

# Pointing Refinement of SIRTf Images

F. J. Masci, 08/14/2002

This document outlines an algorithm for refining the celestial pointings of SIRTf images for purposes of performing robust image coaddition, mosaic generation and an eventual estimate of point source positions therein. The proposed algorithm can refine positions in either a “relative” sense where pointings depend on the arrangement of images in a mosaic relative to an input frame, or, in an absolute sense where absolute point source information from a catalog is used.

## I. Initialization and set-up

All possible correlations amongst input images are found by matching point sources in an absolute coordinate system using both position and flux matching (in the same band). The user can specify a search radius, flux matching interval and flux threshold. It is assumed that the input images are first corrected for optical distortion prior to carrying out the pixel to sky transformation.

## II. Definition of Reference/Fiducial 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:

If NO absolute point source information (known catalog) is available (i.e. only relative pointing refinement is desired), there are two possible choices for the reference frame:

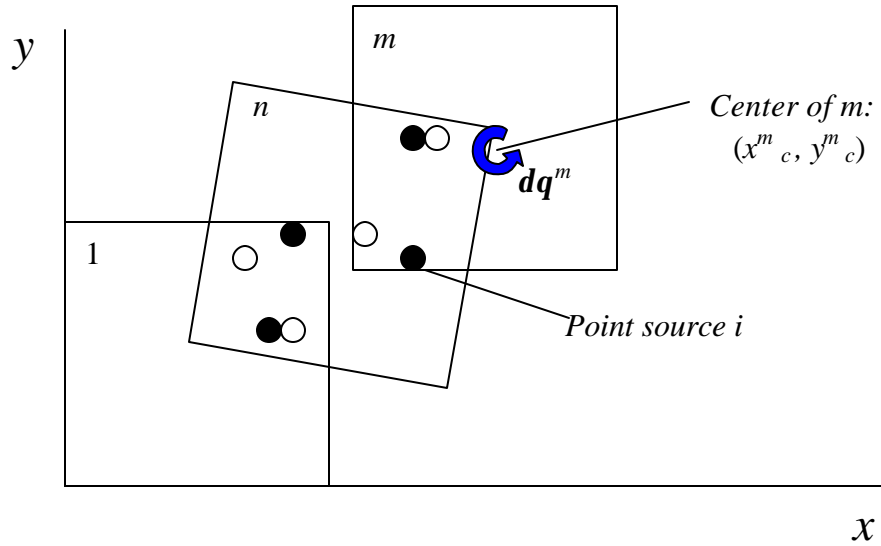
1. The image in the input list which has the maximum number of overlaps with other images, or,
2. The image which is closest to the center of a “fiducial frame” defining the entire mosaic.

The former would be more robust since it 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).

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.

### III. 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 1: 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  $\delta \mathbf{q}^m$  and orthogonal translations  $\delta X^m$  and  $\delta Y^m$ . 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:

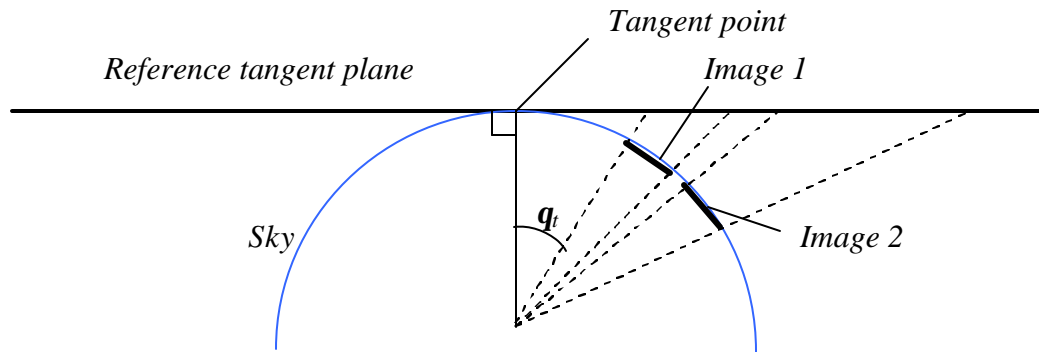
$$\begin{pmatrix} x_i^m \\ y_i^m \end{pmatrix} \rightarrow \begin{pmatrix} \tilde{x}_i^n \\ \tilde{y}_i^n \end{pmatrix} = \begin{pmatrix} x_c^m \\ y_c^m \end{pmatrix} + \begin{pmatrix} \cos \mathbf{d}q^m & -\sin \mathbf{d}q^m \\ \sin \mathbf{d}q^m & \cos \mathbf{d}q^m \end{pmatrix} \begin{pmatrix} x_i^m - x_c^m \\ y_i^m - y_c^m \end{pmatrix} + \begin{pmatrix} \mathbf{d}X^m \\ \mathbf{d}Y^m \end{pmatrix} \quad (1)$$

