Wide-field Infrared Survey Explorer (WISE)

Solar System Object Identification (SSOID) Subsystem Design Document

Version 1.2

12 March 2009

Prepared by: John W. Fowler

Infrared Processing and Analysis Center
California Institute of Technology

WSDC D-D017

Concurred By:

Roc Cutri, WISE Science Data Center Manager

Tim Conrow, WISE Science Data Center System Architect

John Fowler, WISE Science Data Center PRex Cognizant Engineer

Dave Tholen, WISE Solar-System Object Prediction Cognizant Engineer

Revision History

Date	Version	Author	Description
9 October 2008	0.1	J. W. Fowler	Preliminary Draft
15 December 2008	1.0	J.W. Fowler	Initial Version
20 January 2009	1.1	J.W. Fowler	Added frame coordinates for testing unmatched SSOs to determine whether they are truly “missed”
12 March 2009	1.2	J.W. Fowler	Added option to skip one-way light time correction (for supporting tests using simulation), "-xL"; added array-limit specifications, "-cn", "-cx", "-rn", "-rx"; added output meta-data option, "-om"

2.2.2 FITS Input

Table of Contents

1 Introduction

1.1 Subsystem Overview

This document presents the requirements, design, algorithms, and state of implementation of the SSOID (Solar System Object Identification) subsystem of the WSDC data processing system. The SSOID modules run in the scan pipeline with two processing scopes: SSOINIT processes a half-orbit scan to produce a subset of the SSO data base appropriate for the sky covered in that scan, and the other modules run on user-specifiable time-consecutive subsets of the scan.

1.1.1 Requirements

The SSOID subsystem is required to predict the positions of known solar system objects in WISE framesets and to associate these predictions with WISE point sources. The Level 4 requirements supported by this processing are as follows.

> L4WSDC-027**:** The WSDC shall identify and compile a listing of known solar system objects that are positionally associated with source extractions in the WISE single -epoch image frames.
>
> L4WSDC-028: The solar system objects associated with WISE single-epoch extractions shall include asteroids, comets, planets, and planetary satellites.

1.1.2 Liens

- Processing of planetary satellites has not yet been implemented.
- The output table file of SSO predictions and associations will be extended to include for WISE/SSO matches the four WISE fluxes, the derived quantities albedo, diameter, beaming parameter, subsolar temperature at 1 AU, and uncertainties for all of these; this has not yet been done.
- The method of determining whether a WISE flux is usable has not been specified.
- The output table file of SSO predictions and associations has not yet been made compatible with the WSDC table-file policy for representing nulls.
- The SIS for the output table file of SSO predictions and associations has not yet been completed.

1.2 Applicable Documents

This subsystem conforms to the specifications in the following project documents:

- WISE Science Data Center Functional Requirements Document, WSDC D-R001
- WISE Science Data System Functional Design, WSDC D-D001
- Software Management Plan, WSDC D-M003
- SIS sso01: SSO Orbital Elements File, WSDC D-I138
- SIS sso02: SSO ssoinit/ssoid Intermediate File, WSDC D-I139
- SIS sso03: SSO Three-Epoch Ephemerides File, WSDC D-I140
- SIS sso04: SSO/WISE Associations File, WSDC D-I141
- SIS sso05: SSO Meta-Data File, WSDC D-I142
- SIS pht01: WPHot Output Photometry Table, WSDC D-I107

1.3 Acronyms

2MASS	Two-Micron All-Sky Survey
FRD	Functional Requirements Document
SDS	Subsystem Design Specification
SIS	Software Interface Specification
SSO	Solar System Object
SSOID	Solar System Object Identification (subsystem and processor)
SSOINIT	Solar System Object Initialization processor
W1	WISE wavelength channel 1, 3.3 microns
W2	WISE wavelength channel 2, 4.7 microns
W3	WISE wavelength channel 3, 12 microns
W4	WISE wavelength channel 4, 23 microns
WISE	Wide-field Infrared Survey Explorer
WSDC	WISE Science Data Center
WSDS	WISE Science Data System

2 Input

2.1 SSOINIT Input

2.1.1 Control Input

The ssoinit module reads control input in the form of command-line parameters which are shown in its tutorial display:

```
ssoinit: Solar-System Object Initialization Utility vsn 1.1  A90105

 usage:  ssoinit <flags> <specifications>

 where <flags> <specifications> must be:

     -i  InputFNam   (Input orbital elements file name)
     -o  OutputFNam  (Output orbital elements subset file name)
     -f1 FITSnam1    (Name of first-frame FITS file)
     -f2 FITSnam2    (Name of ~mid-frame FITS file)
     -f3 FITSnam3    (Name of last-frame FITS file)
     -s1 s1x s1y s1z (Sun-to-Spacecraft vector components at
                      epoch of first frame, J2000, AU units)
     -s2 s2x s2y s2z (Sun-to-Spacecraft vector components at
                      epoch of ~middle frame, J2000, AU units)
     -s3 s3x s3y s3z (Sun-to-Spacecraft vector components at
                      epoch of last frame, J2000, AU units)
     -w  Window      (Half width of angle window perpendicular
                      to the scan axis, deg; default = 2)
     -i2 SSOlist     (Name of file containing a list of names
                      of SSOs to be output independently of
                      whether they are in the scan swath)
     -o2 SSOephem    (Name of output ephemeris file for SSOs
                      specified via "-i2")

 The first eight specifications above are required.
```

