

COMBINING SHORT ARC OPTICAL TRACKS FOR IMPROVED SSA

Thomas M. Johnson

Analytical Graphics, Inc., 220 Valley Creek Blvd., Exton PA 19468 USA, tj@agi.com

ABSTRACT

We present a method for analyzing short duration optical tracks of Earth orbiting objects and combining them into longer duration tracks. This allows for a high quality initial orbit determination fit and improved observation association, particularly when working with tracks on closely spaced objects. The method takes advantage of the locally linear behavior of the measurements in each track to identify pairs of tracks that fit well together. These pairings are used to build a directed acyclic graph (DAG) of all combinations of the tracks being analyzed. The longest path through the DAG identifies a set of short tracks that can be combined into a longer one. These tracks are then removed from consideration and the process is repeated. The method is in operational use in the Commercial Space Operations Center (ComSpOC) where it is applied to data from multiple sensor networks. Results from interesting operational cases such as the SHIJIAN-17 proximity operations are presented.

1 INTRODUCTION

The Commercial Space Operations Center (ComSpOC) operated by Analytical Graphics Inc. provides commercial Space Situational Awareness (SSA) services. Our primary interest is maintaining a high accuracy catalog of RSOs that are potential threats to spacecraft operating near geostationary orbit. This is loosely defined as RSOs that pass through the region of space centered around the classic geostationary orbit plus or minus 800 km in altitude and with an inclination less than 15 degrees. This encompasses RSOs in geostationary orbit, drift orbits, GEO disposal orbits, and geostationary transfer orbit or other similar high eccentricity orbits.

The ComSpOC owns and operates a worldwide network of 13 dedicated telescopes used for optical tracking of Earth orbiting resident space objects (RSO). Additional tracking data (optical, radar, time difference of arrival, frequency difference of arrival) is provided by our sensor partners [1].

The tracking data is processed to produce orbit solutions on the RSOs and derived data (e.g. conjunction analysis, maneuver calibration, maneuver pattern of life, sensor calibration) for our subscribers. This paper focuses on the optical tracking data processing pipeline and one of the algorithms we use. The particular focus will be on the

data produced by the ComSpOC owned telescopes but the same algorithm is applied to partner tracking data as well.

1.1 Terminology

We use the definitions in Tab. 1:

Table 1. Definitions

Term	Definition
Observation or Measurement	A measure of an RSO position such as right ascension (RA) or declination (DEC).
Observation Set	A collection of observations obtained at the same time and corresponding to the same RSO. Typically (time, RA, DEC).
Track	A time ordered collection of observation sets from a single sensor for a single RSO.
Association	The process of mapping a track to a RSO.
Uncorrelated Track (UCT)	A track that cannot be associated to an RSO in the catalog.

1.2 Telescope Operations

The ComSpOC owned telescopes are tasked to collect tracking data using the modes described in Tab. 2.

Table 2. ComSpOC Telescope Operating Modes

Mode	Description
Sweep	Move from horizon to horizon while maintaining a desired declination angle off the equator and stepping in longitude. The horizon limits occur at a minimum elevation angle of 20-30 degrees (site specific). A GEO sweep assumes a declination of zero and walks across the GEO belt.
Survey	Step through a sequence of declinations at a given longitude then steps in longitude. This allows us to survey a region of space
Targeted	Iterate through a list of RSOs, evaluating which ones are potentially visible, and point at their current location based on two-line element sets (TLEs) generated by the ComSpOC.

Operationally the bulk of the time is spent in either sweep or targeted mode. A GEO sweep takes ≈ 20 minutes.

Each time the telescope moves to a new location it stares

in sidereal mode and collects an image stack at a rate of 4-5 Hz. Each image stack is processed on site to extract the time, right ascension (RA), declination (DEC), and visual magnitude of RSOs in the image. An observation set for each RSO in each image is generated. Tracks for each RSO in the image stack are created from the observation sets. The RSO in each track is identified using an ID number that is unique within an image cube. Multiple image stacks may be obtained before stepping to the next pointing location.

The rapid frame rate allows us to detect RSOs moving at high angular rates and supports light curve analysis on rapidly tumbling objects.

A single telescope generates over 1 TB of imagery a night. It's not practical to send the images back to the ComSpOC in real-time, therefore after being processed into tracks they are archived on site for a few days to facilitate quality control or more detailed analysis.