where  $(x_c^m, y_c^m)$  are coordinates of the center of image  $m$  and  $\delta \mathbf{q}^m$  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  $< 10^\circ$  with image sizes  $< 7'$  is a good working measure. At distances  $\theta_t > 5^\circ$  from the mosaic tangent point, the difference in the projected sizes of two adjacent  $7' \times 7'$  images is  $\approx 1.5\%$  and increases as  $\approx \tan(\theta_t)$ . Figure 2 shows the projection geometry.



**Figure 2: 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  $\delta \mathbf{q}^m$  by assuming

$$d\mathbf{q}^n \approx 0 \Rightarrow \sin d\mathbf{q}^m \approx d\mathbf{q}^m \text{ and } \cos d\mathbf{q}^m \approx 1 . \quad (2)$$

Equation (1) can therefore be re-written as the pair of equations:

$$x_i^m \rightarrow \tilde{x}_i^n = x_i^m - (y_i^m - y_c^m) d\mathbf{q}^m + dX^m \quad (3)$$

$$y_i^m \rightarrow \tilde{y}_i^n = y_i^m + (x_i^m - x_c^m) d\mathbf{q}^m + dY^m , \quad (4)$$

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:

$$L = \sum_{m,n} \sum_i \left\{ \frac{1}{\Delta x_i^{m,n}} [\tilde{x}_i^n - \tilde{x}_i^m]^2 + \frac{1}{\Delta y_i^{m,n}} [\tilde{y}_i^n - \tilde{y}_i^m]^2 \right\} \quad (5)$$

where

$$\begin{aligned} \Delta x_i^{m,n} &= \mathbf{s}^2(x_i^m) + \mathbf{s}^2(x_i^n) \\ \Delta y_i^{m,n} &= \mathbf{s}^2(y_i^m) + \mathbf{s}^2(y_i^n) \end{aligned} \quad (6)$$

and the  $\mathbf{s}^2$  represent variances in extracted point source positions.

The outer sum in equation (5) is over all correlated image pairs  $(m, n)$  including the reference image, while the inner sum is over all correlated point sources  $i$  which belong in the overlap region of image pair  $(m, n)$ .

Equation (5) can be re-written in terms of physical image offsets  $\delta \mathbf{q}$ ,  $\delta X$  and  $\delta Y$  for each image by use of equations (3) and (4):

$$\begin{aligned} L = \sum_{m,n} \sum_i \left\{ \frac{1}{\Delta x_i^{n,m}} \left[ x_i^m - (y_i^m - y_c^m) \mathbf{d}\mathbf{q}^m + \mathbf{d}X^m - x_i^n + (y_i^n - y_c^n) \mathbf{d}\mathbf{q}^n - \mathbf{d}X^n \right]^2 + \right. \\ \left. \frac{1}{\Delta y_i^{n,m}} \left[ y_i^m + (x_i^m - x_c^m) \mathbf{d}\mathbf{q}^m + \mathbf{d}Y^m - y_i^n - (x_i^n - x_c^n) \mathbf{d}\mathbf{q}^n - \mathbf{d}Y^n \right]^2 \right\} \quad (7) \end{aligned}$$

