THOR: An Algorithm for Cadence-independent Asteroid Discovery

Joachim Moeyens\textsuperscript{4}, Mario Jurić\textsuperscript{4}, Jes Ford\textsuperscript{4}, Dino Bektešević\textsuperscript{4}, Andrew J. Connolly\textsuperscript{4}, Siegfried Eggl\textsuperscript{4}, Zeljko Ivezić\textsuperscript{4}, R. Lynne Jones\textsuperscript{4}, J. Bryce Kalmbach\textsuperscript{4}, and Hayden Smotherman\textsuperscript{4}

Department of Astronomy and the DiRAC Institute, University of Washington, 3910 15th Avenue NE, Seattle, WA 98195, USA; moeyensj@uw.edu

\textit{Received 2019 June 13; revised 2021 May 19; accepted 2021 May 21; published 2021 September 15}

Abstract

We present “Tracklet-less Heliocentric Orbit Recovery” (THOR), an algorithm for linking of observations of Solar System objects across multiple epochs that does not require intranight tracklets or a predefined cadence of observations within a search window. By sparsely covering regions of interest in the phase space with “test orbits,” transforming nearby observations over a few nights into the corotating frame of the test orbit at each epoch, and then performing a generalized Hough transform on the transformed detections followed by orbit determination filtering, candidate clusters of observations belonging to the same objects can be recovered at moderate computational cost and with little to no constraint on cadence. We validate the effectiveness of this approach by running on simulations as well as on real data from the Zwicky Transient Facility (ZTF). Applied to a short, two-week slice of ZTF observations, we demonstrate THOR can recover 97.4\% of all previously known and discoverable objects in the targeted (\(a > 1.7\) au) population with five or more observations and with purity between 97.7\% and 100\%. This includes 10 likely new discoveries, and a recovery of an \(e \sim 1\) comet C/2018 U1 (the comet would have been a ZTF discovery had THOR been running in 2018 when the data were taken). The THOR package and demo Jupyter notebooks are open source and available at https://github.com/moeyensj/thor.

Unified Astronomy Thesaurus concepts: Astronomy software (1855); Asteroids (72); Trans-Neptunian objects (1705); Comets (280); Sky surveys (1464)

1. Introduction

The number of Solar System minor planet discoveries is growing rapidly, thanks to the continuation of present-day surveys such as Pan-STARRS (Denneau et al. 2013) and the Catalina Sky Survey (Larson et al. 2003), as well as upcoming surveys such as the Vera C. Rubin Observatory’s Legacy Survey of Space and Time (LSST) (Ivezic et al. 2019) and NEOCam, recently renamed to NEOSurveyor (Mainzer & NEOCam Science Team 2016). In about a decade, the number of known objects will grow from the currently known one million to about six million minor planets. Such an increase in discoveries will enable a higher-resolution look into the dynamical evolution of our Solar System. However, identifying minor planets in survey images and linking their detections into orbits continues to be a challenging problem. First, linking asteroid detections across multiple nights is difficult due to the sheer number of possible linkages, and it is made even more challenging by the presence of false positives (Kubica et al. 2007; Denneau et al. 2013; Veres & Chesley 2017a, 2017b; Jones et al. 2018). Second, the motion of the observer makes the linking problem nonlinear because minor planets will exhibit higher-order motion on the topocentric sky over the course of weeks (Holman et al. 2018). Finally, once potential linkages have been established, they need to be confirmed as possible orbits using computationally expensive orbit determination software. For example, the Vera C. Rubin Observatory estimates it will discover nearly six million Main Belt asteroids that will be observed hundreds of times over the course of its ten-year survey. Naively attempting to link hundreds of millions of asteroid detections over a ten-year period is not computationally feasible (Jones et al. 2018).

To make the linking problem more computationally tractable, surveys that aim to discover minor planets focus on constructing “tracklets”: two-dimensional sky-plane motion vectors consisting of two or more detections spaced typically 20–90 minutes apart that constrain the direction and rate of motion of potential moving objects. Tracklets are constructed to reduce the number of possible linkages that could be formed by providing information on plausible direction and sky-plane angular velocity (Kubica et al. 2007). They are then linked into internight linkages known as “tracks”: sky-plane paths of motion containing several tracklets spanning up to \(\sim 15\) nights, typically modeled with low-order polynomials.\textsuperscript{5} In the case of the LSST, for a moving object to be discoverable, it must be observed at least twice a night on at least three unique nights within the 15 day window to go through the tracklet-to-track creation process (Jones et al. 2018). In part due to the relative motion of the observer and the rate of motion of moving objects, both tracklets and tracks can exhibit high residuals relative to the fitted low-order polynomial, requiring relaxed fitting tolerances that can in turn lead to the creation of many spurious candidate linkages. Orbit determination (OD) algorithms are therefore required to run on each candidate track so spurious linkages can be identified and be removed.

The Zwicky Transient Facility (ZTF), an optical time-domain survey scanning the entire northern hemisphere of sky at a rate of more than 3700 deg\(^2\) hr\(^{-1}\), can be seen as a precursor to the LSST (Graham et al. 2019; Bellm et al. 2019; Masci et al. 2019). ZTF uses the ZTF Moving Object Discovery Engine (ZMODE; Masci et al. 2019) algorithm. Instead of linking tracklets directly into tracks, ZMODE first attempts to build a “stringlet.” A stringlet forms an intermediate step between tracklets and tracks that allows for the linking of pairs of detections across nights before tracks are built. This approach was designed to accommodate

\textsuperscript{5} More modern variants such as HelioLinC (Holman et al. 2018) use transformations to the heliocentric coordinate system to linearize the linking problem.
ZTF’s cadence during its main survey, where the cadence is frequently too sparse to form short intranight tracklets.

Recent work by Holman et al. (2018) has shown promising results by shifting the reference frame for linking detections to the heliocenter. By assuming a heliocentric distance and its rate of change, cleverly fitting tracklets for the remaining unknown parameters in inertial space, then propagating the resulting “arrows” to a common epoch, arrows corresponding to the same minor planets will form clusters. These clusters can then be extracted and subsequently validated by orbit fitting. As a testament to the effectiveness of HelioLinC, some 200,000 new minor planet orbits were recovered from the Minor Planet Center’s Isolated Tracklet File (Holman et al. 2018).

Common in all of these approaches is the requirement to build tracklets, which in turn requires a telescope to perform multiple revisits to the same field in a night, then more revisits a few nights later, and so on. For a survey that cannot cover the entire visible sky twice per night, this leads to up to a factor-of-two reduction of the nightly surveyed area. For a survey such as the LSST, which aims to balance four different science drivers, requiring such a cadence decreases the overall ease with which the other science drivers can be accommodated. It is therefore prudent to investigate whether linking algorithms that are cadence-independent can be constructed and whether such algorithms can perform as well as—or better than—the current methods. An algorithm that does not demand a high revisit cadence could increase the efficiency of future surveys, as well as help multi-science missions such as the LSST.

