

Combining Error Ellipses

Version 0.3-3

John E. Davis

<davis@space.mit.edu>

June 1, 2011

1 Introduction

The purpose of this document is to present a formalism for statistically averaging source positions and their uncertainties for use in the level-3 pipeline. More specifically, the problem addressed here is to find an improved estimate for the position of a source from previous independent estimates of its position. The uncertainties of the estimates are expressed in the form of error ellipses centered upon the estimated positions.

This document is organized as follows: In section §2, the much simpler one-dimensional problem of optimal weighting is addressed. Section §3 extends the approach of §2 to the multivariate case section. Then in section §4 the results of section §3 are applied to the two-dimensional case involving the geometric parameters of the error ellipses after projection to a common tangent plane. The tangent plane projections themselves are discussed in section §5. A brief summary that provides a sort of road map to the key equations necessary for the implementation of the methodology follows in §6. Finally there is an appendix that contains the code-listing for a **S-Lang** implementation as well as an example of its use.

2 Optimal Weighting

Before tackling the more general n -dimensional case, it is useful to consider the simpler 1-d case. Suppose that \bar{x}_a represents the a th estimate of the mean μ of some quantity, e.g., a temperature, and let σ_a^2 be the variance of the a th mean. Given a set of such estimates \bar{x}_a of the mean, and the corresponding set of variances σ_a^2 , what is the best way to combine these to obtain an improved estimate of the mean and the variance of that estimate? The approach taken here is to use an optimal weighting scheme that minimizes the resulting variance. Let \bar{x} denote the improved estimate and let w_a be the set of weights. Then an unbiased estimate of μ is

$$\bar{x} = \frac{1}{w} \sum_a w_a \bar{x}_a, \quad (1)$$

where

$$w = \sum_a w_a \quad (2)$$

That is,

$$\mu = \langle \bar{x} \rangle, \quad (3)$$

where $\langle \cdot \rangle$ denotes the expectation value, and the individual estimates \bar{x}_a are assumed to be unbiased.

From equation (1) it is easy to show that

$$\text{Var}[\bar{x}] = \sum_a \left(\frac{w_a}{w}\right)^2 \text{Var}[\bar{x}_a]. \quad (4)$$

The conditions for σ^2 to be a minimum may be obtained by differentiating the above equation with respect to w_b . This procedure yields

$$w_b \text{Var}[\bar{x}_b] = \frac{1}{w} \sum_a w_a^2 \text{Var}[\bar{x}_a]. \quad (5)$$

Since the right-hand-side of the above equation is independent of b , it follows that w_b is proportional to $1/\text{Var}[\bar{x}_b]$. Substituting these weights into equation (1) produces

$$\bar{x} = \left[\sum_a \text{Var}[\bar{x}_a]^{-1} \right]^{-1} \sum_a \text{Var}[\bar{x}_a]^{-1} \bar{x}_a, \quad (6)$$

which, by construction, is the linear combination of \bar{x}_a with the smallest variance. Substituting the weights into equation (4) yields the variance in \bar{x} :

$$\text{Var}[\bar{x}] = \left[\sum_a \text{Var}[\bar{x}_a]^{-1} \right]^{-1}, \quad (7)$$

allowing equation (6) to be written as

$$\bar{x} = \text{Var}[\bar{x}] \sum_a \text{Var}[\bar{x}_a]^{-1} \bar{x}_a. \quad (8)$$

3 The Multivariate Case

In this section, the previous technique is extended to the multivariate case. Let X_a represent the a th estimate of the mean of some N -dimensional quantity μ , and let σ_a denote the N by N covariance matrix associated with this estimate. That is,

$$\sigma_{a,ij} = \langle (X_{a,i} - \mu_i)(X_{a,j} - \mu_j) \rangle, \quad (9)$$

where $\mu_i = \langle X_{a,i} \rangle$. From the above equation it is straight-forward to show that

$$\langle X_{a,i} X_{a,j} \rangle = \sigma_{a,ij} + \mu_i \mu_j. \quad (10)$$

An improved estimate X for μ may be obtained by a weighted sum of the individual estimates X_a , i.e.,

$$X = \sum_a W_a X_a. \quad (11)$$

Here, W_a are a set of N by N matrices whose matrix elements are to be obtained. In order that $\mu = \langle X \rangle$, it is necessary for the matrix elements to satisfy the constraint

$$\delta_{ij} = \sum_a W_{a,ij}. \quad (12)$$

It is easy to show that the covariance matrix σ is given by

$$\begin{aligned} \sigma_{ij} &= \langle (X_i - \mu_i)(X_j - \mu_j) \rangle, \\ &= \sum_a (W_a \sigma_a W_a^\top)_{ij}. \end{aligned} \quad (13)$$