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

$$\frac{\partial L}{\partial \mathbf{d}\mathbf{q}^m} = 0, \quad \frac{\partial L}{\partial \mathbf{d}X^m} = 0, \quad \frac{\partial L}{\partial \mathbf{d}Y^m} = 0 \quad (8)$$

Evaluating the partial derivatives in (8) for image  $m$  (an arbitrary image in our mosaic), leads us to the following:

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

$$\frac{\partial L}{\partial \mathbf{d}X^m} = \sum_n \sum_i \frac{2}{\Delta x_i^{n,m}} [x_i^m - (y_i^m - y_c^m) \mathbf{d}\mathbf{q}^m + \mathbf{d}X^m - x_i^n + (y_i^n - y_c^n) \mathbf{d}\mathbf{q}^n - \mathbf{d}X^n] \quad (\text{B})$$

$$\frac{\partial L}{\partial \mathbf{d}Y^m} = \sum_n \sum_i \frac{2}{\Delta y_i^{n,m}} [y_i^m + (x_i^m - x_c^m) \mathbf{d}\mathbf{q}^m + \mathbf{d}Y^m - y_i^n - (x_i^n - x_c^n) \mathbf{d}\mathbf{q}^n - \mathbf{d}Y^n] \quad (\text{C})$$

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  $\delta \mathbf{q} = 0$ ,  $\delta X = 0$  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 (7), 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  $m$ .

## IV. Transformation of Point Source Positional Uncertainties

It is important to note that the uncertainties which occur in the denominators of the above equations (defined in equation 6) need to be transformed along with individual source positions to the mosaic (reference) tangent plane. Due to projection effects when one veers away from the tangent point (see Figure 2), 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  $(\Delta x, \Delta y)$  of a source detected at position  $(x, y)$  in the frame of any input image (read from a source extraction table), we first compute the maximum and minimum extents of the source error ellipse in both orthogonal directions. Second, these extents are transformed to the *sky* using standard WCS transformations. And last, these maximum and minimum extents, which span the “error ellipse” on the *sky* are transformed to pixel coordinates in the reference image frame:  $(x_{maxref}, y_{maxref})$  and  $(x_{minref}, y_{minref})$  in the following representation:

$$\begin{aligned}
\left. \begin{aligned} x_{\max} &= x + \Delta x \\ y_{\max} &= y + \Delta y \end{aligned} \right\} & \xrightarrow{\text{PIX2WCS}} \mathbf{a}_{\max}, \mathbf{d}_{\max} & \xrightarrow{\text{WCS2PIX}} x_{\max \text{ ref}}, y_{\min \text{ ref}} \\
\left. \begin{aligned} x_{\min} &= x - \Delta x \\ y_{\min} &= y - \Delta y \end{aligned} \right\} & \xrightarrow{\text{PIX2WCS}} \mathbf{a}_{\min}, \mathbf{d}_{\min} & \xrightarrow{\text{WCS2PIX}} x_{\min \text{ ref}}, y_{\min \text{ ref}}
\end{aligned} \tag{9}$$

Finally, the position errors in the *mosaic reference image frame* can be computed from the new maximum and minimum extents:

$$\begin{aligned}
\Delta x_{\text{ref}} &= \frac{1}{2} |x_{\max \text{ ref}} - x_{\min \text{ ref}}| \\
\Delta y_{\text{ref}} &= \frac{1}{2} |y_{\max \text{ ref}} - y_{\min \text{ ref}}|.
\end{aligned} \tag{10}$$

It's important to note that no covariance information (or possible orientation of the error ellipse) between the  $\Delta x$  and  $\Delta y$  axes is accounted for in the above method. This will allow us to be conservative and overestimate positional errors in the reference frame and sky. However, our refinement will not be as good under this assumption since this will allow more flexibility in source matches due to larger overlapping error regions. Nonetheless, the error-ellipses in the source extractions are circular enough that this should make little difference.