In this paper, we present one such cadence- and observer-independent linking algorithm: “Tracklet-less Heliocentric Orbit Recovery” (THOR). Rather than shifting the origin to the heliocenter like in Holman et al. (2018), we choose to shift the linking frame of reference to a series of dynamically selected heliocentric “test orbits.” The main insight is that transforming detections into the frame of a test orbit linearizes the motion of all objects in a relatively thick bundle of orbits near the test orbit (in phase space), which can then be picked out with line-detection algorithms such as the Hough transform. This provides a path to scanning an otherwise voluminous 6D phase space with a finite number of test orbits and at feasible computational cost.

We describe the THOR algorithm in Section 2. We test its performance on simulated and real survey observations in Sections 3 and 4, respectively. Finally, we discuss planned extensions and future work in Section 5.

2. Algorithm

Algorithms that rely on tracklets to discover moving objects generally search for the subset of orbits that are consistent with the observed tracklets and their constituent observations. THOR takes a different approach—it aims to identify subsets of observations that are consistent with being close to one or more “test orbits” drawn from the 6D orbital phase space. For each such test orbit, THOR searches for detections that lie near or adjacent to it as it is propagated backward or forward in time using clustering and line-finding algorithms (see Section 2.1.4).

This approach is computationally feasible because a single test orbit can be used to recover all objects that exist in a finite (and relatively large) “bundle” around it in phase space. This makes it possible to select and explore a sizable volume of phase space with a finite number of test orbits. It also makes it possible to focus on regions of phase space where we expect to find objects or that we know are populated, which we take advantage of to computationally efficiently search for specific asteroid populations in ZTF (Section 4).

There are several advantages to this approach to minor planets searches. First, the propagation of the test orbit is only limited by the robustness of the used orbit propagator. For example, propagating the test orbit using an $N$-body integrator allows the full gravitational perturbation by the Sun, planets, and massive minor planets to be properly taken into account. This is particularly useful for NEOs or any minor planets that might have nontrivial orbital encounters. Second, a fully defined orbit can be propagated to epochs beyond the scope of any one survey, permitting the use of THOR on archival and combined data sets. Third, by searching for observations near an assumed test orbit, the cadence of observations becomes irrelevant. The orbit can simply be propagated to any epoch where an observation is possible regardless of the temporal spacing between observations themselves. This removes the requirement for tracklets to be observed, and enables the linking of multiple single-night detections instead of the linking of multiple pairs of nightly detections.

The THOR approach does come with challenges. The computational cost, compared to HelioLinC-type algorithms, is higher due to the need to perform reference frame transformations to a (potentially large) number of test orbits rather than a single transformation to the heliocenter. However, as we show in Section 4, this cost can be readily met in practice. A more challenging issue is that of the selection of test orbits to optimally cover the phase space of interest (the phase space where small bodies are likely to be found). While we do show, later in this section, that a relatively ad hoc approach to orbit selection (essentially, importance sampling from the known population) is a very effective way to scan for all objects in a known population, it is also clear this approach will leave the algorithm blind to unknown populations or objects on unusual orbits. We leave the problem of developing an algorithm to select the optimal number of orbits to exhaustively search the entire phase space for future work.

2.1. THOR

We next describe the five main components of the THOR algorithm that enable the discovery of moving objects using a test orbit. We explain how test orbits are selected from 6D orbital phase space at the end of this section.

2.1.1. Select a Test Orbit

A test orbit with heliocentric position vector, $\mathbf{r}$, and heliocentric velocity vector, $\dot{\mathbf{r}}$, at some epoch, $t$, is placed in a survey that contains the detections of moving objects. In the top panel of Figure 1, we show an example test orbit placed in the simulated survey described in Section 3.

2.1.2. Propagate Test Orbit and Gather Detections

The test orbit is propagated to all possible epochs in the survey. For each unique epoch, all detections within some area,
Once all cells of detections have been gathered, they can be transformed into the heliocentric frame, rotated, and then projected onto a gnomonic tangent plane centered about the motion of the test orbit. We refer the reader to the Appendix for details on this procedure.

Figure 2 shows the transformed, rotated, and then projected detections in the corotating frame of the test orbit. Both $\theta_X$ and $\theta_Y$ are measured relative to the location of the test orbit at each unique epoch. Here, $\theta_X$ lies along the plane of the test orbit, whereas $\theta_Y$ is perpendicular to the plane of the test orbit. Once the detections have been transformed to the frame of the orbit and projected onto the tangent plane, each cell of transformed observations can be stacked. Any object that has an orbit similar to the test orbit will appear as a cluster: a circular group in the $\theta_X$–$\theta_Y$ plane (or equivalently, a line in $\theta_X$–$\theta_Y$–$t$ space). In the limit where the test orbit is identical to the orbit of an object in the frame centered on the test orbit’s motion, that object will appear as a cluster with a radius proportional to the mean astrometric uncertainty of its constituent detections. Any objects that are on orbits near the test orbit will appear as lines or curves in the corotating frame (there are a few such examples visible in Figure 2).

### 2.1.4. Hough Transform

The generalized Hough transform is a template-matching algorithm that can extract arbitrary features from multidimensional space (Ballard 1981). Lines and curves can be extracted from the test orbit’s corotating frame of reference using the equivalent of a 3D generalized Hough transform in $\theta_X$, $\theta_Y$, and time, $t$, space. To do so, we make a 2D velocity grid in $d\theta_X/dt$ and $d\theta_Y/dt$. By shifting the detections using a grid of assumed linear velocities relative to the test orbit’s motion, we effectively perform a 3D Hough transform as a 2D clustering problem.

In Figure 3, the subset of data outlined by the black square in Figure 2 is plotted. The left panel shows the corotating frame of the test orbit, with the relative motion of nine other objects drawn as lines. The remaining detections with their underlying true linkages are shown in blue. Note the cluster shown in red at $(0, 0)$. Surrounding the cluster is a circle indicating the chosen clustering radius. In the right panel, the same detections after they have been shifted assuming a linear velocity in both $\theta_X$ and $\theta_Y$ are plotted. Note how one of the nine other objects now forms a cluster; that cluster is also surrounded by a circle indicating the clustering radius (this cluster was found using the actual clustering code). Because the selected test orbit was the orbit of a real object, in the nonzero velocity frame, its detections now also form a line, as shown by the red line toward the middle of the right panel.

With a fine velocity grid, it is likely that similar test velocities will result in the building of clusters with identical subsets of detections. This is also a function of the maximum allowable clustering radius. After all velocities have been tested, the clusters are scanned and any duplicates that have the same constituent detections as another cluster are removed.