The input orbital elements file contains the orbital elements in the format used by the 2MASS DTPGM program. The output orbital elements subset file contains these same elements for the subset of SSOs that fall within the scan swath defined by the epochs of the first and third FITS files (swath endpoints) and the `Window` parameter. In addition to these SSOs' orbital elements, this file also contains six intermediate parameters per SSO to save time in the ssoid module. These six parameters are evaluated at the epoch of the second FITS file. Thus for every SSO in the subset file, there are two lines of ASCII text: the line from the input orbital elements file and a line containing the three components of the spacecraft-centered unit vector to the SSO, the mean motion, the topocentric distance, and the eccentric anomaly.

The second FITS file should be chosen to have an epoch near the center of the scan. This is because its epoch is used as the representative approximate time of the scan and because the cross products of its image-center unit vector with those of the other two FITS files are used to establish the scan axis unambiguously. This depends on the mission design employing survey scans that are always closer to 180 degrees long than to 270 degrees (there is no assumption that scans are not much shorter than 180 degrees).

The three J2000 spacecraft vectors must have epochs matching the three FITS files. The other command-line specifications are optional. The `Window` parameter specifies the half-width of the scan swath and the swath's extent before the first FITS file and after the third; its default is 2 degrees. The last two specifications provide an option to compute three-epoch ephemerides for any desired SSOs, independently of whether they fall inside the scan swath. This is provided for convenience, because the position of every SSO is computed for those epochs anyway, and there may be some interest in a given SSO that did not show up in the subset, for example. To use this option, a list of SSOs of interest must be given via the "`-i2`" specification; this file is an ASCII text file containing the name of an SSO on each line; the name may be the 35-character name employed in the main orbital elements file or any substring of that name that matches to the end of the full name. For example, the object with the full 35-character name

"`(17124)                  1999 JC65`"

may be specified simply as "`1999 JC65`". The substring "`(17124)`", however, would not be found. The three-epoch ephemerides are written to the file named via the "`-o2`" specification in table format. An example for three objects is given below.

```
\ Generated by ssoinit vsn 1.1  A90105
\Epoch1 = 2455212.787627
\Epoch2 = 2455212.803796
\Epoch3 = 2455212.819965
|               Name                |N|    RA     |    Dec    |
|               char                |i|   real    |   real    |
 1999 JC65                           1   29.715946  11.448790
 1999 JC65                           2   29.719113  11.448801
 1999 JC65                           3   29.722115  11.448754
 1994 PA14                           1   30.116379  11.313713
 1994 PA14                           2   30.119693  11.313744
 1994 PA14                           3   30.122826  11.313722
 5168 T-2                            1   29.912686  11.409998
 5168 T-2                            2   29.915956  11.409480
 5168 T-2                            3   29.919062  11.408908
```

The name is given in a 35-character field containing the name as specified in the "`-i2`" input file. The "`N`" parameter is the epoch number as given in the header. Topocentric J2000 RA and Dec are given in floating-point degrees.

2.1.2 Orbital Elements Input

The ssoinit module reads orbital elements input in the format used by the 2MASS DTPGM program (SIS sso01: SSO Orbital Elements File, WSDC D-I138)

2.1.3 FITS Input

The ssoinit module reads three FITS files, which must be in the correct order chronologically. The first and third must correspond to the scan endpoints, and the second should be close to the middle of the scan.