## V. Solving for Offsets in the Mosaic Cartesian Plane

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

$$\begin{aligned}
A_q^m, A_X^m, A_Y^m, A_q^n, A_X^n, A_Y^n, \text{ constant term} &= \Psi_A(m, n) \\
B_q^m, B_X^m, B_q^n, B_X^n, \text{ constant term} &= \Psi_B(m, n) \\
C_q^m, C_Y^m, C_q^n, C_Y^n, \text{ constant term} &= \Psi_C(m, n)
\end{aligned} \tag{12}$$

Using the conditions defined in (8), these coefficients and terms are explicitly given by:

$$\begin{aligned}
A_q^m &= \sum_n \sum_i \frac{(y_i^m - y_c^m)^2}{\Delta x_i^{n,m}} + \frac{(x_i^m - x_c^m)^2}{\Delta y_i^{n,m}} \\
A_X^m &= -\sum_n \sum_i \frac{(y_i^m - y_c^m)}{\Delta x_i^{n,m}} \\
A_Y^m &= \sum_n \sum_i \frac{(x_i^m - x_c^m)}{\Delta y_i^{n,m}} \\
A_q^n &= -\sum_i \frac{(y_i^m - y_c^m)(y_i^n - y_c^n)}{\Delta x_i^{n,m}} + \frac{(x_i^m - x_c^m)(x_i^n - x_c^n)}{\Delta y_i^{n,m}} \\
A_X^n &= \sum_i \frac{(y_i^m - y_c^m)}{\Delta x_i^{n,m}} \\
A_Y^n &= -\sum_i \frac{(x_i^m - x_c^m)}{\Delta y_i^{n,m}} \\
\Psi_A(m, n) &= -\sum_n \sum_i \frac{(y_i^m - y_c^m)(x_i^n - x_c^n)}{\Delta x_i^{n,m}} + \frac{(x_i^m - x_c^m)(y_i^m - y_c^m)}{\Delta y_i^{n,m}} \\
B_q^m &= -\sum_n \sum_i \frac{(y_i^m - y_c^m)}{\Delta x_i^{n,m}} \\
B_X^m &= \sum_n \sum_i \frac{1}{\Delta x_i^{n,m}} \\
B_q^n &= \sum_i \frac{(y_i^n - y_c^n)}{\Delta x_i^{n,m}} \\
B_X^n &= -\sum_i \frac{1}{\Delta x_i^{n,m}} \\
\Psi_B(m, n) &= -\sum_n \sum_i \frac{(x_i^m - x_c^m)}{\Delta x_i^{n,m}} \\
C_q^m &= \sum_n \sum_i \frac{(x_i^m - x_c^m)}{\Delta y_i^{n,m}} \\
C_Y^m &= \sum_n \sum_i \frac{1}{\Delta y_i^{n,m}}
\end{aligned}$$

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

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

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

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 (10) can be represented as:

$$\begin{pmatrix} A_q^{m=m_1} & A_X^{m=m_1} & A_Y^{m=m_1} & A_q^{n=m_2} & A_X^{n=m_2} & A_Y^{n=m_2} \\ B_q^{m=m_1} & B_X^{m=m_1} & 0 & B_q^{n=m_2} & B_X^{n=m_2} & 0 \\ C_q^{m=m_1} & 0 & C_Y^{m=m_1} & C_q^{n=m_2} & 0 & C_Y^{n=m_2} \\ A_q^{n=m_1} & A_X^{n=m_1} & A_Y^{n=m_1} & A_q^{m=m_2} & A_X^{m=m_2} & A_Y^{m=m_2} \\ B_q^{n=m_1} & B_X^{n=m_1} & 0 & B_q^{m=m_2} & B_X^{m=m_2} & 0 \\ C_q^{n=m_1} & 0 & C_Y^{n=m_1} & C_q^{m=m_2} & 0 & C_Y^{m=m_2} \end{pmatrix} \begin{pmatrix} d\mathbf{q}^{m_1} \\ d\mathbf{X}^{m_1} \\ d\mathbf{Y}^{m_1} \\ d\mathbf{q}^{m_2} \\ d\mathbf{X}^{m_2} \\ d\mathbf{Y}^{m_2} \end{pmatrix} = \begin{pmatrix} \Psi_A(m_1, n) \\ \Psi_B(m_1, n) \\ \Psi_C(m_1, n) \\ \Psi_A(m_2, n) \\ \Psi_B(m_2, n) \\ \Psi_C(m_2, n) \end{pmatrix} \quad (13)$$