We consider any cluster with detections belonging to a single object a “pure” cluster. Some clusters may be contaminated with observations from artifacts or some other, unassociated object(s). We call these “partial” clusters. For example, a cluster with a total
of ten detections, where eight of those detections belong to one object and two detections are either false positives or belong to another object or objects, would have a contamination percentage of 20%. Partial clusters could still permit for the discovery of objects, as outlier rejection can be used to filter out the impostor detections. Any cluster that is neither pure nor partial is considered a “mixed” cluster. In the next section, we describe how the process of orbit determination is used to test the validity of the clusters, and how outlier rejection is used to clean partial clusters.

### 2.1.5. Orbit Determination

Orbit determination in THOR is a three-step process with the aim of testing the validity of the clusters produced by the Hough transform described in Section 2.1.4.

Initial orbit determination (IOD) serves as a first-level filter against erroneous linkages, intended to efficiently reduce the number of clusters to be considered for full differential orbit determination (OD). We implemented the method of Gauss following the formalism introduced by Milani et al. (2008). It seeks to fit a Keplerian orbit to three observations on the sky by solving for the orbit at the time of the second observation. This involves solving an eighth-order polynomial, which can result in up to three solutions per linkage. A challenge for this approach is choosing which three observations to use: as our implementation is not computationally expensive, we elected to fit every combination of three observations for each cluster (i.e., a cluster of five constituent observations would mean testing ten combinations) until a combination is selected such that the $\chi^2$ with respect to the observations meets a user-defined threshold. If all combinations are exhausted before the threshold is met, and no outliers can be removed, then the candidate orbit is discarded.

To clean and filter “partial” linkages or clusters, we added outlier rejection to the IOD component of THOR’s orbit determination pipeline. The user sets a contamination threshold that defines the maximum percent of observations in a linkage that can be flagged as outliers. If all combinations of three observations have been tested via the method of Gauss, the linkage has not yet met the acceptance threshold, and the linkage has sufficient observations to allow for outlier testing, then the observation with the highest $\chi^2$ residual is removed. If the orbit’s overall $\chi^2$ meets the user-defined threshold, the orbit is accepted. If the orbit does not meet the threshold, the

---

**Figure 2.** Plotted in blue are the cells of detections from Figure 1 after they have been transformed to the heliocentric frame and then rotated and projected into the frame of the test orbit’s motion. Once transformed and projected, the cells can be stacked. Note that the corotating reference frame is fundamentally three-dimensional. The time dimension ($z$-axis) has been collapsed onto the gnomonnic plane ($\theta_X-\theta_Y$ plane). In its own frame of reference, the test orbit (black plus sign) lies at (0, 0). Objects with orbits similar to the test orbit will appear as lines or curves, some of which are visible in this image. We plot the detections of 50 such objects in red, with their underlying true linkages plotted as red lines. There are a total of 1397 objects with at least five detections in this image; these are considered findable. The black square outlines the subset of the data used to describe how clustering is performed in Figure 3. This figure was generated using plots_simulations.ipynb.

---

5 We note that one must be careful when interpreting the distribution of $\chi^2$ values when only three observations are used to fit the preliminary orbit, but residuals are calculated against all $N$ observations in the cluster. The resulting distribution will not follow the $\chi^2$ distribution with $2N - 6$ degrees of freedom, but generally be wider.
next-highest residual observation is also removed, and so on, until the maximum number of observations have been removed as outliers or the threshold has been met.

The dynamic range of \( \chi^2 \) values obtained for clusters may span many orders of magnitude. This is also true even for pure clusters, where the approximate nature of the method of Gauss might yield residuals as high as several arcseconds. To further improve these solutions, a refit could be attempted using a more robust technique such as the method of Herget (1965). However, we found it sufficiently computationally efficient to simply pass all orbits satisfying a very relaxed IOD cut to the differential corrector. The corrector then acts as a much stronger and more accurate filter.

After IOD, the preliminary orbits are differentially corrected (OD) using the method of weighted linear least squares in combination with numerical differencing to determine the Jacobian (the matrix of partials of the observations with respect to changes in the orbit state). We implemented the differential correction algorithm described in Vallado (2013), with a few notable differences: we added outlier rejection in a similar fashion to the IOD code, a \( \chi^2 \) acceptance threshold, and adaptive finite differences.

The goal of differential correction is to improve a preliminary orbit solution by minimizing the residuals across all its constituent observations. The improved orbit is propagated to all observation times, and residuals with respect to the observations are calculated. If such a differentially corrected orbit meets a user-defined \( \chi^2 \) threshold, it is accepted. If the differentially corrected orbit does not meet the user-defined threshold, iteration continues until the maximum number of iterations is reached. If, at this point, the orbit does not meet the user-defined \( \chi^2 \) threshold, the observation with the highest \( \chi^2 \) value is removed and iterative improvement is continued using the new set of observations. The iterative outlier removal continues until the maximum allowable number of observations have been removed as determined by the user-settable contamination threshold. This process is an effective filter against erroneous (“mixed”) orbits, and is an effective method to clean contaminated (“partial”) orbits.

Orbits that pass the differential correction step will likely contain only a subset of the observations that belong to those orbits. Some orbits also will share observations with other orbits (e.g., orbits that represent different sub-arcsec of one longer arc). The last step in THOR’s orbit determination process aims to merge orbits that represent the same object, and extend orbits that could have more observations associated with it. This is done through an iterative combination of OD and observation attribution. We perform attribution by propagating the differentially corrected orbits to all epochs of the observations contained in the gnomonic tangent plane of the test orbit. Any observations that lie within a user-defined angular distance of the predicted location are identified as possible attributions. If, at this point, any observations are identified as belonging to multiple orbits, these orbits are merged to form one larger orbit (a “child” orbit) and refit. If such a refitted child orbit passes the OD filter (i.e., meets the \( \chi^2 \) threshold), its parent orbits are removed from the active pool of possible orbits. Any orbits that pass OD but could not be improved further are unlikely to have more observations attributed to them and are output as high-quality candidate orbits. This iterative combination of attribution and refitting is done until each
orbit has only unique constituent observations and the orbits are not further improved by OD.

In summary, the orbit determination stage filters and transforms the candidate clusters produced by the Hough transform step into a catalog of high-quality, differentially corrected orbits. These successfully linked orbits and their observations are removed from the active pool of observations, and steps 2.1.1–2.1.5 are repeated with new test orbits to attempt to link the remaining observations.

2.2. Phase-space Gridding

A key element of THOR is the selection of test orbits and how to best populate the relevant phase space to maximize completeness for different populations of minor planets. The bulk of the Solar System small bodies that LSST will discover are Main Belt asteroids (Jones et al. 2018). We initially focus on populating the phase space to discover populations in the Main Belt and the slower-moving outer Solar System populations. Future work is planned to target the faster-moving NEO population, and to further optimize test orbit selection.