Now let \mathcal{R} be a rotation matrix that transforms the vector X to $X' = \mathcal{R}X$. Then it is easy to show that σ transforms as

$$\sigma' = \mathcal{R}\sigma\mathcal{R}^{-1}. \quad (14)$$

It is well known that the above (similarity) transformation may be used to diagonalize σ by choosing \mathcal{R} appropriately. In the basis where σ' is diagonal the product of the diagonal elements σ'_{ii} may be used as a measure total variance as this product is related to the volume of the ellipsoid associated with the covariance matrix. In the diagonal basis, the product $\prod_i \sigma'_{ii}$ also corresponds to the determinant of σ' , denoted here as $\det(\sigma')$. Since the determinant is invariant under similarity transformations, it follows that $\det(\sigma') = \det(\sigma)$.

In other words, $\det(\sigma)$ corresponds to the volume of the covariance ellipsoid and is taken as a scalar measure of the “total error”. Hence, the weights $W_{a,ij}$ will be chosen to minimize the determinant of the covariance matrix σ subject to the normalization conditions of equation (12). The constraints are most easily handled through the use of Lagrange multipliers λ_{ij} , where the function to be minimized may be written as

$$\det(\sigma) + \lambda_{ij}(\delta_{ij} - \sum_a W_{a,ij}). \quad (15)$$

Here and in the following, the Einstein summation convention is used where unless otherwise specified, repeated indices i, j, \dots are to be summed over.

The minimization conditions follows in the usual way and may be written as

$$0 = \frac{\partial \det(\sigma)}{\partial W_{a,ij}} - \lambda_{ij} \quad (16)$$

and

$$0 = \delta_{ij} - \sum_a W_{a,ij}. \quad (17)$$

The derivatives involving $W_{a,ij}$ may be carried out using the chain rule

$$\frac{\partial \det(\sigma)}{\partial W_{a,ij}} = \frac{\partial \det(\sigma)}{\partial \sigma_{lm}} \frac{\partial \sigma_{lm}}{\partial W_{a,ij}}. \quad (18)$$

It is left as an exercise for the reader to show that

$$\frac{\partial \sigma_{lm}}{\partial W_{a,ij}} = \delta_{il} \sigma_{a,jk} W_{a,mk} + W_{a,lk} \sigma_{a,kj} \delta_{mi}. \quad (19)$$

By expanding $\det(\sigma)$ in terms of its cofactors, one can show (see any advanced linear algebra text) that

$$\frac{\partial \det(\sigma)}{\partial \sigma_{lm}} = \det(\sigma) \sigma_{lm}^{-1}. \quad (20)$$

Together with the last two results and equation (18), the minimization condition given by equation (16) may be written in the form

$$\lambda_{ij} = 2 \det(\sigma) (\sigma^{-1} W_a \sigma_a)_{ij}. \quad (21)$$

Since the left-hand-side of this equation is independent of a , it follows that W_a must be of the form $A \sigma_a^{-1}$, where A is some matrix that is independent of a . The normalization condition of equation (12) may be used to determine A yielding

$$W_a = \left[\sum_b \sigma_b^{-1} \right]^{-1} \sigma_a^{-1}. \quad (22)$$

Substituting this result into equation (13) and exploiting the symmetry of σ produces

$$\sigma = \left[\sum_a \sigma_a^{-1} \right]^{-1}. \quad (23)$$

Finally equation (11) may be written

$$X = \sigma \sum_a \sigma_a^{-1} X_a, \quad (24)$$

which is the main result of this section. Note the formal resemblance of this equation to the univariate case in equation (8).

4 Computing Covariance Matrices

As seen in section §3, covariance matrices play a fundamental role in combining error ellipses. This section deals with the computation of the covariance matrices from the parameters that characterize the elliptical geometry. It is assumed that the ellipses have been projected to a common tangent plane, as described in section §5.