2.2 SSOID Input

2.2.1 Control Input

The ssoid module reads control input in the form of command-line parameters which are shown in its tutorial display:

```
ssoid: Solar-System Object Identification Program vsn 1.31 A90213

 usage:  ssoid <flags> <specifications>

 where <flags> <specifications> must be:

     -i1 InputFNam1  (Input full orbital elements file name)
     -i2 InputFNam2  (Input subset orbital elements file name)
     -w  WISEnam     (Name of WISE source file)
     -f  FITSnam     (Name of FITS file for WISE frame)
     -o  OutputFNam  (Output WISE/SSOID associations file name)
     -s  Sx Sy Sz    (Sun-to-Spacecraft vector components at
                      epoch of WISE frame, J2000, AU units)
     -v  Vx Vy Vz    (Spacecraft velocity vector components at
                      epoch of WISE frame, J2000, AU/day units)
     -om MDnam       (Meta-Data output file name; optional)
     -n  NLnam       (Namelist input file name; optional)
     -c  ChiSqMax    (Maximum 2-D chi-square value to accept
                      association; optional; default = 16)
     -d  DistMax     (Distance for coarse search for matches,
                      arcsec; optional; default = 10)
     -m  MaxAng      (Maximum angle from frame center to SSO
                      to consider, deg; optional; default = 2)
     -h  HalfDiag    (Half-diagonal of frame, arcsec; optional;
                      default = 2000)
     -xL             (Omit one-way light time correction [to
                      support testing with simulation data];
                      optional; default = F)
     -cn ColMin      (Minimum SSO band-frame column coordinate
                      to be considered inside array; optional;
                      default = 1)
     -cx ColMax      (Maximum SSO band-frame column coordinate
                      to be considered inside array; optional;
                      default = 1016)
     -rn RowMin      (Minimum SSO band-frame row coordinate to
                      be considered inside array; optional;
                      default = 1)
     -rx RowMax      (Maximum SSO band-frame row coordinate to
                      be considered inside array; optional;
                      default = 1016)

 Either "-i1" or "-i2" must be specified but not both; "-w",
 "-f", "-o", "-s", and "-v" are required.
```

The ssoid module can operate with either the full orbital-elements file or the subset file prepared by the ssoinit module; one or the other of these, but not both, must be specified via the "`–i1`" or "`–i2`" options, respectively. The latter allows a reduction in CPU time of typically a factor of 30. This module processes a single frameset, and so a single FITS file is needed, and the corresponding list of point sources is also needed. The former is used to establish the mapping between celestial coordinates and WISE U-scan and band-frame coordinates. The latter file supplies the WISE point sources to be sought for matches to the SSO positions. The "`–o`"option specifies the name of the main output file, which has entries for all SSOs that fall into the band-frame area and includes WISE information and derived parameters for SSOs with WISE matches.

The components of the sun-to-spacecraft vector are supplied via the "-s" option, and the corresponding velocity vector is specified via the "-v" option. These are J2000 vectors in units of AU and AU/day, respectively, at the frame epoch, which is also the epoch of the FITS file. The latter may be for any WISE band; whatever band is used determines the band-frame mapping of SSO position coordinates for deciding whether each tested SSO really fell inside the array area.

All other command-line specifications are optional. The "-om" option allows an output meta-data file to be generated; this contains information regarding the number of objects processed, match rate, etc. An example is shown below.

```
\ ssoid Meta-Data (SSMD) file - WISE/SSO association parameters
\ Generated by ssoid vsn 1.2  A90126 on 27-01-09 at  9:01:02
|    name           | band | hdrname | type | value                 | comment                                                     |
|    c              | i    | c       | c    | c                     | c                                                           |
 ssoid:ChiSqMx        0     CHISQMAX   r             1.600000E+01 Max 2-D position chi-square to accept WISE/SSO match
 ssoid:DistMax        0     DISTMAX    r             1.000000E+01 Coarse search distance in arcsec
 ssoid:Nin            0     NIN        i                     4171 No. of orbital element sets read
 ssoid:Nsso           0     NSSO       i                       37 No. of SSOs in WISE frame
 ssoid:Nmch           0     NMCH       i                        5 No. of WISE/SSO matches
 ssoid:NCnfzd         0     NCNFZD     i                        0 No. of SSO-WISE confused matches
 ssoid:Nmiss          0     NMISS      i                       32 No. of unmatched SSOs
 ssoid:RatMch         0     RATMCH     r             1.351351E-01 Match rate, NMch/Nsso
 ssoid:AvgDX          0     AVGDX      r    -3.349664878845215E+00 Average X offset in arcsec
 ssoid:AvgDY          0     AVGDY      r    -1.279651689529419E+00 Average Y offset in arcsec
 ssoid:SigDX          0     SIGDX      r     4.853681981853966E+00 Sigma of X offset in arcsec
 ssoid:SigDY          0     SIGDY      r     3.825556837101503E+00 Sigma of Y offset in arcsec
 ssoid:AvgChSq        0     AVGCHSQ    r     5.106672966480255E+00 Reduced 2-D chi-square
 ssoid:Nstat          0     NSTAT      i                        5 No. of clean matches for stats
```

The "-n" option allows a namelist file to be specified (see section 2.2.4). The other options are as documented in the tutorial display.

2.2.2 FITS Input

The ssoid module reads FITS input to obtain coordinate mapping information and the epoch of the observation.