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  $\approx 22\%$  (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  $n = \text{reference}$  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:

$$N(\text{min non - zeros}) = 7(N_{corr} - 1)$$

$$N(\text{max non - zeros}) = 7(N_{corr} - 1)^2, \quad (14)$$

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:  $\delta \mathbf{q}^m = \delta \mathbf{X}^m = \delta \mathbf{Y}^m = 0$ . 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).

## VI. Offset Uncertainties in the Mosaic Cartesian Plane

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  $AA^{-1} = I$ .

## VII. Refinement of Celestial Pointings

Once the offsets of every mutually correlated image  $m$  ( $\delta q^m$ ,  $\delta X^m$ ,  $\delta Y^m$ ) are computed, we correct the tangent points (usually image centers in reference image coordinates – i.e.  $x_c^m$ ,  $y_c^m$  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:

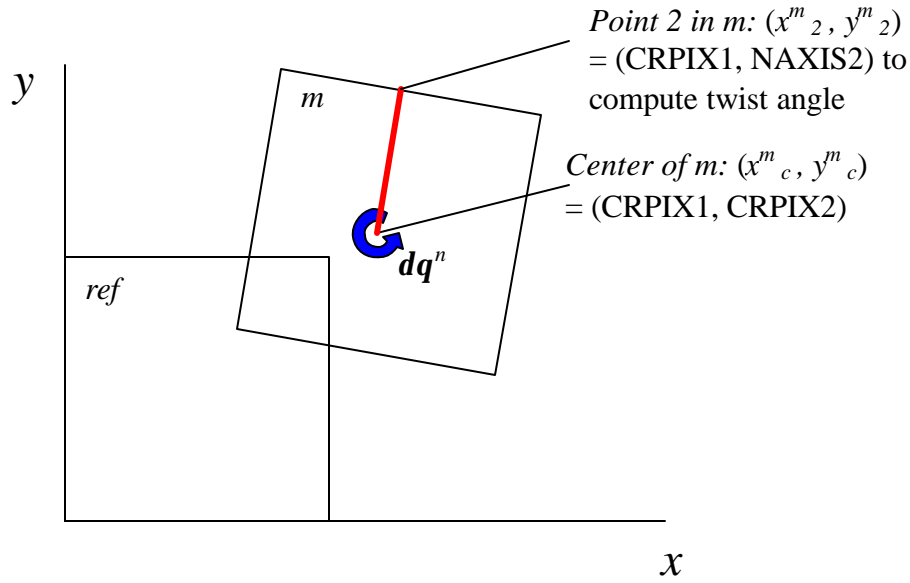
$$x_c^m(\text{new}) = x_c^m(\text{old}) + dX^m \quad (15)$$

$$y_c^m(\text{new}) = y_c^m(\text{old}) + dY^m \quad (16)$$

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  $\delta\mathbf{q}$ . 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 3: 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  $\Delta ABC$  forms a spherical triangle with sides  $d_{AB}$ ,  $d_{BC}$  and  $d_{AC}$ . The angle  $\gamma$  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:

$$\frac{\sin \mathbf{g}}{\sin d_{AB}} = \frac{\sin |\mathbf{a}_c - \mathbf{a}_2|}{\sin d_{BC}} \quad (17)$$

$$\Rightarrow \mathbf{g} = \sin^{-1} \left[ \frac{\sin d_{AB}}{\sin d_{BC}} \sin |\mathbf{a}_c - \mathbf{a}_2| \right] \quad (18)$$