Guided by tests on real ZTF data, the current implementation of THOR uses nine average orbits from the known population of minor planets per patch of sky. The patches of sky are currently a simple rectangular sky-plane subdivision 15 by 15 deg in (α, δ) with no overlap. This notional subdivision could easily be replaced by something more robust such as HEALPix (Gorski et al. 2005). The cell size was set to 1000 deg² to maximize the number of detections gathered by each test orbit at each epoch. This size permits a test orbit to be selected along any edge of a patch and still gather all detections in that patch of sky during the first epoch of observation. The patches in which an average orbit is calculated do not have any effect on the detections that propagating a test orbit can gather; they are simply used to find a set of test orbits.

The test orbits are selected in bins of semimajor axis (a), with bin edges roughly corresponding to the location of the most prominent mean-motion resonances (the Kirkwood gaps). To target the Main Belt, we set the innermost edge at a = 1.7 au with the outermost edge at a = 5.0 au. The remaining bin edges were set to a = 2.06, 2.5, 2.82, 2.95, and 3.27 au, which correspond roughly to the Saturn v₆ secular resonance, and the 3:1, 5:2, 7:3, and 2:1 Jovian mean-motion resonances, respectively (Minton & Malhotra 2009). To target the outer Solar System, we added a single bin with edges a = 5.0 au and a = 50.0 au.

For each bin, an object with average eccentricity (e) and inclination (i) is selected from the Minor Planet Center’s (MPC) catalog of known objects. No test orbits with eccentricity greater than 0.5 were selected. This relatively simple orbit selection is not sufficient to target the complete Main Belt population. To target the missed populations from earlier trials, we subdivided the first semimajor axis bin further—in particular, the bin located at the approximate location of the Hungaria family of asteroids (e < 0.18, 1.78 < a < 2.00 au, 16 < i < 34 deg). The Hungarias are notable for their wider spread of orbits in orbital phase space, especially in inclination (Warner et al. 2009). Three test orbits were selected in different bins of eccentricity (e) with bin edges (including the exterior edges) e = 0.0, 0.1, 0.2, and 0.4.

In Figure 4, we plot the median linear velocities in θₓ and θᵧ of all the findable objects relative to one of the test orbits selected for the simulated survey. The simulated survey is described in the following section. The plotted error bars are the standard deviations of the velocities, and are a proxy for the curvature in the 3D space of the gnomonic plane and time. The test orbits that were selected for use on the simulated survey are plotted in red. We used many different iterations of Figure 4 to inform decisions regarding the test orbit selection code. The test orbit selection criteria and code were designed in such a way as to maximize coverage in this space.

Note that, by choosing different upper and lower velocity bounds, different heliocentric distances are probed by each test orbit. For example, the velocity grid can be set to sweep out a wider heliocentric distance at the cost of greater computation time. Equivalently, an additional test orbit could be used with a smaller grid to the same effect and at a possibly lower computational cost. These are trade-offs that might depend on the survey and its data properties—in particular, a survey’s observational depth influences the number of observations, and its hardware and software characteristics influence the occurrence of false positives. In general, the propagation and gathering of detections takes on the order of seconds on a single thread, whereas the current Hough transform implementation takes on the order of minutes to hours with moderate parameters on a few dozen threads. The Hough transform component of THOR is where future optimization will be focused, in addition to improving test orbit selection.
Figure 5. Plotted in blue are the simulated detections of real orbits generated using the MPC’s orbit catalogue for the first night of the simulation. The ephemerides for the underlying orbit catalogue from which the blue detections are drawn are shown in gray. The red points indicate the locations of the selected test orbits on the first night. The red circle traces the 1000 deg$^2$ cell size for one of the orbits. The 16 square fields (black squares) each ten deg$^2$ in size are visited once every other night over a period of 14 nights. The footprint was roughly centered on the ecliptic to maximize minor planet density. No false-positive detections are plotted, even though they were included in the simulated survey. This figure was generated using plots_simulations.ipynb.

3. Validation with Simulated Data

3.1. Simulation Setup

To test the efficacy of our algorithm, we created a simulated mini-survey composed of 16 square fields 10 deg$^2$ in size. Each field is visited once every other night over a 14 day period, yielding seven unique visits per field over the course of the simulated survey. We downloaded the Minor Planet Center’s catalog$^6$ of known orbits and used open-source orbit propagation and determination software, oorb, to generate ephemerides for each known object at the different survey times (Granvik et al. 2009). In practice, THOR will run on the detections resulting from difference imaging; however, simulated ephemerides are sufficient to test the performance of linking detections. The survey footprint was roughly centered on the ecliptic to maximize the density of minor planet detections (see Figure 5). We added 100 ± 10 randomly distributed false-positive detections per squared degree to simulate the effect of image processing artifacts. The final survey consisted of 139,120 (55.4%) simulated observations belonging to real orbits and 112,078 (44.6%) false-positive observations. The simulated observations were randomized with 100 mas scatter to reflect the expected astrometric errors for moving objects as observed by current-generation surveys.

We consider any minor planet with at least five detections throughout the survey to be findable, yielding a total of 18,332 such objects. Any object that is recovered in a pure orbit is considered to be found. For comparison, the Vera C. Rubin Observatory’s current implementation of MOPS would require three tracklets (six detections in three intranight pairs) for an object to be findable—as a tracklet-based algorithm, it would recover none of the objects in the simulated survey.

The contamination threshold for partial clusters was set to 20%, with a minimum cluster size of five detections (see Section 2.1.4). The default grid in both $d\theta_x/\,dt$ and $d\theta_y/\,dt$ for the Hough transform was set to 300 equally binned velocities between −0.1 and 0.1 deg day$^{-1}$ relative to the motion of the test orbit, yielding a total of 90,000 velocities tested per test orbit. The minimum cluster size was set to five detections (equal to the findability criterion), with a cluster radius of 0.005. To extract clusters for every velocity tested, we use scikit-learn’s implementation of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm (Ester et al. 1996; Pedregosa et al. 2011). DBSCAN is a nonparametric clustering algorithm that finds clusters in higher-dimensional space of approximately equivalent density given a maximum distance parameter and marks unclustered data as outliers. We use open-source orbit determination software, oorb, to propagate the test orbits to all possible times in the survey (Granvik et al. 2009).

For orbit determination, the $\chi^2$ and contamination thresholds were set to (10$^2$, 20%), (10, 20%), and (10, 0%) for IOD, OD, and the iterative combination of differential correction and attribution, respectively. The maximum attribution distance was set to one arcsecond. The minimum number of observations and minimum arc length for an orbit to be accepted were set to five observations and one day, respectively.

3.2. THOR Performance

Using nine test orbits, selected as described in Section 2.2, THOR recovered 16,729 candidate orbits representing 16,728 (91.3%) of the findable objects in the simulated survey. Only one erroneous orbit was produced.

