orbit Checkout
IRAF	Image Reduction and Analysis Facility
PA	Position Angle
SDS	Subsystem Design Specification
SIS	Software Interface Specification
TBD	To Be Determined
TBR	To Be Resolved
WCS	World Coordinate System

10. Appendix I

In this section we give expressions for measurement uncertainties in RA, Dec and Twist angle (CROTA2) for an image, transformed into a Cartesian reference frame. These are used in computation of the a-priori (inverse variance) weight terms as defined by Equation (8) and matrix elements derived therefrom (section 4.2.8).

The figure below shows a pointing error ellipse of an image on the sky with RA, Dec axes in red and a superimposed reference image (black grid) with axes *X* and *Y* rotated clockwise by an amount θ - the image twist angle. Our aim is to transform errors along the RA and Dec axes into equivalent errors along *X* and *Y* of the reference image.

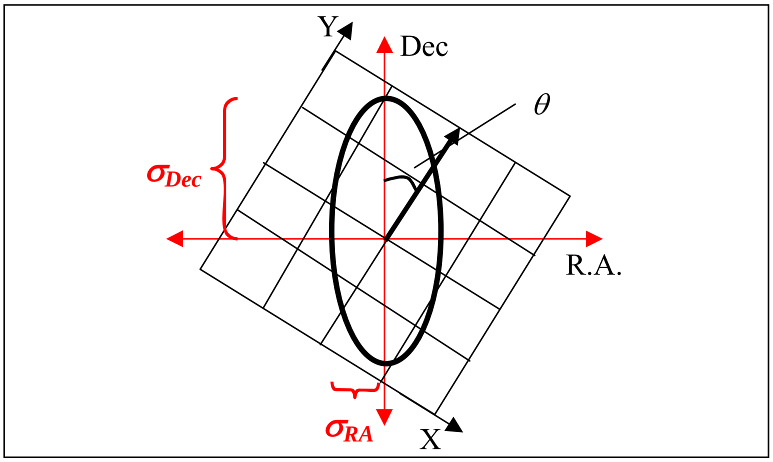


Figure 7. Schematic showing sky error ellipse projected onto Cartesian plane

Given pointing errors for an image along the RA and Dec axes as shown in Figure 7, we first define the quantity:

$$\sigma = \left(\frac{\sigma_{Dec}}{\sigma_{RA}}\right)^2 .$$

From simple trigonometry and using the equation of a "rotated ellipse" leads us to the following expressions for uncertainties along X and Y in pixel units of the rotated reference image frame:

$$\sigma_X = \frac{\sigma_{Dec}}{p_x}\left\{\sigma\left[\cos\theta - \frac{(\sin^2\theta\cos\theta - \sigma\sin^2\theta\cos\theta)}{\sigma\sin^2\theta + \cos^2\theta}\right]^2 + \left[\sin\theta - \frac{(\cos^2\theta\sin\theta - \sigma\cos^2\theta\sin\theta)}{\sigma\sin^2\theta + \cos^2\theta}\right]^2\right\}^{-1/2}$$

$$\sigma_Y = \frac{\sigma_{Dec}}{p_y}\left\{\sigma\left[\frac{(\sigma\cos^2\theta\sin\theta - \cos^2\theta\sin\theta)}{\sigma\cos^2\theta + \sin^2\theta} - \sin\theta\right]^2 + \left[\frac{(\sigma\sin^2\theta\cos\theta - \sin^2\theta\cos\theta)}{\sigma\cos^2\theta + \sin^2\theta} + \cos\theta\right]^2\right\}^{-1/2},$$

where p_x and p_y represent pixel scales in the reference image frame (say in degrees per pixel). In WCS FITS keyword terminology, $p_x \equiv$ CDELT1 and $p_y \equiv$ CDELT2 .

To good approximation, we assume that the rotational uncertainty of an image in another *rectilinear* reference image frame is equal to the uncertainty in measured twist angle (CROTA2). This is reasonably accurate at the equator, but at the poles, the twist angle is strongly correlated with R.A. Close to the poles, one can easily induce an uncertainty in twist angle by virtue of an uncertainty in RA. Nonetheless, this assumption will lead us to (conservatively) *overestimate* the corresponding uncertainty in the reference image frame. We therefore assume:

$$\sigma_\theta \approx \sigma(\mathrm{CROTA2})\ .$$

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