The geometry of each error ellipse is specified by five parameters, of which three are directly related to the covariance matrix. These are the angle θ that the major

axis of the ellipse makes with respect to the tangent plane y axis, and the semi-major and semi-minor axis lengths. The lengths of the semi-major and semi-minor axes correspond to the 1-sigma confidence intervals along these axes. More specifically, in a basis whose origin is at the center of the ellipse, and whose y axis is along the ellipse's major axis, the correlation matrix is

$$\sigma' = \begin{pmatrix} \sigma_1'^2 & 0 \\ 0 & \sigma_2'^2 \end{pmatrix}. \quad (25)$$

Here, σ_1' is the 1-sigma confidence value along the minor axis of the ellipse, and σ_2' is that along the major axis ($\sigma_2' \geq \sigma_1'$). The form of the covariance matrix σ in the unrotated system follows from equation (14) using

$$\mathcal{R} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (26)$$

to yield

$$\sigma = \begin{pmatrix} \sigma_1'^2 \cos^2 \theta + \sigma_2'^2 \sin^2 \theta & (\sigma_2'^2 - \sigma_1'^2) \cos \theta \sin \theta \\ (\sigma_2'^2 - \sigma_1'^2) \cos \theta \sin \theta & \sigma_1'^2 \sin^2 \theta + \sigma_2'^2 \cos^2 \theta \end{pmatrix}. \quad (27)$$

It is left as an exercise for the reader to show that the inverse relations are

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{2\sigma_{12}}{\sigma_{22} - \sigma_{11}} \right), \quad (28)$$

$$\sigma_1'^2 = \frac{1}{2} \left[\sigma_{11} + \sigma_{22} - \sqrt{(\sigma_{22} - \sigma_{11})^2 + 4\sigma_{12}^2} \right], \quad (29)$$

and

$$\sigma_2'^2 = \frac{1}{2} \left[\sigma_{11} + \sigma_{22} + \sqrt{(\sigma_{22} - \sigma_{11})^2 + 4\sigma_{12}^2} \right]. \quad (30)$$

5 Tangent Plane Projections

In order to combine the confidence ellipses via equation (24), it is first necessary to project them to a common tangent plane. This procedure is described in this section.

The a th estimate of the source position is specified as a confidence-ellipse centered upon the celestial coordinate (α_a, δ_a) , with the major-axis making an angle θ_a ($-\pi \leq \theta < \pi$) with respect to the local line of declination at the center of the ellipse. The arc-length of the semi-minor axis is given by the value ϕ_a^{minor} and that of the semi-major axis is given by ϕ_a^{major} .

The celestial coordinate (α_a, δ_a) corresponds to a unit-vector \hat{p}_a on the celestial sphere with coordinates given by

$$\hat{p}_a = \hat{x} \cos \alpha_a \cos \delta_a + \hat{y} \sin \alpha_a \cos \delta_a + \hat{z} \sin \delta_a. \quad (31)$$

Conversely, a unit vector \hat{p}_a corresponds to the celestial coordinate

$$(\alpha_a, \delta_a) = \left(\tan^{-1} \frac{(\hat{p}_a \cdot \hat{y})}{(\hat{p}_a \cdot \hat{x})}, \sin^{-1}(\hat{p}_a \cdot \hat{z}) \right), \quad (32)$$

which is the inverse of equation (31).

An orthonormal coordinate system is defined at the point represented by \hat{p}_a consists of the three unit vectors \hat{p}_a , $\hat{\alpha}_a$, and $\hat{\delta}_a$ where

$$\hat{\alpha}_a = -\hat{x} \sin \alpha_a + \hat{y} \cos \alpha_a \quad (33)$$

and

$$\hat{\delta}_a = -\hat{x} \sin \delta_a \cos \alpha_a - \hat{y} \sin \delta_a \sin \alpha_a + \hat{z} \cos \delta_a. \quad (34)$$

Note that $\hat{\delta}_a$ points along the direction of increasing declination at the position \hat{p}_a , whereas $\hat{\alpha}_a$ points in the direction of increasing right-ascension. The semi-major axis of the confidence ellipse associated with this position makes an angle θ_a with respect to $\hat{\delta}_a$. The sign of θ_a is in accordance with the right hand rule with $-\hat{p}_a$ as the rotation axis. For $\theta_a = 0$, the “positive” end of the semi-minor axis will have coordinates $(\alpha_a + \phi_a^{\text{minor}}, \delta_a)$ and correspond to a unit vector \hat{p}_a^{minor} , whereas the “positive” end of the semi-major axis will lie at $(\alpha_a, \delta_a + \phi_a^{\text{major}})$ and correspond to the unit vector \hat{p}_a^{major} . For $\theta_a = 0$, these unit vectors are given by an equation of the same form as equation (31). The vectors that correspond to non-zero values of θ_a may be obtained by rotating the $\theta_a = 0$ values about the $-\hat{p}_a$ axis by the angle θ_a . This operation is most easily carried out in the local coordinate basis $(\hat{\alpha}_a, \hat{\delta}_a, \hat{p}_a)$ producing