1.3 SSA Processing

Text files containing the astrometric and photometric measurements are sent to the ComSpOC for standard SSA processing such as observation association (OA), orbit determination (OD), maneuver processing (MP), and uncorrelated track (UCT) processing.

The input tracks are assumed to be atomic – a track only contains observation sets for one RSO. This is true (>98%) for the ComSpOC telescopes but not necessarily for data from partner telescopes. We execute a separate algorithm to identify “dirty” tracks containing multiple RSOs and split them into individual atomic tracks.

Association is done at the track level. Tracks from the ComSpOC telescopes are not associated until within the ComSpOC itself. Tracks from partner sensors may have already been associated by the partner. All tracks are run through the OA process to make an initial association or validate a pre-existing association and re-associated as necessary. Initial association is done based on the angular position and rate. Final association is performed in Mahalanobis space using a full ephemeris and covariance.

1.4 Motivation

The rapid frame rate means the tracks from the ComSpOC telescopes are very short (2-3 seconds). A typical sensor produces 800-2000 tracks per day. Many of the tracks correspond to the same RSO but have different IDs since they came from a different image stack or different sweep.

The SSA software must address the following challenges:

- Short tracks degrade the initial association process because the angular rate uncertainties

are high. The track may match multiple RSOs.

- UCTs must be analyzed to determine if they represent a new RSO. The complexity of this process is a function of the number of UCTs.
- Angles only initial orbit determination methods often fail on short tracks due to the limited geometric diversity.

A method to merge short tracks into longer ones assists with all of these. The ideal merge algorithm is simple, fast, and fails gracefully.

2 RELATED WORK

Solutions to the merge problem typically fall into two categories: classic astrodynamics or a pure geometrical approach. The astronomy community has extensive experience applying astrodynamics to map uncorrelated tracks to an asteroid or comet and perform initial orbit determination [2]. This was examined in detail for geostationary satellites in [3]. These approaches try to solve the IOD problem and struggle because of the short track durations. We are primarily concerned with the “linkage” attributable (from [2]) between tracks.

The linkage problem for asteroids was examined in [4] and is a nice combination of geometry and astrodynamics. They applied a quadratic fit in RA and DEC vs. time to identify tracks that linked together.

Figs. 1 and 2 show 12 tracks for the ANIK satellite cluster (ANIK F1, ANIK F1R, and ANIK G1) and its neighbor EHOSTAR 17 (top right cluster in Fig. 2). They are drawn in the inertial frame (ICRF) and a pseudo Earth Fixed frame (by subtracting the sidereal time of Greenwich from the RA values to form a pseudo Greenwich Hour Angle (GHA)). Three image stacks were collected over 30 seconds with each RSO present in each stack.

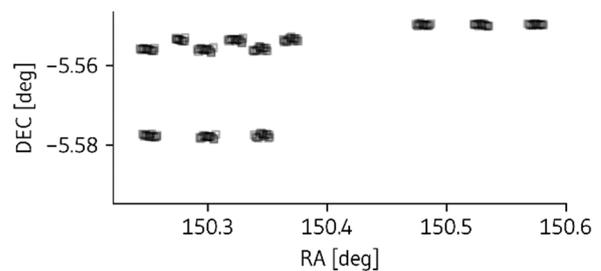


Figure 1. ANIK Cluster in the Inertial Frame

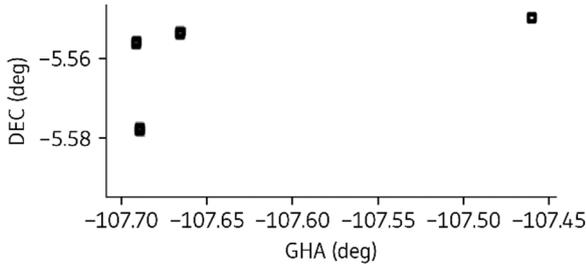


Figure 2. ANIK Cluster in the Pseudo-Fixed Frame

It's not obvious in the inertial frame that there are four RSOs. The pseudo-fixed frame clearly shows 4 clusters of tracks, suggesting that a cluster detection algorithm such as K-Means or DBSCAN [5] might be useful. It's not uncommon for multiple RSOs to have tracks that stack on top of each other (particularly when a non-GEO RSO passes close by). This can defeat many of the clustering algorithms.