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:

$$d_{AB} = \cos^{-1} [\sin \mathbf{d}_2] \quad (19)$$

$$d_{BC} = \cos^{-1}[\sin d_2 \sin d_c + \cos d_2 \cos d_c \cos(\alpha_2 - \alpha_c)]. \quad (20)$$

Care must be taken when computing  $\gamma$  from equation (18) since  $\gamma$  is usually defined to lie within  $0 \leq \gamma \leq 360^\circ$  and will need to be re-scaled for declinations  $< \delta_c$  and whether  $\alpha_2 < \alpha_c$  or  $\alpha_2 > \alpha_c$ .

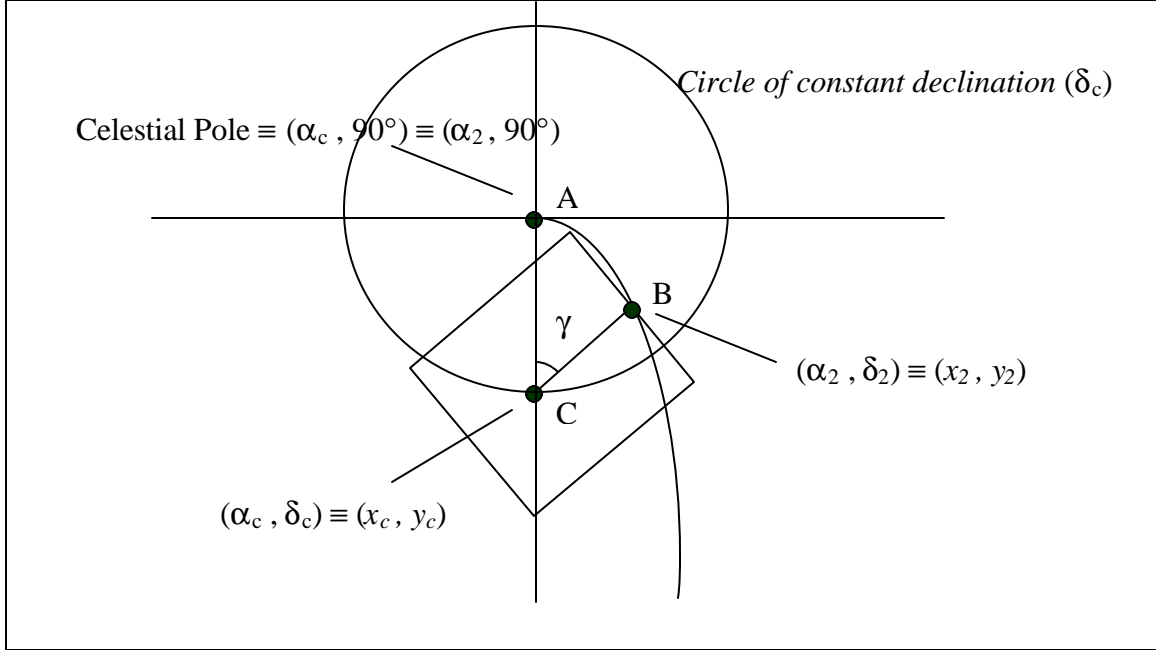


Figure 4: Schematic used in the derivation of the sky twist angle.

## VIII. Uncertainties in Refined Pointings

The reverse procedure of transforming positional uncertainties on the sky to pixel coordinates in the reference frame (section IV) is used. Now, we wish to transform offset uncertainties in the reference image frame to the sky after positions have been refined. However, the only difference in the method is when the maximum and minimum extents in RA are computed, the distance formula for two points lying along a great circle must be used, not the actual differences in RA as was computed using equation (10) in the image frame. This is due to the  $\cos(\delta)$  dependence in the representation of RA differences as arc-length measures, which becomes strongest near the poles.