$$\hat{p}_a^{\text{minor}} = \hat{p}_a \cos \phi_a^{\text{minor}} + \hat{\alpha}_a \sin \phi_a^{\text{minor}} \cos \theta - \hat{\delta}_a \sin \phi_a^{\text{minor}} \sin \theta \quad (35)$$

and

$$\hat{p}_a^{\text{major}} = \hat{p}_a \cos \phi_a^{\text{major}} + \hat{\alpha}_a \sin \phi_a^{\text{major}} \sin \theta + \hat{\delta}_a \sin \phi_a^{\text{major}} \cos \theta. \quad (36)$$

These two equations along with equations (31), (33), and (34) are sufficient to compute the unit vectors associated with the major and minor axes of the error ellipses. The inverse relations are easily obtained by taking the appropriate dot-products, producing

$$\theta = \tan^{-1} \left(\frac{\hat{p}_a^{\text{major}} \cdot \hat{\alpha}_a}{\hat{p}_a^{\text{major}} \cdot \hat{\delta}_a} \right) = -\tan^{-1} \left(\frac{\hat{p}_a^{\text{minor}} \cdot \hat{\delta}_a}{\hat{p}_a^{\text{minor}} \cdot \hat{\alpha}_a} \right), \quad (37)$$

$$\phi_a^{\text{major}} = \cos^{-1}(\hat{p}_a^{\text{major}} \cdot \hat{p}_a), \quad (38)$$

and

$$\phi_a^{\text{minor}} = \cos^{-1}(\hat{p}_a^{\text{minor}} \cdot \hat{p}_a). \quad (39)$$

Let \hat{p}_0 denote the position on the celestial sphere where a tangent plane is to be erected. To minimize any distortion effects created when mapping from the celestial sphere onto the tangent plane, \hat{p}_0 will be taken as the arithmetic mean of the ellipse centers \hat{p}_a , i.e.,

$$\hat{p}_0 = \frac{\sum_a \hat{p}_a}{|\sum_a \hat{p}_a|}. \quad (40)$$

A coordinate system may be given to the tangent plane with the origin at \hat{p}_0 and orthonormal basis vectors \hat{e}_x and \hat{e}_y parallel to the local lines of right ascension and declination at \hat{p}_0 , i.e.,

$$\hat{e}_x = \hat{\alpha}_0 = -\hat{x} \sin \alpha_0 + \hat{y} \cos \alpha_0 \quad (41)$$

$$\hat{e}_y = \hat{\delta}_0 = -\hat{x} \sin \delta_0 \cos \alpha_0 - \hat{y} \sin \delta_0 \sin \alpha_0 + \hat{z} \cos \delta_0. \quad (42)$$

Here, (α_0, δ_0) are the celestial coordinates that correspond to \hat{p}_0 .

The tangent plane projection of \hat{p} is defined by

$$\hat{p}t = \hat{p}_0 + x\hat{e}_x + y\hat{e}_y \quad (43)$$

where (x, y) denote the tangent plane coordinates associated with \hat{p} . It is a trivial matter to show that $t = 1/(\hat{p} \cdot \hat{p}_0)$,

$$x = (\hat{p} \cdot \hat{e}_x)/(\hat{p} \cdot \hat{p}_0) \quad (44)$$

and

$$y = (\hat{p} \cdot \hat{e}_y) / (\hat{p} \cdot \hat{p}_0) \quad (45)$$

It also follows from equation (43) that a point (x, y) in the tangent plane corresponds to the unit vector

$$\hat{p} = \frac{\hat{p}_0 + x\hat{e}_x + y\hat{e}_y}{\sqrt{1 + x^2 + y^2}}. \quad (46)$$

Note that the mapping from \hat{p} to (x, y) is non-linear. The source of the non-linearity is the factor $\hat{p} \cdot \hat{p}_0$, which represents the cosine of the angle between \hat{p} and \hat{p}_0 . Since this angle is expected to be small, e.g., < 10 arc-minutes, the effects of this term may be ignored ($\cos 10' \approx 1 - 4 \times 10^{-8}$). Now consider two vectors \hat{p}_1 and \hat{p}_2 with an angle ξ between them such that $\hat{p}_1 \cdot \hat{p}_2 = \cos \xi$, and let their tangent plane coordinates be (x_1, y_1) and (x_2, y_2) , respectively. Then in the small angle regime where $\hat{p} \cdot \hat{p}_0$ may be taken to be 1,

$$\hat{p}_1 - \hat{p}_2 = (x_1 - x_2)\hat{e}_1 + (y_1 - y_2)\hat{e}_2 \quad (47)$$

and

$$|\hat{p}_1 - \hat{p}_2| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (48)$$