2.2.3 WISE Source Input

The ssoid module reads a table file of WISE point sources which are to be matched with the SSO predictions (SIS pht01: WPHot Output Photometry Table, WSDC D-I107).

2.2.4 Orbital Elements Input

The ssoid module reads orbital elements input in one of two optional ways: (a.) the format used by the ssoinit module (SIS sso01: SSO Orbital Elements File, WSDC D-I138), which covers the full set of known SSOs; (b.) the subset file generated by the ssoinit module (SIS sso02: ssoinit/ssoid Intermediate File, WSDC D-I139).

2.2.5 NAMELIST Input

The ssoid module optionally reads NAMELIST input to control table-file column names in the input WISE source table file.The name of the NAMELIST is `ssoidin`. The parameters defined in the NAMELIST are as follows. Default values are shown in nominal case, but the processing is case-insensitive.

Name	Description	Dim	Type	Units	Default
hdrNsrcs	Name of the header parameter that specifies the number of sources	1	C*25	-	'Nsrc'
tblRA	Name of the column parameter that specifies RA	1	C*25	-	'ra'
tblDec	Name of the column parameter that specifies Dec	1	C*25	-	'dec'
tblSigX	Name of the column parameter that specifies 1-sigma RA uncertainty	1	C*25	-	'sigra'
tblSigY	Name of the column parameter that specifies 1-sigma Dec uncertainty	1	C*25	-	'sigdec'
tblSigXY	Name of the column parameter that specifies RA/Dec co-sigma uncertainty	1	C*25	-	'sigradec'
tblW1nois	Name of the column parameter that specifies W1 noise	1	C*25	-	'w1sigsk'
tblW2nois	Name of the column parameter that specifies W2 noise	1	C*25	-	'w2sigsk'
tblW3nois	Name of the column parameter that specifies W3 noise	1	C*25	-	'w3sigsk'
tblW4nois	Name of the column parameter that specifies W4 noise	1	C*25	-	'w4sigsk'
tblW1m	Name of the column parameter that specifies W1 magnitude	1	C*25	-	'w1mpro'

Name	Description	Dim	Type	Units	Default
tblW1sgm	Name of the column parameter that specifies W1 magnitude uncertainty	1	C*25	-	‘w1sigmpro’
tblW2m	Name of the column parameter that specifies W2 magnitude	1	C*25	-	‘w2mpro’
tblW2sgm	Name of the column parameter that specifies W2 magnitude uncertainty	1	C*25	-	‘w2sigmpro’
tblW3m	Name of the column parameter that specifies W3 magnitude	1	C*25	-	‘w3mpro’
tblW3sgm	Name of the column parameter that specifies W3 magnitude uncertainty	1	C*25	-	‘w3sigmpro’
tblW4m	Name of the column parameter that specifies W4 magnitude	1	C*25	-	‘w4mpro’
tblW4sgm	Name of the column parameter that specifies W4 magnitude uncertainty	1	C*25	-	‘w4sigmpro’

3 Processing

3.1 SSOINIT Processing

The ssoinit module begins by reading the command-line input, verifying that all required specifications have been made, and verifying that all input files exist. If meta-data output was requested, then that file is opened and initialized. If any unrecognized specifications were made, an error message is generated and processing terminates with a 64 return code.

The three FITS input files are opened, and the following header parameters are read for each and stored in three-element arrays: CRVAL1, CRVAL2, and DATE_OBS. The last parameter string is converted to Julian Date and used as the corresponding frame epoch. The other parameters are used to compute the scan axis and azimuthal scan swath; since these values are the RA and Dec of each frame center, corresponding unit vectors can be constructed:

$$\begin{aligned} X_i &= \cos\alpha_i \cos\delta_i \\ Y_i &= \sin\alpha_i \cos\delta_i \\ Z_i &= \sin\delta_i \end{aligned} \tag{1}$$

where i = 1 to 3 for the three epochs T_i.