It's convenient to draw the tracks in the two astrometric dimensions (RA/DEC or GHA/DEC) but in doing so we have ignored two others – time and visual magnitude. Use of the latter for observation association was examined in [6]. They were able to additionally leverage Two Line Element (TLE) sets for each of the spacecraft to assist with associating the tracks. For our application we can't assume we have any apriori state and must work with the raw tracks.

The tracks plotted in three dimensions (Fig. 3) appear as lines, dominated by the large RA rate from the Earth's rotation since the sensor is ground-based). Tracks for the same RSO appear co-linear and suggests a line finding algorithm is appropriate.

Finding lines is a common problem in computer vision applications. We examined using a Hough transform [7, 8] to map each measurement into the Hesse normal form of a line.

$$\rho = RA \cos\theta + DEC \sin\theta \quad (1)$$

The normal form is preferred over slope-intercept form to avoid the singularity in the slope for a vertical line. Fig. 4 shows several tracks in the inertial frame and the equivalent Hough transform. The RA and DEC values in the figure are relative to the overall minimum values. Each observation set produces a unique sinusoidal curve in (ρ, θ) space. The measurements from co-linear tracks show up as intersecting curves at common (ρ, θ) coordinates. The intersection points occur primarily near $\theta=90$ degrees, consistent with the "horizontal" appearance of the original tracks.

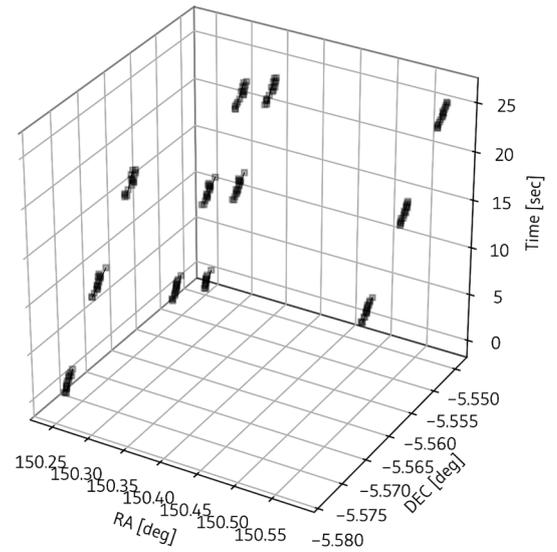


Figure 3. ANIK Cluster in Three Dimensions

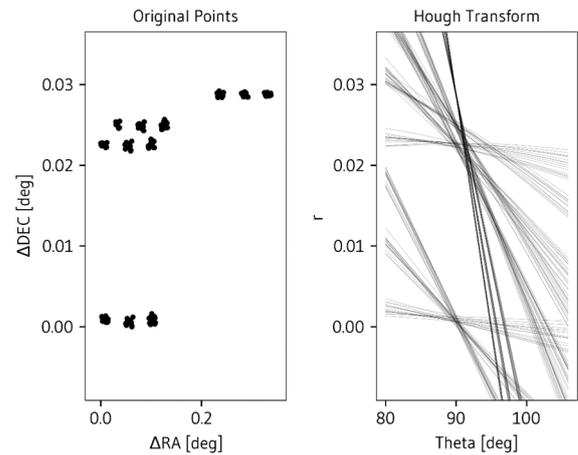


Figure 4. Hough Transform of ANIK Cluster

Classic applications of the Hough transform establish an accumulator grid of pixels in (ρ, θ) space and increment a pixel each time a curve passes through it. Pixels with peak values mark unique (ρ, θ) coordinates and thus the corresponding lines in the input data. This approach was considered but assessed as too inefficient because of the number of (ρ, θ) samples that must be obtained. The accumulator also does not retain memory of which measurements (and thus tracks) contributed to a particular hot spot – a critical element for our application when we want to combine the original tracks.

The equivalent problem in three dimensions is finding lines in point clouds. Reference [9] applied the Hough transform in three dimensions to process LIDAR data, but this approach has the same problem of using an accumulator. Reference [10] applied a 3D beamlet

transform to finding linear structures in galaxies.