If the angle between \hat{p}_1 and \hat{p}_2 is ξ , then

$$\begin{aligned} |\hat{p}_1 - \hat{p}_2| &= \sqrt{2(1 - \cos \xi)} \\ &= \xi + O(\xi^3), \end{aligned} \quad (49)$$

which shows that in the small angle regime, the arc-length between two celestial coordinates is equal to the distance between the tangent plane projections of those coordinates. This means that the arc-lengths of the semi-major and minor axes of the error ellipses will be preserved to sufficient accuracy by the tangent plane projection.

Armed with these relations, it is easy to compute the tangent plane projections of the error ellipses. The tangent plane coordinate (x_a, y_a) of the center of the a th ellipse follows from equations (44) and (45), i.e.,

$$x_a = (\hat{p}_a \cdot \hat{e}_x) / (\hat{p}_a \cdot \hat{p}_0) \quad (50)$$

and

$$y_a = (\hat{p}_a \cdot \hat{e}_y) / (\hat{p}_a \cdot \hat{p}_0), \quad (51)$$

where \hat{p}_a is given by equation (31). Similar equations give the tangent plane coordinates that correspond to the end-point positions \hat{p}_a^{major} and \hat{p}_a^{minor} of the semi-major and semi-minor axes of the ellipse. Denoting these coordinates as $(x_a^{\text{major}}, y_a^{\text{major}})$ and $(x_a^{\text{minor}}, y_a^{\text{minor}})$, the lengths of the semi-major and semi-minor axes in the tangent plane are given by

$$\sigma'_2 = \sqrt{(x_a^{\text{major}} - x_a)^2 + (y_a^{\text{major}} - y_a)^2} \quad (52)$$

and

$$\sigma'_1 = \sqrt{(x_a^{\text{minor}} - x_a)^2 + (y_a^{\text{minor}} - y_a)^2}, \quad (53)$$

respectively. Here, the symbols representing these lengths have been chosen to be consistent with equation (25). As noted above, the lengths of the semi-major and minor-axes in the tangent plane should differ by a negligible amount from those of the celestial system, assuming the small angle approximation. In contrast, the angle that the semi-major axis makes with respect to the local line of declination will differ between the two systems, particularly when the ellipse is located near the poles of the celestial sphere. The angle as seen in the tangent plane is

$$\theta'_a = \tan^{-1} \left(\frac{x_a^{\text{major}} - x_a}{y_a^{\text{major}} - y_a} \right). \quad (54)$$

From equations (52), (53), and (54), it is easy to show that the inverse relations are

$$(x_a^{\text{major}}, y_a^{\text{major}}) = (x_a + \sigma'_2 \sin \theta'_a, y_a + \sigma'_2 \cos \theta'_a), \quad (55)$$

and

$$(x_a^{\text{minor}}, y_a^{\text{minor}}) = (x_a + \sigma'_1 \cos \theta'_a, y_a - \sigma'_1 \sin \theta'_a), \quad (56)$$

and from these the corresponding unit vectors may be obtained through the use of equation (46).

6 Summary

Equations (50), (51), (52), (53), and (54) constitute a set of equations that may be used to map error ellipses from the celestial system onto a common tangent

plane, whose location is given by equation (40). Once projected to the tangent plane, covariance matrices may be computed using equation (27), permitting the error ellipses to be combined via equation (24). This process produces the geometric parameters of a combined error ellipse on the tangent plane. The mapping of the error ellipse from the tangent plane back to the celestial system may be carried out using equations (37), (38), (39), (46), (55), and (56).

The weighting procedure as proposed here for the problem of combining error ellipses is not new. Equation 22 appears to be the basis for merging source positions for the 2MASS catalog as described in [1]. However, no mention of where this equation comes from is given. This problem was also dealt with by Orechovesky [2] in 1996 for military purposes involving geographic locations. His formalism makes use of Bayesian methods and Gaussian statistics. In fact, equation (22) can be derived very simply by assuming a (multivariate) Gaussian probability distribution and demanding that the likelihood be a maximum. In contrast, the minimum-variance derivation of equation (22) presented in section §3 makes no reference to Gaussian statistics, and as such may be of more general validity.