The following is an algebraic representation of the process. Let  $(x_c, y_c)$  be the pixel coordinates of the “refined” pointing centers of an input image in the reference frame as computed from Equations (15) and (16). Then we transform the maximum and minimum extents to the sky:

$$\begin{array}{l}
x_{\max} = x_c + \Delta x_c \\
y_{\max} = y_c + \Delta y_c
\end{array}
\left. \vphantom{\begin{array}{l} x_{\max} \\ y_{\max} \end{array}} \right\} \xrightarrow{\text{PIX 2WCS}} \mathbf{a}_{\max}, \mathbf{d}_{\max}$$

$$\begin{array}{l}
x_{\min} = x_c - \Delta x_c \\
y_{\min} = y_c - \Delta y_c
\end{array}
\left. \vphantom{\begin{array}{l} x_{\min} \\ y_{\min} \end{array}} \right\} \xrightarrow{\text{PIX 2WCS}} \mathbf{a}_{\min}, \mathbf{d}_{\min}$$
(21)

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

$$\Delta \mathbf{a}_c = \frac{1}{2} \cos^{-1} \left[ \sin^2 \mathbf{d}_c + \cos^2 \mathbf{d}_c \cos(\mathbf{a}_{\max} - \mathbf{a}_{\min}) \right]$$

$$\Delta \mathbf{d}_c = \frac{1}{2} \left| \mathbf{d}_{\max} - \mathbf{d}_{\min} \right|.$$
(22)

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  $|\delta| < 50^\circ$  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.

## IX. 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 updated with a new set of keywords regardless if a refined solution existed or not. For images where no refinement was possible, the input CRVAL1, CRVAL2, CROTA2 keywords are reproduced to maintain a consistent set of pointing keywords for use in downstream ensemble processing. The NASTROM keyword represents the number of absolute sources used in the refinement and is only present if the software is executed in “absolute-refinement” mode. The following is an example of the keywords written to each input header:

```

SOFTWARE= 'pointingrefine' / Pointing refinement using pnt-src correlation
PTGVERSN= 2. / Version number of pointingrefine program
RARFND = 159.038955629263 / [deg] Refined RA pertaining to mosaic
DECRFND = 59.2056788625632 / [deg] Refined DEC pertaining to mosaic
CT2RFND = 0.456492264476405 / [deg] Refined CROTA2 pertaining to mosaic
ERARFND = 5.88717016082167E-05 / [deg] Error in refined RA
EDECRFND= 5.49725141212321E-05 / [deg] Error in refined DEC
ECT2RFND= 0.0905610226630317 / [deg] Error in refined CROTA2
NASTROM = 2 / # Astrometric sources for absolute refinement