The Hough and beamlet transforms find lines, but our data may not always be globally linear. Therefore we pursued an algorithm that only required local linearity.

3 ALGORITHM

3.1 Assumptions

The track merging algorithm requires the following:

- All tracks are from the same sensor.
- Each track is atomic – containing data for only one RSO.
- An individual track is short enough that it can be modeled as a line in time, RA, DEC space.
- Two tracks can be merged only if combined they can still be modeled as a line (they are locally linear).

A long stare (e.g. an hour) at a single RSO may violate the linear assumption but in practice the sensors automatically break the stare into shorter tracks so they can be delivered in an operationally relevant time frame rather than waiting until the end of the stare.

Two tracks can be evaluated for merging if:

- They don't overlap in time. If they overlap they must be different RSOs due to the atomic track requirement.
- The gap time between tracks (start time of the late track minus end time of the early track) is less than a specified threshold T . T is chosen to be small enough to enforce the "locally linear" assumption.

3.2 Overview

Fig. 5 illustrates some sample tracks. Using the syntax $[X, Y]$ to refer to a merge of tracks X and Y , then:

- $[A, B]$ and $[C, D]$ are invalid because they overlap in time.
- $[A, E]$ is invalid because the tracks are too far apart in time.
- $[A, C]$, $[A, D]$, $[B, C]$, $[B, D]$, $[B, E]$, $[C, E]$, and $[D, E]$ are valid pairs and should be evaluated.

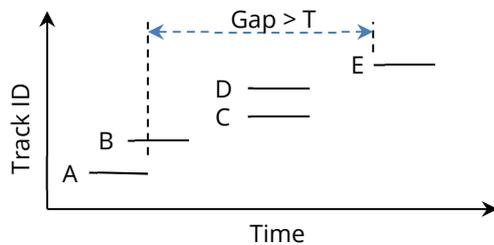


Figure 5. Track Merging Rules

This is expressed more compactly as a matrix in Fig. 6

where the dark blocks are valid track pairs. Only the upper triangular half is relevant since the matrix is symmetric.

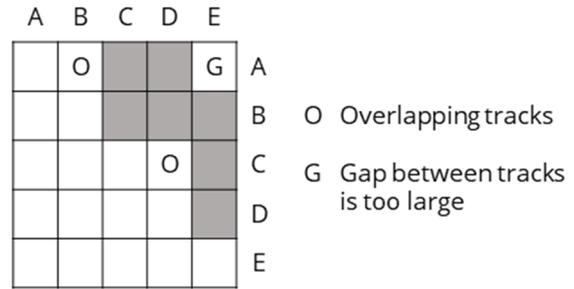


Figure 6. Track Merge Matrix

If $[X, Y]$ can be merged and $[Y, Z]$ can be merged then we can combine them to form a multi-track $[X, Y, Z]$. Using this property *and assuming that each of the valid pairs from Figs. 5 and 6 can be merged*, then the potential multi-track solutions are shown in Tab. 3. Individual tracks are the worst case solution (nothing can be merged).

Table 3. Initial Multi-Tracks

Length 3	Length 2	Length 1
$[B, C, E]$	$[A, C]$	$[A]$
$[B, D, E]$	$[A, D]$	$[B]$
	$[B, C]$	$[C]$
	$[B, D]$	$[D]$
	$[B, E]$	$[E]$
	$[C, E]$	
	$[D, E]$	

We prefer longer (in length, not time) multi-tracks. We choose $[B, C, E]$ (arbitrarily, see section 4 for more discussion on this). The algorithm is greedy – once we use a track as part of a multi-track then it is no longer available and other multi-track solutions that contain it are removed from consideration. Removing tracks $B, C,$ and E leaves us with the solutions in Tab. 4.

Table 4. Multi-Tracks after One Iteration

Length 2	Length 1
$[A, D]$	$[A]$
	$[D]$

The process of selecting the longest solution and removing its nodes continues until we have exhausted all possible solutions – including single tracks. The final multi-tracks are $[B, C, E]$, and $[A, D]$.

In the worst case the tracks under evaluation may be non-

linear or simply not be the same RSO at all. In this case the merge process will simply reproduce the input tracks.