A Appendix

This appendix contains provides the code for **S-Lang** implementation of the algorithm proposed in this memo. The program is an **slsh** script that loads a data file of input ellipses and writes out the combined result.

For example, consider the three input ellipses described by the following data file:

```
# alpha  delta  semi-major  semi-minor  theta
# [deg]  [deg]  [arc-min]  [arc-min]  [deg]
   30    71.6    50         24         18
  29.2   71.7    23         16         27
  30.3   72.3    47         5          -56
```

Running the **slsh** script produces the following output for the combined ellipse:

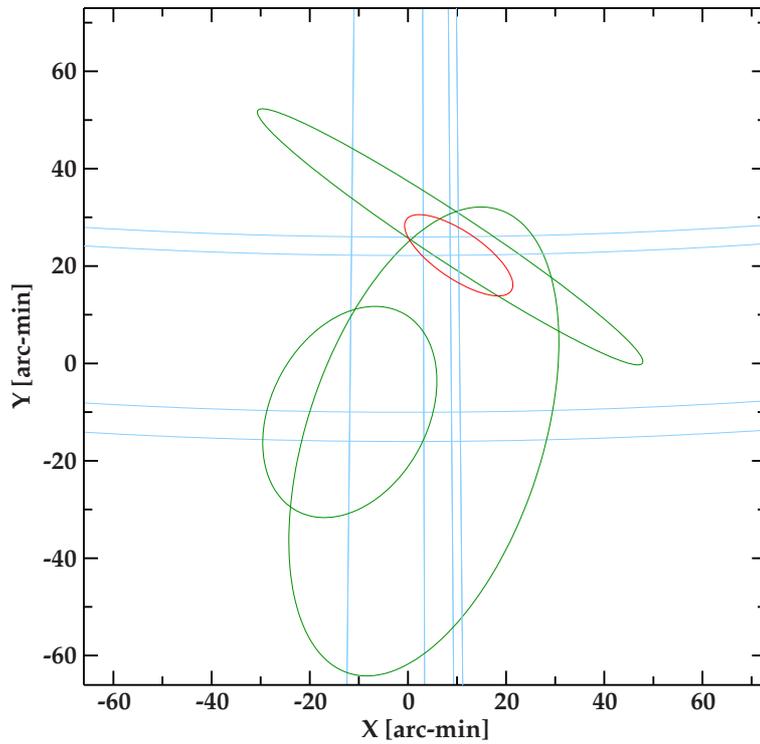


Figure 1: The input ellipses are shown in green and the resulting combined ellipse is in red. The x and y values represent tangent plane coordinates (arc-minutes).

```

alpha: 30.393881846278557 degrees
delta: 72.23672003826117 degrees
theta: 124.42667707460919 degrees
major: 12.945681354683803 arc-min (776.7408812810281 arc-sec)
minor: 4.855409627388941 arc-min (291.32457764333645 arc-sec)

```

The input ellipses (green) and the combined ellipse (red) are shown in Figure 1.

```

#!/usr/bin/env slsh
require ("readascii");

% This structure will be used to hold information about each ellipse
private variable Ellipse_Type = struct
{
    alpha, delta, phi_major, phi_minor, theta_cel, % celestial
    x, y, sigma_major, sigma_minor, theta, % tangent plane
    p_hat, alpha_hat, delta_hat, p_major, p_minor
};

private define dotprod (x, y)
{
    return sum(x*y);
}
private define norm (x)
{
    return sqrt (sum(x*x));
}

private define new_ellipse (alpha, delta, phi_major, phi_minor, theta_cel)
{
    variable e = @Ellipse_Type;
    e.alpha = alpha;
    e.delta = delta;
    e.phi_major = phi_major;
    e.phi_minor = phi_minor;
    e.theta_cel = theta_cel;
    variable ca=cos(alpha), cd=cos(delta), sa=sin(alpha), sd=sin(delta);
    e.p_hat = [ca*cd, sa*cd, sd]; % eq 31
    e.alpha_hat = [-sa, ca, 0]; % eq 33
    e.delta_hat = [-sd*ca, -sd*sa, cd]; % eq 34
    e.p_minor = e.p_hat*cos(phi_minor)
        + e.alpha_hat*sin(phi_minor)*cos(theta_cel)
        - e.delta_hat*sin(phi_minor)*sin(theta_cel); % eq 35
    e.p_major = e.p_hat*cos(phi_major)
        + e.alpha_hat*sin(phi_major)*sin(theta_cel)
        + e.delta_hat*sin(phi_major)*cos(theta_cel); % eq 36

    return e;
}

private define get_tangent_plane_from_vector (p0)
{
    variable alpha0 = atan2 (p0[1], p0[0]); % eq 32
    variable delta0 = asin (p0[2]);
    variable ex_hat = [-sin(alpha0), cos(alpha0), 0]; % eq 41
    variable ey_hat = [-sin(delta0)*cos(alpha0),
        -sin(delta0)*sin(alpha0), cos(delta0)]; % eq 42
    return p0, ex_hat, ey_hat;
}

private define get_tangent_plane_from_ellipses (ellipses)
{
    variable p0 = 0;
    foreach (ellipses)
    {