The top panel of Figure 6 shows the recovered orbit completeness in bins of semimajor axis (a) and inclination (i). The red contours indicate the number of objects that should be findable, while the red points show the selected test orbits. The vertical dashed lines trace the semimajor axis bin edges. The percentages state the overall completeness in each bin of semimajor axis (note that the first bin has three test orbits). In the bottom panel, we repeat the same plot configuration as in the top panel, but we plot eccentricity (e) instead of inclination. Instead of percentage completeness in the five semimajor axis bins, we state the numbers of unique objects found to give the percentages context. As shown in both panels, the orbits selected in the bins beyond 2.5 au recover the vast majority (96.1%) of the findable objects within that region. Performance in the semimajor axis bin between 2.06 au and 2.5 au (85.2%) suggests that either more test orbits are needed or that the test orbit we selected is not adequately representative of the underlying orbital distribution. The same is true for the three orbits selected in the semimajor axis bin between 1.7 au and 2.06 au (61.5%). Unsurprisingly, the two Hungaria orbits did not completely recover their constituent population, because of the larger spread of Hungaria orbits in orbital element phase space. The high-eccentricity orbit selected did, however, enable other high-eccentricity objects to be recovered in addition to many objects in bins with larger semimajor axis. The test orbits were run in series, starting in order of increasing semimajor axis. The second orbit that ran was the one selected in the inner Main Belt, and over 10,000 of the findable objects in the survey were successfully found by using this test orbit. Overall completeness for the Main Belt and beyond (a > 1.7 au) is 91.6%. While we did not focus test orbit selection to tackle the

---

$^6$ https://minorplanetcenter.net/iau/MPCORB.html
NEO population and populations interior to the Main Belt ($a < 1.7$ au), serendipitous completeness for these objects is 18.8%.

3.3. Computational Performance

The complete simulations run, starting from a catalog of simulated observations to producing a catalog of 16,729 orbits using nine test orbits took seven hr on a personal workstation using 60 threads on 30 cores. The workstation has the following specifications: a 32-core/64-thread 2990WX AMD Threadripper, 128 GB DDR4 RAM, and 2 TB of SSD drive space. The run required a moderate amount of memory, especially during the attribution of orbits, approximately 1.5 GB per thread. Given the heavy reliance of the current implementation on Pandas data frames, it is likely the memory usage could be further optimized. Theoretically, the algorithm should require little more than 2–3x the space occupied by the observations.

3.4. Completeness and Purity Considerations

In Figure 7, we plot completeness (in blue) and purity (in green) as functions of the different pipeline components. Plotted in red is the number of linkages at each stage. From clustering to the final stage, we see only a 0.2% drop in completeness, while the purity increases from 48.5% to 99.9%. Moreover, the number of clusters, initial orbits, and recovered orbits, quickly approaches the true number of objects recovered at each subsequent stage of the pipeline. The nearly constant completeness across the different pipeline components indicates that further completeness is likely to be found by improving our test orbit selection algorithm.

4. Validation with Zwicky Transient Facility Data

The Zwicky Transient Facility (ZTF) is a robotic time-domain survey of the northern sky capable of scanning more than 3700 deg$^2$ hr$^{-1}$ with its 47 deg$^2$ field of view, 600 megapixel camera, and readout time of just 8 s. ZTF can observe to a median 5$\sigma$ $r$-band depth of 20.6 mag (Bellm et al. 2019; Graham et al. 2019; Masci et al. 2019). During a full night of observing, ZTF may issue as many as 1.2 million alerts ($\sim$1/10 of what is expected for LSST) to science users through its alert broker system (Patterson et al. 2019). Contained within the alerts are the detections of moving objects. In particular, if a previously discovered Solar System object is predicted to be at
the location of a difference image point source, the source can be attributed to the known object (Masci et al. 2019). In its first three months of operations, ZTF submitted $\sim$600,000 observations of minor planets and has discovered about 320 new objects (Bellm et al. 2019). As only its public survey cadence is designed to emulate the baseline cadence of LSST and produce tracklets, the full ZTF data set provides a good framework to test THOR on data that otherwise would not be readily useful for minor planet discovery.

4.1. ZTF Alert Data Set

We downloaded 15 nights’ worth of ZTF alerts dating from 2018 September 3 to 2018 September 17. We filtered out static sources and selected all alerts with real bogus values above 0.5, emulating future LSST requirements. This yielded a final count of 255,358 (30.9%) known object observations that could be matched to objects in the Minor Planet Center’s orbit catalog. The remaining 572,188 (69.1%) alerts are unassociated and therefore unknown. It is likely that among the unassociated detections there are undiscovered moving objects in addition to astrophysical transients and false positives (e.g., image subtraction artifacts).

During the 15 night window of alerts, ZTF scanned over 20,000 deg$^2$ of sky (for comparison, the simulated survey had a footprint of 160 deg$^2$). We divided the sky plane into 15 by 15 deg patches of sky, and for each patch, selected nine average known objects in different bins of $a$ as described in Section 2.2. If no known object exists in a bin of $a$ for a particular patch, no test orbit was used for that bin. A total of 821 test orbits were chosen. In Figure 8, we plot the ZTF observations in blue with the underlying catalog of known orbits in gray. The red dots represent the on-sky location of the 821 test orbits chosen and the five red circles indicate the effective cell area on the sky. The black lines outline the 15 by 15 deg patches used to select up to nine test orbits per patch.

The Hough transforms and orbit determination for each test orbit were conducted with the same parameters and configuration as in the simulated survey (300 equally binned velocities between $-0.1$ to 0.1 deg day$^{-1}$ and a maximum allowable clustering radius of 0.005 deg). For clustering, the contamination threshold was set to 20% and the minimum cluster size was set to five observations. For orbit determination, the $\chi^2$ and contamination thresholds were set to (10, 20%), (10, 20%), and (10, 0%) for IOD, OD, and the iterative combination of differential correction and attribution, respectively. The maximum attribution distance was set to one arcsecond. The minimum number of observations and minimum arc length for an orbit to be accepted were set to five observations and one day, respectively.

Compared to the simulated survey, nearly 100 times as many test orbits were chosen. We grouped the test orbits into chunks of five patches (yielding a maximum of 45 test orbits per chunk). Each chunk was submitted to an individual node on the University of Washington’s HPC cluster, Hyak. After each chunk finished processing, the recovered orbits were concatenated and a simple algorithm was run to select the orbit with most observations and longest arc for each set of orbits that shared observations. After shared observations were assigned to their best-fitting orbit, a final step of differential correction was executed to update any orbits that had any observations removed.

Like the simulated survey, we consider any known object with at least five detections throughout the 15 nights of alerts to be findable, resulting in a total of 21,542 such objects. Assuming a classical MOPS approach, we find that only 9,381
(44.8%) of the known objects would be findable. With the ZMODE algorithm, we find 14,291 (68.2%) would be findable\textsuperscript{7}. ZMODE is designed to work with ZTF’s cadence and attempts to link objects with a minimum of four detections over four consecutive nights, while the classical LSST MOPS algorithm looks to link three pairs of detections over 15 consecutive nights and thus requires a minimum of six observations. We consider any object recovered in a pure orbit with at least five observations to be found.