3.3 Implementation

We calculate a linear fit of RA and DEC against time for each individual track. We initially tried using the track start and stop points to directly solve for the equation of a line – which performed quite poorly due to measurement noise. We also tried standard least squares regression but it performed poorly in the presence of outliers. We switched to a Thiel-Sen estimator [11] which was quite effective.

The measurement residuals from the linear fit and their standard deviations σ_{RA} and σ_{DEC} are calculated and stored with each track as a measure of how linear the track is. We don't assume that the measurement white noise uncertainty is the same in RA and DEC but we do treat them as independent. We calculate the root sum square (RSS) of the standard deviations to form an overall track uncertainty.

$$\sigma_T = \sqrt{\sigma_{RA}^2 + \sigma_{DEC}^2} \quad (2)$$

We determine whether a pair of tracks can be merged by assessing how well the two tracks combined can be modeled as a line. We repeat the linear fit process (Thiel-Sen) using the measurements from both tracks. The measurement residuals are calculated against the line model and their standard deviations $P\sigma_{RA}$ and $P\sigma_{DEC}$. We use P to indicate the pair and not the individual tracks. These are RSSed to form an overall track pair uncertainty.

$$P\sigma_T = \sqrt{P\sigma_{RA}^2 + P\sigma_{DEC}^2} \quad (3)$$

We calculate the ratio of the pair uncertainty to the RSS of the track uncertainties for tracks 1 and 2.

$$R_{12} = \frac{P\sigma_T}{\sqrt{\sigma_{T1}^2 + \sigma_{T2}^2}} \quad (4)$$

This normalizes the pair fit quality relative to the individual track fit quality and becomes unitless. R is our metric for evaluating connectivity between pairs of tracks.

We start with a list of tracks of length N for a sensor, sorted by the start time of each track. We evaluate R via a function on each track T_i against the future tracks T_j in the list and create a $N \times N$ matrix M using the pseudo-code below.

```

M = 0
for I = 1 to N do
  for J = I+1 to N do
    if  $T_i$  and  $T_j$  is a valid pair then
       $M[i][j] = R(T_i, T_j)$ 
    end
  end
end

```

The valid pair decision follows the rules outlined in section 3.1. We convert M to a Boolean connectivity matrix C by applying a threshold check against each element of the matrix. The value of 1.3 was determined empirically.

$$C_{ij} = M_{ij} \leq 1.3 \quad (5)$$

Applying this process to the ANIK data from Fig. 4 generates the metric and connectivity matrices in Figs. 7 and 8.