```

```

        variable e = ();
        p0 += e.p_hat;
    }
    p0 /= norm(p0); % eq 40
    return get_tangent_plane_from_vector(p0);
}

private define project_ellipse(e, p0_hat, ex_hat, ey_hat)
{
    variable p, p_dot_p0;
    variable xa, ya, xa_major, xa_minor, ya_major, ya_minor;

    p = e.p_hat;
    p_dot_p0 = dotprod(p, p0_hat);
    xa = dotprod(p, ex_hat)/p_dot_p0; % eq 50
    ya = dotprod(p, ey_hat)/p_dot_p0; % eq 51

    p = e.p_major;
    p_dot_p0 = dotprod(p, p0_hat);
    xa_major = dotprod(p, ex_hat) / p_dot_p0; % eq 50
    ya_major = dotprod(p, ey_hat) / p_dot_p0; % eq 51

    p = e.p_minor;
    p_dot_p0 = dotprod(p, p0_hat);
    xa_minor = dotprod(p, ex_hat) / p_dot_p0; % eq 50
    ya_minor = dotprod(p, ey_hat) / p_dot_p0; % eq 51

    e.x = xa;
    e.y = ya;
    e.sigma_major = hypot(xa_major-xa, ya_major-ya); % eq 52
    e.sigma_minor = hypot(xa_minor-xa, ya_minor-ya); % eq 53
    e.theta = atan2(xa_major-xa, ya_major-ya); % eq 54
}

private define deproject_ellipse(e, p0, ex_hat, ey_hat)
{
    variable p = p0 + e.x*ex_hat + e.y*ey_hat;
    p /= norm(p); % eq 46
    e.p_hat = p;
    e.alpha = atan2(p[1], p[0]); % eq 32
    e.delta = asin(p[2]);

    variable x_major = e.x + e.sigma_major*sin(e.theta); % eq 55
    variable y_major = e.y + e.sigma_major*cos(e.theta);
    p = p0 + x_major*ex_hat + y_major*ey_hat;
    e.p_major = p/norm(p); % eq 46

    variable x_minor = e.x + e.sigma_minor*cos(e.theta); % eq 56
    variable y_minor = e.y - e.sigma_minor*sin(e.theta);
    p = p0 + x_minor*ex_hat + y_minor*ey_hat;
    e.p_minor = p/norm(p); % eq 46

    variable ca=cos(e.alpha), cd=cos(e.delta);
    variable sa=sin(e.alpha), sd=sin(e.delta);
    e.alpha_hat = [-sa, ca, 0]; % eq 33
    e.delta_hat = [-sd*ca, -sd*sa, cd]; % eq 34
}