4.2. THOR Performance

THOR recovered 21,723 candidate orbits consisting of nearly 167,000 unique observations from the described two-week slice of the ZTF data set. These include 21,018 pure orbits for 20,940 (97.2%) of the 21,542 minor planets with at least five detections. The remaining 705 orbits contain either only unassociated/unknown observations or a mix of associated and unassociated observations. Of the 21,723 recovered orbits, 1,783 orbits contain only singleton observations (orbits made completely without tracklets). In the top panel of Figure 9, we plot completeness in bins of semimajor axis (a) and inclination (i). The red number density contours show the number of objects that should be findable, while the red points indicate the chosen test orbits. The vertical dashed lines trace the bin edges. The outer Solar System bin stretches from 5.0 to 50 au, but is plotted in its entirety. The percentages at the top of each bin list the overall completeness in each bin. In the bottom panel, we repeat the same plot configuration, but instead of inclination, we plot eccentricity (e). We replace the percentage completeness values in each bin with the number of found objects, to give the percentages greater context. As in the simulated survey, the performance on ZTF alerts beyond 2.5 au suggests most of the findable objects were recovered (98.4%). The performance in the bins nearer than 2.5 au is better than in the simulated survey. This is likely due to the designed overlap of selecting nine orbits per 15 $\times$ 15 deg$^2$ patch of sky and setting the cell size to 1000 deg$^2$ (see Figure 8). Note that, in the bottom panel, it is clear that high-eccentricity objects were not favored in recovery. Test orbit selection was limited to orbits with eccentricity less than 0.5, and some of these objects may appear nonlinearly within the corotating frames of the test orbits. Overall completeness for the Main Belt and beyond (a > 1.7 au) is 97.4%, including serendipitously linking 28.9% of objects with orbits interior to the Main Belt (a < 1.7 au).

Our test orbit selection yielded a total of 821 orbits; however, only 621 of those orbits lead to known objects being found. Two hundred and twenty of the test orbits yielded no known object discoveries, of which 88 had no findable objects in their corotating frame of reference. This is due to several reasons: first, some of the selected test orbits might belong to objects that are too dim to be detectable by ZTF. Second, because the orbits are run in series, moving outward in semimajor axis space, it is likely that some orbits that ran earlier swept out their population of findable objects—resulting in the later test orbits not having any objects to find, especially if the two orbits were similar.

In all, the total number of known objects recovered is 20,940 (97.2%) of the 21,542 with five or more observations. This represents a factor of $\sim$2 discovery increase over traditional MOPS, and a factor of $\sim$1.5 increase over ZMODE.

4.3. Computational Performance

We grouped each set of test orbits into chunks of five patches. Each group of test orbits was then set to run on a node on UW’s HPC cluster, Hyak. A total of 23 nodes, with 28 cores

\textsuperscript{7} The numbers stated here assume ideal performance by both pipelines—i.e., 100% completeness.
Completeness and purity as functions of pipeline components are plotted in blue and green, respectively. The number of linkages is plotted in red. The “combined” step is the simple algorithm used to combine orbits recovered by individual nodes on the HPC cluster. Each subsequent component of THOR enhances linkage purity while completeness remains largely unchanged. The number of linkages quickly approaches the true number of recovered objects. This figure was generated using `plots_ztf.py`.

4.4. Completeness and Purity Considerations

We next characterize the completeness, purity, and any additional filtering required for submission of candidate new discoveries to the Minor Planet Center. In Section 4.4.1, we evaluate which objects we would have discovered and reported in 2018, given the state of MPC’s orbit catalog at the time. In Section 4.4.2, we use a more recent catalog of known orbits from April 2021 to assess the purity of the sample we would have submitted in 2018, and look for still-unknown objects.

4.4.1. 2018 Orbit Catalog Analysis

The ZTF alerts that correspond to possible observations of known minor planets are labeled as such part of the ZTF alert pipeline. These labels or associations are the “ground truth” with which we compare THOR’s performance. As stated in the previous section, THOR recovered 21,723 candidate orbits including 21,018 pure orbits for 20,940 (97.2%) of the 21,542 minor planets with at least five detections.

In Figure 10, we plot completeness and purity as functions of the different pipeline components. The purity rises from only 3.8% at the Hough transform stage to 96.8% for the final recovered orbits, while—quite strikingly—completeness drops by only 0.8%. This suggests that further completeness is likely to be found by improving our test orbit selection algorithm. In absolute terms, the number of linkages drops from $\sim 7 \times 10^5$ clusters to 21,723 recovered orbits. Among the 21,018 pure orbits are 150 orbits that contain only sub-arcs of the longer observations arcs of 72 unique objects in the data set. The continued presence of these duplicates suggests further improvements to the orbit merging and extension algorithm are necessary.

We apply two additional filters to produce a final set of high-confidence candidates that could be submitted to the MPC. First, we remove any observations that occurred within 30 minutes of another observation. This helps in removing observations of stationary sources that may not have been filtered out during preprocessing. Doing so reduces the number of unknown orbits from 705 to 526. Second, we remove any orbits that contain the observations of known objects. In practice, if THOR were deployed as a discovery algorithm, it would not run on observations that have already been associated with known objects. This reduces the number of unknown orbits to 488 high-quality discovery candidates that, had THOR been running in 2018, would have been submitted to the MPC.

4.4.2. 2021 Orbit Catalog Analysis

At the time of writing, two-and-a-half years of minor planet discovery have passed since ZTF made the observations in 2018. We downloaded the Minor Planet Center’s (MPC) catalog of known orbits on 2021 April 20, and ran attribution of the MPC’s orbits on the two weeks of ZTF observations, creating a new set of updated associations. We then analyzed how THOR performed, with the benefit of hindsight, to establish the lower limits on the purity of the sample that would have been reported to the MPC in 2018.

Of the 488 orbits that were identified as high-quality, submission-ready discoveries using the 2018 associations, we find that 477 correspond to observations of 476 unique minor planets that have been discovered or had their orbits improved since 2018 (one object was identified twice, as two shorter sub-arcs). The remaining 11 orbits in the high-quality sample are as-yet undiscovered objects or false linkages; based on visual inspection and comparisons of their magnitudes, they do appear to be real objects. The observations and predicted motion of the 11 discovery candidates are plotted in Figure 11.