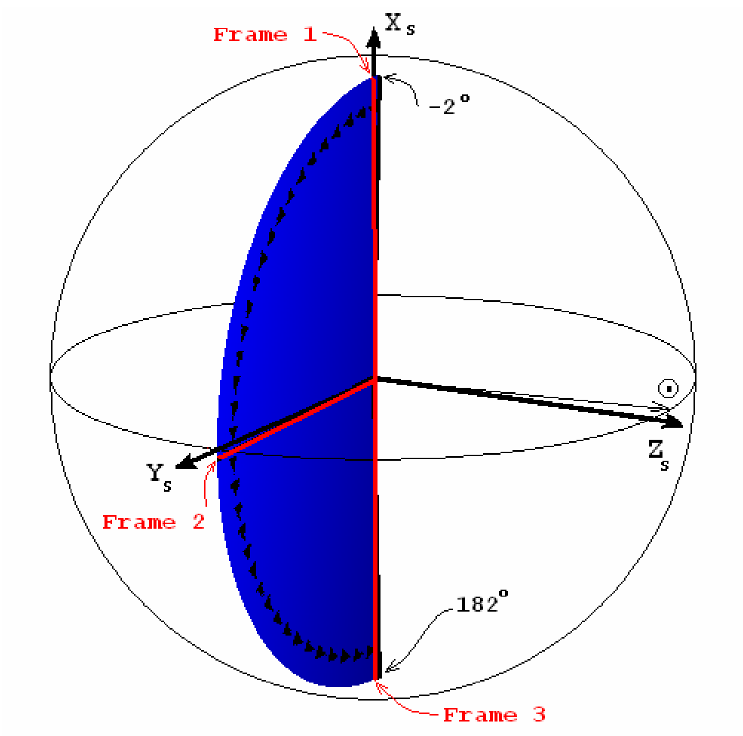
If three-epoch ephemerides were requested for a list of SSOs (see section 2.1.1, command-line options “`-i2`” and “`-o2`”), then the input file containing the object names is opened, the number of lines is counted, memory is allocated for an array of the 35-character names and for a logical flag for each, `GotEphem`, that indicates whether the object has been found in the orbital-elements input and processed, the file is rewound, the names are read into memory, and `GotEphem` is set to `F` for each. The output file is then opened, and the header is written.

The scan axis is computed by averaging two estimates, each obtained via the cross product of two unit vectors shown in Equation 1:

$$\begin{aligned}
\hat{U}_i &\equiv (X_i, Y_i, Z_i) \\
\hat{S}_1 &= \frac{\hat{U}_1 \times \hat{U}_2}{\left|\hat{U}_1 \times \hat{U}_2\right|} \\
\hat{S}_2 &= \frac{\hat{U}_2 \times \hat{U}_3}{\left|\hat{U}_2 \times \hat{U}_3\right|} \\
\hat{S} &= \frac{\hat{S}_1 + \hat{S}_2}{\left|\hat{S}_1 + \hat{S}_2\right|} \\
\Delta\theta &= \sin^{-1}(\hat{S}_1 \times \hat{S}_2)
\end{aligned} \tag{2}$$

If $\Delta\theta$ > 3 arcsec, a warning message is issued, and the scan-swath half-width W (default 2 degrees; see "`-w`" in section 2.1.1) is increased by $\Delta\theta$.

The scan axis is defined as the pole of the "scan" coordinate system, denoted by a subscript "s":



$$\begin{aligned}
\hat{Z}_s &= \hat{S} \\
\hat{X}_s &= \hat{U}_1 \\
\hat{Y}_s &= \hat{Z}_s \times \hat{X}_s
\end{aligned} \tag{3}$$

The X_s axis points to the center of Frame 1, which is thus the zero point of the azimuthal angle. The "scan swath" is the area whose azimuthal center line is in the X_sY_s plane (hence perpendicular to the scan axis) with a width of $\pm W$ and extending from $-W$ (i.e., below the X_s axis, preceding Frame 1 along the scan) to $+W$ past Frame 3. The illustration on the left shows a scan center line from the north ecliptic pole to the south ecliptic pole; this is the scan-system azimuthal range from 0° to 180°. The "scan swath" therefore extends from -2° to 182° in azimuth and is 4° full width in elevation. For a morning launch, the Z_s scan axis should be generally very close to the spacecraft-sun vector as shown in the illustration, but it may be a few degrees off to allow some variation in which ecliptic meridian is chosen to be scanned. For an evening launch, the two vectors should be close to anti-parallel. In either case, the scan axis should lie very close to the ecliptic plane.

The unit vectors in Equation 3 can be used to form a direction-cosine matrix which is a transformation matrix for computing scan-system coordinates corresponding to any RA and Dec, (α,δ), as follows, where unit vectors will be denoted *u*, subscript "*c*" indicates "celestial" (J2000 equatorial) coordinates, and subscript "*s*" indicates "scan-system" coordinates:

$$
\begin{aligned}
u_{c1} &= \cos\alpha\cos\delta \\
u_{c2} &= \sin\alpha\cos\delta \\
u_{c3} &= \sin\delta \\
\begin{pmatrix} u_{s1} \\ u_{s2} \\ u_{s3} \end{pmatrix} &= \begin{pmatrix} X_{s1} & X_{s2} & X_{s3} \\ Y_{s1} & Y_{s2} & Y_{s3} \\ Z_{s1} & Z_{s2} & Z_{s3} \end{pmatrix} \begin{pmatrix} u_{c1} \\ u_{c2} \\ u_{c3} \end{pmatrix} \\
\theta &= \tan^{-1}\left(\frac{u_{s2}}{u_{s1}}\right) \\
\phi &= \sin^{-1}(u_{s3})
\end{aligned}
\tag{4}
$$

where the azimuthal and elevation angles are θ and ϕ, respectively. The scan swath is the set of all points satisfying the inequalities:

$$
\begin{aligned}
-W &\le \phi \le W \\
\theta_1 - W &\le \theta \le \theta_3 + W
\end{aligned}
\tag{5}
$$

where the "1" and "3" subscripts on θ indicate the azimuthal coordinates of Frame 1 and Frame 3, respectively. The criterion defining whether any given point on the sky is inside the scan swath is simply whether its azimuthal and elevation angles satisfy these two inequalities.