```

(2). Optionally (with the “-o” <outfile> option), a table in IPAC format listing FITS image names (with directory paths), refined RA, DEC, CROTA2 values and uncertainties can be generated. Ancillary information is included in the header of this table. An example is shown below.

```

\character Pointing_Refinement_Program = "pointingrefine", Version 2.00
\character Creation_Date_Time = Thu Aug 15 09:27:10 2002
\character Input_Image_List = ./radectest/test.fits
\character Input_Source_Table_List = ./radectest/test.tbl
\character Input_Fiducial_Frame_Table = fiducial_800.tbl
\character Input_Absolute_RA_DEC_List = absolute_source_list.txt_test5
\character Reference_Image (pointings are relative to) = image defined by fiducial frame table
(fiducial_800.tbl)
\character RA = Refined right ascension of CRPIX1,CRPIX2
\character DEC = Refined declination of CRPIX1,CRPIX2
\character CROTA2 = Refined twist angle measured East from North on sky
\character sigma_RA = Uncertainty in refined right ascension
\character sigma_DEC = Uncertainty in refined declination
\character sigma_CROTA2 = Uncertainty in refined twist angle
\integer Number_of_Frames = 10
|Index|Filename|RA|DEC|CROTA2|sigma_RA|sigma_DEC|sigma_CROTA2|
|int|char|real|real|real|real|real|real|
|null|null|degree|degree|degree|degree|degree|degree|
1 IRAC.1.1001.fits 158.998642 59.156143 0.499386 0.000054 0.000085 0.102698
2 IRAC.1.2001.fits 159.036918 59.165245 0.399818 0.000044 0.000061 0.104464
3 IRAC.1.3001.fits 158.946091 59.188117 0.594639 0.000046 0.000125 0.104017
4 IRAC.1.4001.fits 158.974626 59.194462 0.476763 0.000051 0.000102 0.101616
5 IRAC.1.5001.fits 159.016861 59.167344 0.479648 0.000043 0.000072 0.101570
6 IRAC.1.6001.fits 159.038956 59.205679 0.456492 0.000059 0.000055 0.090561
7 IRAC.1.7001.fits 158.957133 59.161555 0.516593 0.000049 0.000117 0.103518
8 IRAC.1.8001.fits 158.940095 59.173085 0.462516 0.000041 0.000130 0.102333
9 IRAC.1.9001.fits 159.008266 59.170001 0.488810 0.000041 0.000078 0.101210
10 IRAC.1.1001.fits 158.956352 59.184437 0.481531 0.000041 0.000116 0.100883

```

(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.

Img#	theta  deg	X_shift  ref_pix	Y_shift  ref_pix	Err_theta  deg	Err_X  ref_pix	Err_Y  ref_pix	NASTROM
1	-0.026478	-1.233473	0.850536	0.102698	0.156743	0.254244	1
2	0.068312	-1.110399	0.153084	0.104464	0.127155	0.181946	0
3	-0.094970	0.418239	0.330770	0.104017	0.132257	0.372433	3
4	0.011755	-0.395867	0.897485	0.101616	0.146039	0.305199	0
5	-0.023190	0.620723	0.590670	0.101570	0.125114	0.216150	0
6	0.021597	0.141953	0.215126	0.090561	0.172765	0.165146	4
7	-0.004009	-0.733558	-0.750462	0.103518	0.140041	0.348311	0
8	0.033793	0.611526	0.861340	0.102333	0.116684	0.388635	3
9	-0.041469	0.242030	0.545410	0.101210	0.119720	0.232493	0
10	0.063298	-0.671616	-0.263467	0.100883	0.118393	0.347003	2

(4). Ancillary information on which images could not be refined due to non-correlations, fraction of input images actually refined 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 are 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=10 (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) = 76.2%

Following values are in units of degrees.

Image #	Inp-Refnd RA	Inp-Refnd DEC	Inp-Refnd CROTA2	#Abs.Sources used
1	-0.000804	-0.000292	-0.025809	1
2	-0.000729	-0.000056	0.069011	0
3	0.000278	-0.000110	-0.095186	3
4	-0.000253	-0.000304	0.011977	0
5	0.000413	-0.000197	-0.023535	0
6	0.000095	-0.000072	0.021513	4
7	-0.000489	0.000250	-0.003590	0
8	0.000410	-0.000288	0.033419	3
9	0.000164	-0.000183	-0.041601	0
10	-0.000444	0.000086	0.063718	2

(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\_matrix\_coeffs.txt*”: Lists all non-zero matrix elements and row locations (in Compressed Sparse Column or Harwell-Boeing format).
- e. “*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).
- f. “*pointingrefine\_RHS\_coeffs.txt*”: Lists elements of right-hand-side vector  $B$  of the matrix equation  $AX = B$ .
- g. “*covariance\_matrix.txt*”: Lists non-zero elements of the full error-covariance matrix for reference-frame offsets of all *correlated* images.