Using the IMCCE SkyBot service (Berthier et al. 2006), we conducted a cone search for all observations belonging to the 11 orbits. The observations of 10 candidates could not be associated to any known objects within 10’. Only the candidate with $e > 1$ orbit had all observations within 5” of a known object—the comet C/2018 U1. C/2018 U1 is classified as a hyperbolic comet and was discovered on 2018 October 27. We compared the orbit fit of C/2018 U1 to the orbit as reported in JPL’s SBDB. The inclinations agree to within 0.3 deg, and the perihelion distances agree to within 0.2 au, while our fit overestimates eccentricity by 0.1. Given the short arc (6.13 days) and small number of observations (five observations) with which the orbit was fit, we conclude our solution agrees remarkably well with the one present in the SBDB. The ZTF observations were made during a two-week slice that took place six to eight weeks prior to the discovery date. Had THOR

\footnote{Note that the purity as computed here is actually a lower limit. The 705 orbits that have been identified as mixed or unknown will contain objects that were undiscovered in 2018, objects that had known orbits not of sufficient quality such that observations could be attributed to them, as well as truly erroneous linkages.}

\footnote{https://ssd.jpl.nasa.gov/sbdb.cgi}
Figure 11. The observations of the 11 discovery candidates identified in Section 4.4.2 are plotted in red, with the sky-plane motion of their best-fit orbits plotted as lines. Ten of the objects show MBA-like best-fit orbit solutions. The remaining object has a hyperbolic orbit solution and corresponds to recovery observations of the hyperbolic comet C/2018 U1. The “wiggles” apparent in some of the best-fit orbit lines are due to the motion of the observer (topocentric motion). This figure was generated using plots_ztf.ipynb.
been running at the time when data was taken, this comet 
would have been discovered by ZTF.

Therefore, we conclude that the purity of submissions—an 
important consideration for the Minor Planet Center—is at least 
97.7%. If all 11 discovery candidates are confirmed as real, 
then the purity of the submitted sample could be as high as 
100%. This demonstrates that, at a practical level, THOR is 
ready to be used for discovery and reporting of new discoveries 
with a high degree of confidence.

5. Summary and Future Work

We present an observer- and cadence-independent asteroid 
linking algorithm, which we name “Tractless Heliocentric Orbit Recovery” (THOR), and show its performance on both 
simulated and real detections. We have shown that THOR can 
link Main Belt asteroids and more distant populations at high 
completeness and at moderate computational cost. On two 
weeks of simulated data, THOR recovered 91.3% of the 18,332 
that were ideally findable, whereas a tracklet-based linking 
algorithm would have recovered none. On two weeks of ZTF 
data, THOR linked 97.2% of the 21,542 objects with at least 
five detections (a factor of ~2 recovery increase over MOPS 
and a factor of ~1.5 increase over ZMODE). THOR recovered 
orbits for 97.4% of objects beyond 1.7 au, with 98.4% of 
objects recovered beyond 2.5 au. Furthermore, by comparing 
the 2018 sample to the catalog of orbits as presently known 
(April 2021), we show that the lower limit on purity of THOR 
submissions to the MPC would be 97.7% and—assuming all 
candidates shown in Figure 11 are confirmed as real—possibly 
as high as 100%. This, in combination with its capability to 
discover objects regardless of cadence or observer, renders it 
meaningfully useful for Main Belt and KBO searches on survey 
data and archival data sets.

While THOR can be applied to running surveys, application to 
archival data sets—such as ZTF, CSS, or PanSTARRS—is 
interesting as well. The most straightforward way to do this 
would be to run a sliding ꞌtwo-week window over the data set, 
from the beginning to the end, running THOR at each window 
instance. For each window run, orbits could be computed for 
the discovered objects and projected to the past (and future) 
for discovery of additional observations (and improvement of 
orbital solutions). This is the approach we ourselves plan to 
take with the ZTF archival data. Assuming the 11 objects 
discovered here are representative of remaining undiscovered 
objects in ZTF, this search would likely yield on order of 1,000 
aroid potentials with deeper archival data sets (e.g., DES data or the 
DECam archive) is likely to be significantly larger.

5.1. Future Directions

There are several avenues for improvement, which we 
discuss here below.

Test orbit selection: As discussed previously, THOR can 
discover objects at very high completeness in the Main Belt 
and beyond using the current test orbit selection criteria. Those 
selection criteria are, however, not ideal for the inner Main 
Belt, NEOs, objects interior to Earth’s orbit, or objects on 
generally unusual orbits (e.g., ISOs). For the linking algorithm 
to be universally viable, these populations need to be reliably 
discoverable. The focus of this paper has been to demonstrate 
that the overall approach works even with a rather simplistic 
orbit selection algorithm; developing an optimal test orbit 
allocation algorithm is the focus of follow-up work and is 
currently in progress.

Hyperparameter optimization: Throughout the description of 
the algorithm’s performance on both the simulated survey and 
on two weeks of ZTF alerts, we stated the values of the 
hyperparameters used. These values were decided on after 
numerous trial runs, early testing, and optimizations. We plan 
to quantify in much greater detail how THOR performs as these 
parameters change, and how these parameter might change 
with different combinations of data sets and surveys. This 
includes automated data-driven selection of the clustering 
radius, the velocity grid, and its granularity, as well as 
decisions on the width of the time window over which to 
attempt linking.

Orbit determination: With our goal of extending THOR to 
the NEO population will likely come the need to make 
improvements to the orbit determination component of THOR. 
Improvements such as iteration at the IOD stage, as opposed to 
delegating the work directly to the differential corrector, could 
be useful here. We have also started researching the possibility 
of bootstrapping orbit determination directly from the corotat-

ing reference frame of the test orbit. The test orbit effectively 
serves as the initial orbit guess for the orbits of objects 
contained in the tangent plane centered on the motion of the 
test orbit. The relative velocities (determined by the Hough 
transform), along with the location in the tangent plane, should 
adquatly constrain four of six orbital parameters. The 
remaining two can be fit for at lower computation cost than 
performing initial orbit determination and differential correc-
tion on sky-plane coordinates.

Scalability and “discovery-as-a-service”: We plan to deploy 
THOR as a scalable cloud-based service to which observations 
can be submitted for automated identification of Solar System 
objects, with candidate discoveries returned to the user and/or 
submitted to the Minor Planet Center. This would effectively 
provide the Solar System community with a “discovery-as-a-
service” platform. The advantage of such a cloud-based 
platform is that it can elastically scale to meet THOR’s 
computational needs as data sets are submitted. Additionally, a 
centralized discovery service would enable cross-survey and 
cross-data set discovery.

5.2. Software and Transparency

The THOR code is maintained on GitHub. All the analysis 
presented in this document, the figures, and the data sets are 
available online.10 Of particular note is a series of Jupyter 
notebooks that reproduce all the numbers, results, and plots in 
this paper.11 Included with the GitHub repository are instruc-
tions on how to install all the software dependencies and code 
in a containerized environment using Docker. We encourage 
the reader to visit the GitHub repository for more details.