```

```

    % Equations 37, 38, 39
    e.theta_cel = atan (dotprod (e.p_major, e.alpha_hat)
                          / dotprod (e.p_major, e.delta_hat));
    e.phi_major = acos (dotprod (e.p_major, e.p_hat));
    e.phi_minor = acos (dotprod (e.p_minor, e.p_hat));
}

% Implements eq 27
private define ellipse_to_correlation_matrix (e)
{
    variable sigy2 = e.sigma_major^2, sigx2 = e.sigma_minor^2;
    variable c = cos(e.theta);
    variable s = sin(e.theta);
    variable c2 = c*c, s2 = s*s;

    variable sx2 = sigx2*c2 + sigy2*s2;
    variable sy2 = sigx2*s2 + sigy2*c2;
    variable rho_sxsy = c*s*(sigy2-sigx2);

    return _reshape ([sx2, rho_sxsy, rho_sxsy, sy2], [2,2]);
}

% Implements equations 28, 29, 30
private define correlation_matrix_to_ellipse (matrix, x0, y0)
{
    variable sx2 = matrix[0,0];
    variable sy2 = matrix[1,1];
    variable rho2_sxsy = 2*matrix[0,1];
    variable sum = sy2+sx2;
    variable diff = sy2-sx2;

    variable e = @Ellipse_Type;
    e.x = x0, e.y = y0;
    e.theta = 0.5*atan2 (rho2_sxsy, diff);
    diff = hypot (diff, rho2_sxsy);
    e.sigma_major = sqrt (0.5*(sum + diff));
    e.sigma_minor = sqrt (0.5*(sum - diff));
    return e;
}

private define inverse_2x2 (a)
{
    variable det = a[0,0] * a[1,1] - a[0,1]*a[1,0];
    if (det == 0.0)
        throw RuntimeError, "matrix is singular";
    variable a1 = Double_Type[2,2];
    a1[0,0] = a[1,1];
    a1[0,1] = -a[0,1];
    a1[1,0] = -a[1,0];
    a1[1,1] = a[0,0];
    return a1/det;
}

% Implements eq 24
private define combine_ellipses_internal (es)
{
    variable num = length(es);

```

```

variable Cinv = 0;
variable mu = 0;
_for (0, num-1, 1)
{
    variable i = ();
    variable e = es[i];
    variable C_m = ellipse_to_correlation_matrix (e);
    variable Cinv_m = inverse_2x2 (C_m);
    mu += Cinv_m # [e.x, e.y];
    Cinv += Cinv_m;
}
variable C = inverse_2x2 (Cinv);
mu = C # mu;
return correlation_matrix_to_ellipse (C, mu[0], mu[1]);
}

% alpha, delta, theta are in degrees, phi_major/minor are arc-minutes
define combine_ellipses (alphas, deltas, phimajors, phiminors, thetas)
{
    % convert to radians
    variable rad_per_deg = PI/180.0;
    alphas *= rad_per_deg;
    deltas *= rad_per_deg;
    phimajors *= rad_per_deg/60.0;
    phiminors *= rad_per_deg/60.0;
    thetas *= rad_per_deg;

    variable i, e, ellipses = {};
    _for i (0, length(thetas)-1, 1)
    {
        e = new_ellipse (alphas[i], deltas[i], phimajors[i],
                        phiminors[i], thetas[i]);
        list_append (ellipses, e);
    }

    variable p0, ex_hat, ey_hat;
    (p0, ex_hat, ey_hat) = get_tangent_plane_from_ellipses (ellipses);

    foreach e (ellipses)
        project_ellipse (e, p0, ex_hat, ey_hat);

    variable new_e = combine_ellipses_internal (ellipses);
    deproject_ellipse (new_e, p0, ex_hat, ey_hat);

    return
        new_e.alpha/rad_per_deg,
        new_e.delta/rad_per_deg,
        new_e.phi_major/rad_per_deg*60.0,
        new_e.phi_minor/rad_per_deg*60.0,
        new_e.theta/rad_per_deg;
}

define slsh_main ()
{
    variable alphas, deltas, phimajors, phiminors, thetas;
    variable use_arc_secs = 0;

```

```

if (__argc != 2)
{
    () = fprintf (stderr, "Usage: %s ellipse.dat\n", __argv[0]);
    exit (1);
}

variable infile = __argv[1];
() = readascii (infile, &alphas, &deltas, &phimajors, &phiminors, &thetas;
               type="%lf");

if (use_arc_secs)
{
    phiminors/=60.0; phimajors/=60.0;
}

variable alpha, delta, phimajor, phiminor, theta;

(alpha, delta, phimajor, phiminor, theta)
= combine_ellipses (alphas, deltas, phimajors, phiminors, thetas);

theta = (theta+180.0) mod 180.0;
() = fprintf (stdout, "alpha: %18S degrees\n", alpha);
() = fprintf (stdout, "delta: %18S degrees\n", delta);
() = fprintf (stdout, "theta: %18S degrees\n", theta);
() = fprintf (stdout, "major: %18S arc-min (%S arc-sec)\n", phimajor, phimajor*60.0);
() = fprintf (stdout, "minor: %18S arc-min (%S arc-sec)\n", phiminor, phiminor*60.0);
}

```

References

- [1] http://www.ipac.caltech.edu/2mass/releases/allsky/doc/seca6_2.html
- [2] Joseph R. Orechovesky Jr, Single Source Error Ellipse Combination, 1996
Master's Thesis, Naval Postgraduate School, Monterey California