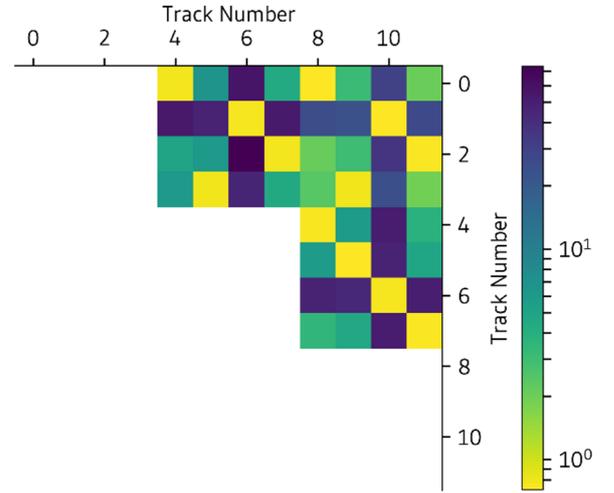


Figure 7. ANIK Metric Matrix

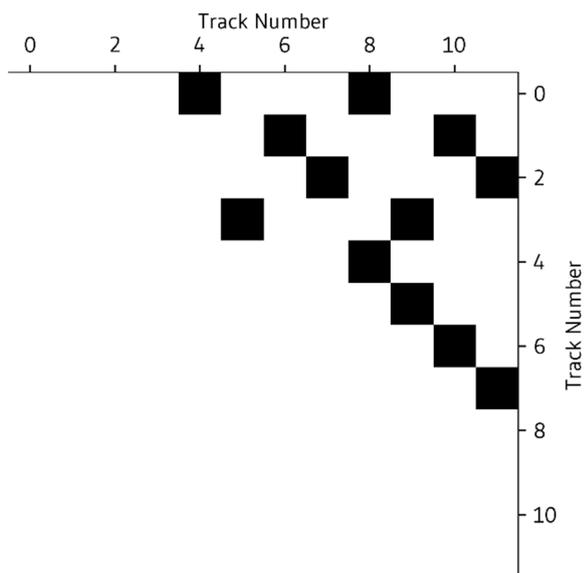


Figure 8. ANIK Connectivity Matrix

When multiple tracks are co-linear a track T_i will have a nice fit with several tracks. The goal is to extract sequences of connected tracks to form multi-tracks. We convert our connectivity matrix to a Directed Acyclic Graph (DAG) using the NetworkX Python package [12]. Each node in the DAG represents a track and two nodes I and J have an edge when C_{ij} is true.

Using the DAG we extract the longest connected path and combine these tracks into a single multi-track. The nodes used are then removed from the DAG. This process repeats until all nodes in the DAG have been removed and all of the original tracks have been accounted for. The resulting list of multi-tracks is written to disk for use by the rest of the SSA processing chain.

The construction of the M and C matrixes is useful for visualization and debugging but are not necessary in practice – we can build the DAG directly from the tracks being assessed.

The track merging process can be executed as frequently as desired. The longer you wait the more tracks are available to be merged. But this introduces an inherent delay in making the multi-tracks accessible to the rest of the processing chain. Operationally we execute it every 15 minutes.

4 RESULTS

Applying the algorithm to our ANIK cluster generates the four merged tracks in Fig. 9. We have drawn connecting lines between the first and last point of the merged tracks to highlight which tracks were merged. In this case the merged tracks appear to be locally and globally linear.

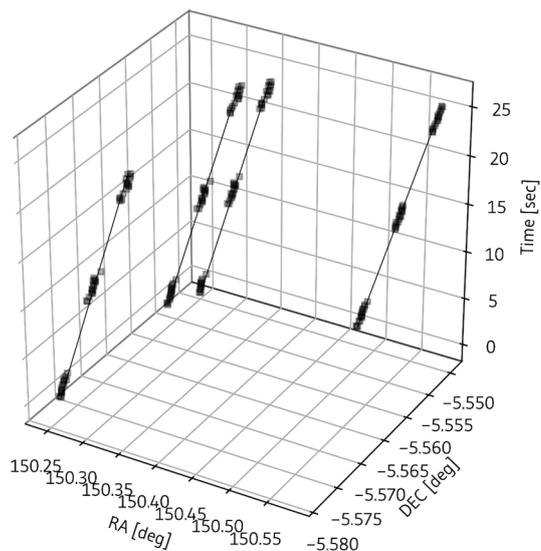


Figure 9. Merged ANIK Tracks

Fig. 10 (at the end of this paper) shows the result from processing a stare at GSAT-6A while it was in geosynchronous transfer orbit. 116 tracks were merged into 1 multi-track. The line connecting the end points of the multi-track is not co-linear with the individual tracks, indicating that the local linear model allows us to build a global non-linear track.

Fig. 11 shows 2 multi-tracks after merging 99 tracks (across 15 minutes) from SHIJIAN-17 and CHINASAT 1C on 22 Jul 2018 when they were within 8 km of each other. We used the pseudo-fixed frame to clearly show the proper separation.

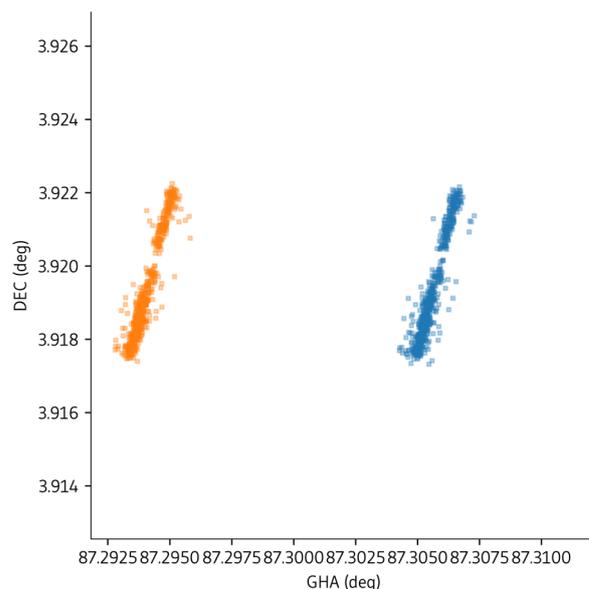
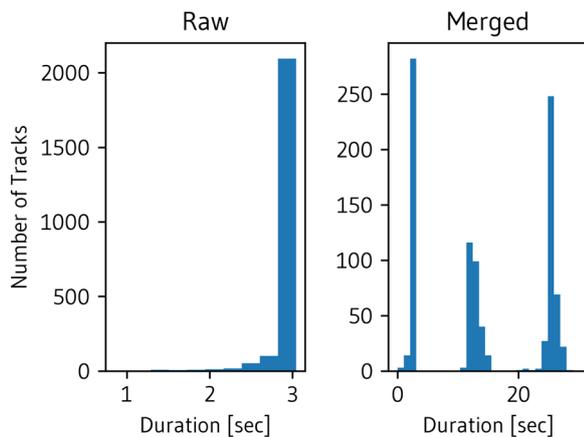