Once the transformation matrix in Equation 4 has been set up, the processing involves computing (α,δ) for each SSO at each of the three epochs, computing the corresponding (θ,ϕ), and applying the test in Equation 5. Objects satisfying that test at any of the three epochs are written to the main output file (specified via "-o" on the command line; see section 2.1.1). In addition, objects that straddle the scan swath in either angle over any two epochs are also written to the main output file. Such output involves the data line from the input orbital elements file, followed by a line containing the following six numbers evaluated at the second epoch: the three components of the spacecraft-centered unit vector to the SSO, the mean motion, the topocentric distance, and the eccentric anomaly.

If output three-epoch ephemerides were selected (the "`-i2`" and "`-o2`" options in section 2.1.1), then each object name is checked against the input list of names, and if it is found, then the information shown at the end of section 2.1.1 is written to the output ephemerides file.

All that remains is the computation of (α,δ) for each SSO at each epoch. This is performed in the same way for each object/epoch combination, and it is performed in both modules, ssoinit and ssoid, so we will describe the case for a single object and a single epoch in Appendix A. There is actually one slight difference between epoch 1 and the other two epochs: once the eccentric anomaly has been found iteratively for epoch 1, it becomes the first estimate for the iteration at epoch 2; similarly, epoch 2 provides the first estimate for epoch 3. This is done to save CPU time by converging the iterative solution for the eccentric anomaly faster, but since it involves merely a better starting estimate for iterations 2 and 3, its interest is strictly computational efficiency and will not be discussed further.

3.2 SSOID Processing

The ssoid module begins by reading the command-line input, verifying that all required specifications have been made, and verifying that all input files exist. If meta-data output was requested, then that file is opened and initialized. If any unrecognized specifications were made, an error message is generated and processing terminates with a 64 return code.

The FITS input file is opened, and the following header parameters are read: CRVAL1, CRVAL2, CRPIX1, CRPIX2, CD1_1, CD1_2, CD2_1, CD2_2, and DATE_OBS. The last parameter string is converted to Julian Date and used as the frame epoch; the other parameters are used in the mapping of RA and Dec to U-Scan coordinates and band-frame coordinates via the PRex subroutines j2k2us, sete2u, setus, and u2fnd2.

A file containing orbital elements is required; this may be the "full" set ("`-o1`"), or the "subset" generated by ssoinit ("`-o2`"). The subset contains only objects in the scan swath, and for each such object, an additional data line exists which contains the spacecraft-centered J2000 unit vector to the object, the mean motion, the topocentric distance, and the eccentric anomaly, all evaluated for the epoch of a frame near the center of the scan being processed (or at least between the first and last frames of the scan).

Memory is allocated for arrays that will hold SSO information. The size necessary for these arrays to hold all relevant SSOs is not known in advance, and so an initial size of 1000 is used, and as SSOs are found in the array area and added to these arrays, if more than 1000 are found (extremely unlikely), the memory is released, the array size is doubled (this may be performed as many times as necessary), and execution reverts to the memory allocation step. Any SSO processing that had been performed is abandoned, and execution begins anew. It is not expected that this recovery procedure will ever be needed in practice, but a warning message will be issued if it ever is used. The arrays store the following information for each SSO found to be inside the array area of the WISE band indicated in the FITS file:

X, Y:	U-scan coordinates (arcsec)
VarX, VarY, VarXY:	Position error covariance matrix elements
ObjID:	35-character name
RA, Dec:	Apparent J2000 RA and Dec (deg)
Rhelio:	Heliocentric distance (AU)
Rtopo:	Topocentric distance (AU)
Phase:	Phase angle (deg)
Vmag:	Visual magnitude
Mu:	Proper motion (arcsec/s)
Theta:	Direction of proper motion (deg E of N)
ErrMaj, ErrMin:	Major and minor axes of uncertainty ellipse (arcsec, 1-sigma)
ErrAng:	Direction of ErrMaj axis (deg E of N)
ChiSq:	WISE Position-match 2-D chi-square
Nmatch:	No. WISE sources with acceptable position-match chi-squares
WISEnum:	WISE source ID for best position match
P(3):	Orbital P vector
Q(3):	Orbital Q vector
PerD:	Orbital perihelion distance (AU)
Ecc:	Orbital eccentricity
PerT:	Time of perihelion passage (JD)
Err:	Orbital element quality estimate
H:	Absolute magnitude
G:	Slope parameter

The P vector is the unit vector from the Sun toward perihelion in J2000; the Q unit vector is orthogonal to the P vector in the orbital plane, right-handed about the orbital angular momentum vector. H and G, the absolute magnitude and slope parameter, are used in computing the phase-dependent visual magnitude using the asteroid phase function developed by Bowell, Harris, and Lumme. The following processing is performed for each SSO in the input file.