Figure 11. Merged SHIJIAN-17 and CHINASAT 1C Tracks

We have seen occasions where the algorithm has incorrectly merged tracks. This can occur when the input tracks are close together, near parallel, and have the same start and stop times. This happens when two RSOs are in the same frame and nearly identical orbits. The pair-wise fitting process can create local cross connections between the two RSOS. The “longest path” philosophy can’t discriminate between the cross-connection and the proper connection. A more robust path validation is required to address this.

Fig. 12 shows the histogram of track durations before and after the merge process for one sensor in one night. We reduced 2309 tracks to 1008 multi-tracks and extended the track durations. The default time gap threshold of 15 minutes was used.



The merge algorithm has been in operations for the last 9 months with great success. We have seen a reduction in the OA processing time and fewer mistags - reducing the amount of time spent evaluating possible maneuvers.

5 FUTURE WORK

We plan on investigating a number of improvements to the merge process.

- Update the connectivity model to include the photometric measurements (similar to [6]).
- Update the overall path selection process to use a quadratic fit test similar to [4].
- Assigning an objective function value to the edges of the DAG and look for connected sets that minimize the objective function.
- Performing merges over longer periods of time to retrofit tracks together even after they have been processed by the SSA software.

6 REFERENCES

1. Hall, R., Johnson, T. (2016). Commercial SSA Catalog Performance. AMOS Conference Sep. 2016.
2. Gronchi, G. F. (2004), “Classical and modern orbit

determination for asteroids,” IAU Colloquium No. 196: Transits of Venus: New Views of the Solar System and Galaxy, ed. by D.W. Kurtz, Preston, UK: Cambridge University Press, pp. 293–303.

3. Siminski, J. (2016). Object Correlation and Orbit Determination for Geostationary Satellites Using Optical Measurements, Universitätsbibliothek der Universität der Bundeswehr München.
4. Kubica, J., et. al. (2007). Efficient intra- and inter-night linking of asteroid detections using kd-trees. Online at <https://arxiv.org/abs/astro-ph/0703475> (as of 12 January 2019).
5. <https://scikit-learn.org/stable/modules/clustering.html> (as of 12 January 2019).
6. Dao P., Heinrich-Josties E., & Boroson, T. (2016). Automated Algorithms to Identify Geostationary Satellites and Detect Mistagging using Concurrent Spatio-Temporal and Brightness Information. AMOS Conference Sep. 2016.
7. Duda, R. O. and P. E. Hart (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures. In *Comm. ACM*, Vol. 15, pp. 11–15.
8. Rahmdel P., Comley R., Shi D., & McElduff S. (2015). A Review of Hough Transform and Line Segment Detection Approaches. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, Science and Technology Publications, Lda., pp. 411-418.
9. Jeltsch, M., Dalitz, C. & Pohle-Fröhlich, R (2016). Hough Parameter Space Regularisation for Line Detection in 3D. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Volume 4, Science and Technology Publications, Lda., pp. 345-352.
10. Starck, J.L., Martínez, V.J., Donoho, D.L. et al. (2005). Analysis of the Spatial Distribution of Galaxies by Multiscale Methods. In *EURASIP Journal on Applied Signal Processing* 200: 15, Hindawi Publishing Corporation, pp. 2455–2469.
11. https://www.wikipedia.org/wiki/Theil-Sen_estimator (as of 12 January 2019).
12. Aric A. Hagberg, Daniel A. Schult and Pieter J. Swart, “Exploring network structure, dynamics, and function using NetworkX”, in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Gael Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, Aug 2008.

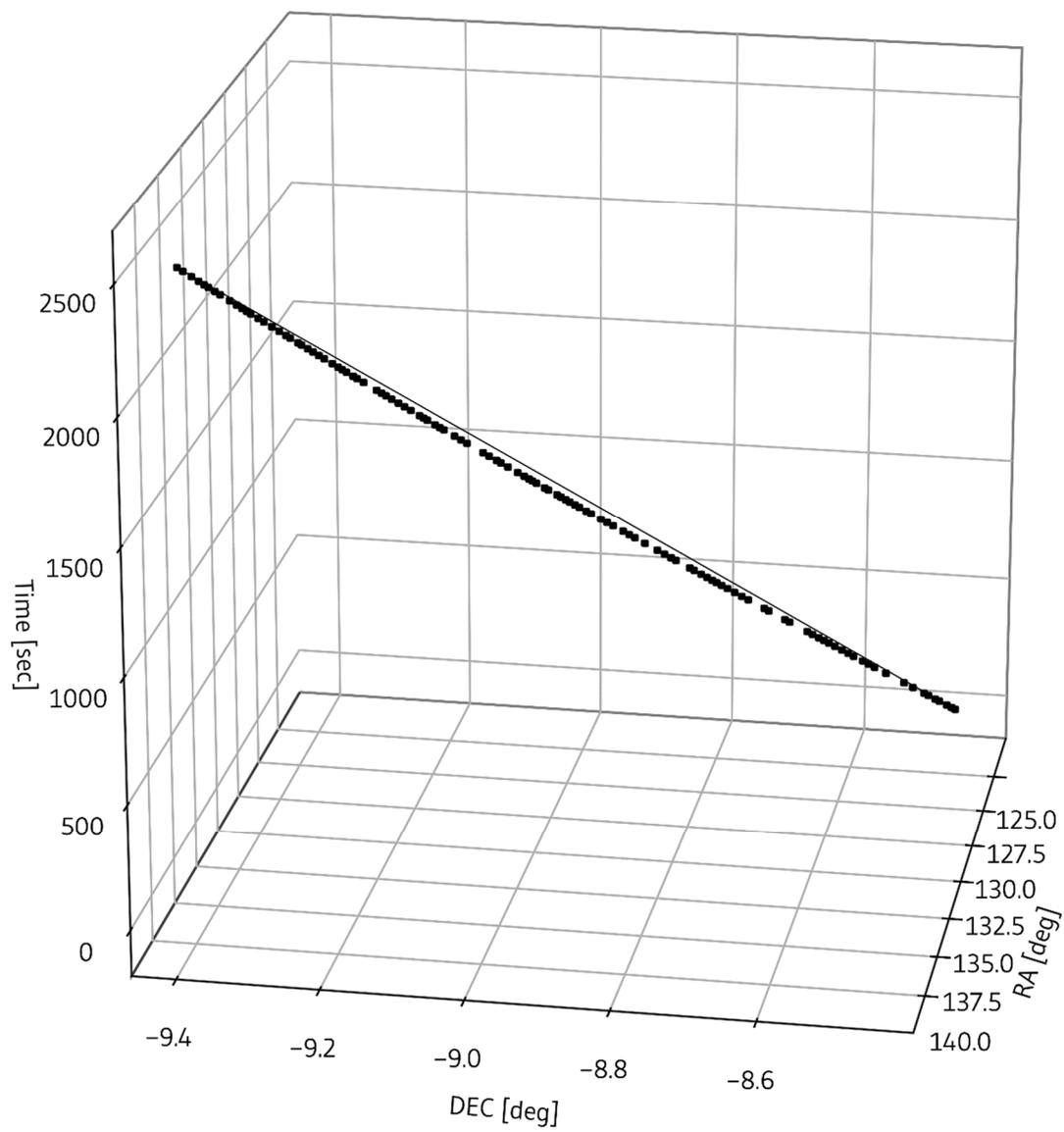


Figure 10. Merged GSAT-6A Tracks