If the subset orbital input is being used, then the apparent-position unit vector for the SSO is read and used in a dot product with the array-center unit vector, and if the result is less than cos(*W*), the SSO is immediately discarded. *W* is the scan-swath half-width (default 2 degrees; see "`-w`" in section 2.1.1, or "`-m`" in section 2.2.1, where it is called `MaxAng`). The mean motion is also read from the subset input or computed from the full input.

If the command-line specifications included "`-xL`", then no one-way light-time corrections are made to the SSO orbital solution (this supports testing with simulation data for which this correction was not made). Otherwise, if the subset orbital input is being used, then the topocentric distance is already known to sufficient accuracy, and the one-way light-time correction can be made on the first orbital solution, making the second solution slightly more accurate (this accuracy boost has never been observed to be larger than a milli-arcsec, but it is used aince it is available).

The standard orbital solution, including topocentric corrections, is then obtained as described in Appendix A. This provides the (α,δ) for the SSO at the frame epoch, along with the corresponding unit vector. The latter is used in a dot product with the frame-center, and if the result is less than cos(*W*), the SSO is discarded as described two paragraphs back. Otherwise, the U-scan coordinates of the SSO are obtained via the PRex subroutine j2k2us. The U-scan system is Cartesian in arcsecond units, right-handed, and with the Y axis pointing north. If either coordinate is greater than `HalfDiag` (default 2000 arcsec; see "`-h`"in section 2.2.1). Otherwise the SSO is recorded in the arrays described above, with the ChiSq, NMatch, and WISEnum entries set to -9.9, 0, and 0, respectively.

After the input file has been read and processed completely, the number of SSOs retained in memory is checked; if this is zero, only the number of WISE sources is read from the header of the WISE source file, and the rest of the WISE source I/O is skipped. Otherwise this number is used to allocate memory for WISE sources in arrays containing the following information for each source.

RA, Dec:	J2000 RA and Dec (deg)
X, Y:	U-scan coordinates (arcsec)
VarX, VarY, VarXY:	Position error covariance matrix elements
Noise(4):	Noise in the the four WISE bands
Wmag(4):	Magnitudes in the the four WISE bands
SigWmag(4):	Magnitude uncertainties in the the four WISE bands, 1-sigma
SSOnum:	Index of SSO with which this WISE source is matched

SSOnum is initialized to zero. The average noise in each WISE band is computed for use in output SSO records for which no WISE association was found. Then a loop over all SSOs in memory is performed; for each SSO the following match processing done.

A loop over all WISE sources in memory is performed; for each WISE source, a position comparison to the SSO is made in U-scan coordinates. Subscripts "*w*" and "*o*" will be used to indicate WISE and SSO objects, respectively. First a coarse test is made:

$$\begin{aligned}
&\Delta X = X_o - X_w \\
&if\ \left|\Delta X\right| > D_{\max}\ then\ discard \\
&\Delta Y = Y_o - Y_w \\
&if\ \left|\Delta Y\right| > D_{\max}\ then\ discard
\end{aligned} \tag{6}$$

where D_{max} defaults to 10 arcsec (see "`-d`" in section 2.2.1). SSO/WISE pairs passing this test proceed to the chi-square test:

$$
\begin{aligned}
V_x &= VarX_o + VarX_w \\
V_y &= VarY_o + VarY_w \\
V_{xy} &= VarXY_o + VarXY_w \\
\chi^2 &= \frac{V_y \Delta X^2 + V_x \Delta Y^2 - 2V_{xy} \Delta X \, \Delta Y}{V_x V_y - V_{xy}^2} \\
&\mathit{if}\ \chi^2 > \chi^2_{\max}\ \mathit{then\ reject\ match}
\end{aligned}
\tag{7}
$$

where χ^2_{max} is the position-match threshold (see "-c" in section 2.2.1), with a default value of 16. This value for a 2-D chi-square random variable implies that 99.966% of all true matches should be accepted, i.e., 1 out of every 2981 true matches will be sacrificed in the attempt to avoid false matches.

If the match is acceptable, then the following logic is executed, where N indicates the array index of the SSO, K indicates the array index of the WISE source, Ntotal is a counter for the total number of SSOs that are matched, and Ncnfzd is a counter for the number of SSOs that have more than one match.

```
If Nmatch(N) = 0 then                              { first match for this SSO }
        Nmatch(N) = 1
        Ntotal ⇐ Ntotal + 1
        if SSOnum(K) = 0 then                      { first match for this WISE source }
                WISEnum(N) = K                     { store match parameters }
                ChiSq(N)   = χ2
                SSOnum(K)  = N
        else if ChiSq(SSOnum(K)) > χ2 then         { new match better than previous }
                WISEnum(N) = K                     { store match parameters}
                ChiSq(N)   = χ2
                SSOnum(K)  = N
                WISEnum(SSOnum(K)) = 0             { detach other SSO's connection }
                ChiSq(SSOnum(K))   = 9.9d9
        endif
else                                               { not this SSO's first match }
        Nmatch(N) ⇐ Nmatch(N) + 1
        if (Nmatch(N) .eq. 2) NCnfzd = NCnfzd + 1  { count confused SSOs }
        if ChiSq(N) > χ2 then                      { new match better than previous }
                if SSOnum(K) = 0 then              { first match for this WISE source }
                        WISEnum(N) = K             { store match parameters }
                        ChiSq(N)   = χ2
                        SSOnum(K)  = N
                else if ChiSq(SSOnum(K)) > χ2 then { new match better than previous }
                        WISEnum(N) = K             { store match parameters}
                        ChiSq(N)   = χ2
```

```
                    SSOnum(K)  = N
                    WISEnum(SSOnum(K)) = 0  { detach other SSO's connection }
                    ChiSq(SSOnum(K))   = 9.9d9
                endif
            endif
        endif
```

This is the same algorithm used by 2MASS. In unconfused cases, it provides correct associations. In confused cases, it does not attempt to unravel every possible association. For example, if SSO A has a best-match involving WISE source B, but WISE source B already has a better match to SSO C, no search for a second choice for SSO A is made. Similarly, if SSO A has a better match to WISE source B than SSO C, then SSO C is disconnected from WISE source B and no search for a second choice for it is made. Only best-match associations are kept. The fact that confusion occurred is seen in the Nmatch values. A matched SSO may have Nmatch > 1, and an unmatched SSO may have Nmatch > 0.

It can also happen that a WISE source has two acceptable matches to SSOs, loses its best match to another WISE source, and does not fall back to its second choice SSO.

After all SSOs have been processed for association with WISE sources, another loop over all SSOs is performed to verify that unmatched SSOs had positions inside the real band-frame area, which is generally rotated relative to the U-scan system. The association processing is performed in U-scan because it is the most convenient system for that purpose, but since it is aligned with the local RA and Dec directions, the coarse position test must use a rectangular area whose width is the hypotenuse of the actual array. An unmatched SSO should not be considered "missed" if it fell into a part of the U-scan area not overlapped by the rotated band-frame. So the positions of all unmatched SSOs are transformed to band-frame coordinates (using the PRex subroutine u2fnd2), and to be considered truly "missed", the X coordinate must lie between `ColMin` and `ColMax` (defaults 1 and 1016, resp.; see "`–cn`" and "`–cx`" in section 2.2.1), and the Y coordinate must lie between `RowMin` and `RowMax` (defaults 1, and 1016, resp.; see "`–rn`" and "`–rx`" in section 2.2.1). Only matched and truly "missed" SSOs are written to the output table file.

4 Output

4.1 SSOINIT Output

4.1.1 Orbital Element Subset Output

The orbital-element information for SSOs inside the scan swath is written to the file whose name is specified on the command line via the "`-o`" option (SIS sso02: SSO ssoinit/ssoid Intermediate File, WSDC D-I139).

4.1.2 Three-Epoch Ephemerides File

If three-epoch ephemerides were requested (command-line options "`-i2`" and "`-o2`"), then these are written to the specified output file (SIS sso03: SSO Three-Epoch Ephemerides File, WSDC D-I140).

4.2 SSOID Output

4.2.1 SSO/WISE Associations Output File

The ssoid module always generates a file for SSO/WISE associations, even if there are no SSOs and/or WISE sources in the array area (SIS sso04: SSO/WISE Associations File, WSDC D-I141).

4.2.2 SSO Meta-Data Output File

If a meta-data file was requested via the command-line option "`-om`", then the ssoid module always generates one, even if there are no SSOs and/or WISE sources in the array area (SIS sso05: SSO Meta-Data File, WSDC D-I142).

5 Testing

TBS

Appendix A   SSO Orbit and Apparent Position Computation

Orbit computation is performed by the same algorithm as the 2MASS DTPGM program. A complete description will be supplied in this SDS when the V3 version is released.