J. Moeyens wishes to acknowledge the support of the 
Asteroid Institute and B612 Foundation. This research was 
made possible by major gifts to support the ADAM project 
provided by the W.K. Bowes Jr. Foundation, Steve Jurvetson, 
The P. Rawls Family Fund, Tito’s Handmade Vodka, Yishan 
Wong and Kimberly Algeri-Wong, and two anonymous

10 https://github.com/moeyensj/thor
11 https://github.com/moeyensj/thor_notebooks
donors, along with Founding Circle and Asteroid Circle members: AGI (in-kind), B. Anders, R. Armstrong, G. Baehr, The Barringer Crater Company, B. Burton, D. Carlson, S. Cerf, V. Cerf, C. Chapman, Y. Chapman, J. Chervenak, D. Corrigan, E. Corrigan, A. Denton, E. Dyson, A. Eustace, A. Fritz, L. Fritz, S. Galitsky, E. Gillum, L. Girand, Glaser Progress Foundation, Google Cloud (in-kind), D. Glasgow, J. Grimm, S. Grimm, G. Gruener, V. K. Hsu & Sons Foundation Ltd., J. Huang, J. D. Jameson, J. Jameson, M. Jonsson Family Foundation, D. Kaiser, S. Krausz, V. Lasas, J. Leszczenski, D. Liddle, S. Mak, G. McAdoo, S. McGregor, J. Mercer, M. Mullenweg, D. Murphy, P. Norvig, S. Pishvaer, R. Quindlen, N. Ramsey, R. Rothrock, E. Sahakian, R. Schweickart, A. Slater, T. Trueman, F. B. Vaughn, R. C. Vaughn, M. Welty, S. Welty, B. Wheeler, M. Wyndowe, and seven anonymous donors in addition to donors from over 46 countries around the world. J. Moeyens further wishes to thank the LSSTC Data Science Fellowship Program; his time as a Fellow has benefited this work.

M. Jurić and J. Moeyens wish to acknowledge the support of the Washington Research Foundation Data Science Term Chair fund, as well as the University of Washington Provost’s Initiative in Data-Intensive Discovery.

All authors acknowledge support from the University of Washington College of Arts and Sciences, Department of Astronomy, and the DiRAC Institute. The DiRAC Institute is supported through generous gifts from the Charles and Lisa Simonyi Fund for Arts and Sciences, and the Washington Research Foundation.

Based on observations obtained with the Samuel Oschin 48 inch Telescope and the 60 inch Telescope at the Palomar Observatory as part of the Zwicky Transient Facility project. ZTF is supported by the National Science Foundation under Grant No. AST-1440341 and a collaboration including Caltech, IPAC, the Weizmann Institute for Science, the Oskar Klein Center at Stockholm University, the University of Maryland, the University of Washington, Deutsches Elektronen-Synchrotron and Humboldt University, Los Alamos National Laboratories, the TANGO Consortium of Taiwan, the University of Wisconsin at Milwaukee, and Lawrence Berkeley National Laboratories. Operations are conducted by COO, IPAC, and UW.

This work was supported by the Washington Research Foundation and by a Data Science Environments project award from the Gordon and Betty Moore Foundation (Award #2013-10-29) and the Alfred P. Sloan Foundation (Award #3835) to the University of Washington eScience Institute.

This research has made use of data and/or services provided by the International Astronomical Union’s Minor Planet Center.

This work was facilitated through the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system and funded by the STF at the University of Washington.

The authors would like to thank Victoria Meadows, Matthew McQuinn, and Scott Anderson at the University of Washington for their many helpful comments on this work.

The authors are grateful to the anonymous referee for a thorough review of multiple revisions of this manuscript, and for encouraging us to complete and add the orbit determination elements that were originally intended to be a part of a follow-up paper. This has enabled us to deliver a fully functional software package that can be immediately applied for Solar System object searches.

Software: oorb (Granvik et al. 2009), spiceypy (Annex et al. 2020), numpy (Harris et al. 2020), scikit-learn (Pedregosa et al. 2011), astroty (Astropy Collaboration et al. 2013, 2018), matplotlib (Hunter 2007), seaborn (Waskom 2021), plotly (Plotly Technologies Inc., 2015), bokeh (Bokeh Development Team 2018).

Appendix

Transformations, Rotations, and Projections

Here, we summarize the key equations and relations used to carry out the heliocentric transformation and projection into the corotating frame of the test orbit.

A.1. Equatorial and Ecliptic Coordinates

For convenience, we reproduce a subset of the equations presented in Bernstein & Khushalani (2000) that outline the coordinate transformations between the equatorial and ecliptic systems. A set of observations in equatorial coordinates \((\alpha, \delta)\) can be transformed into Cartesian equatorial coordinates as follows:

\[
\begin{bmatrix}
x \\
y \\
z_{\text{ec}}
\end{bmatrix} = \begin{bmatrix}
\cos \delta \cos \alpha \\
\cos \delta \sin \alpha \\
\sin \delta
\end{bmatrix}. \tag{A1}
\]

Similarly, a set of ecliptic coordinates \((\lambda, \beta)\) can be transformed into Cartesian ecliptic coordinates:

\[
\begin{bmatrix}
x \\
y \\
z_{\text{ec}}
\end{bmatrix} = \begin{bmatrix}
\cos \beta \cos \lambda \\
\cos \beta \sin \lambda \\
\sin \beta
\end{bmatrix}. \tag{A2}
\]

Cartesian equatorial coordinates can be transformed to ecliptic Cartesian coordinates. Here, \(\epsilon\) is the obliquity of the ecliptic (\(\epsilon = 84381.8448\)):

\[
\begin{bmatrix}
x \\
y \\
z_{\text{ec}}
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \epsilon & \sin \epsilon \\
0 & -\sin \epsilon & \cos \epsilon
\end{bmatrix} \begin{bmatrix}
x \\
y \\
z_{\text{ec}}
\end{bmatrix}. \tag{A3}
\]

Likewise, ecliptic Cartesian coordinates can be converted to equatorial Cartesian coordinates:

\[
\begin{bmatrix}
x \\
y \\
z_{\text{ec}}
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \epsilon & -\sin \epsilon \\
0 & \sin \epsilon & \cos \epsilon
\end{bmatrix} \begin{bmatrix}
x \\
y \\
z_{\text{ec}}
\end{bmatrix}. \tag{A4}
\]

Using Equations (A1)–(A4), coordinates in either the angular or Cartesian form for both the ecliptic and equatorial systems can be trivially transformed between the respective systems and the different forms. To proceed with the heliocentric transformation, which is described in the next section, THOR transforms all equatorial coordinates to ecliptic coordinates.

A.2. Heliocentric Transformation

Before proceeding, it is useful to define certain quantities in relation to the assumed observing geometry. The location of the observer in the heliocentric frame we denote as \(r_o\), and is assumed to be known. The distance from the Sun to the observer is then simply the magnitude of said vector \(|r_o| = r_o\). The unit vector \(r_{eo}\) lies in the direction of an observed asteroid from the position of the observer. Unless the asteroid’s orbit is well-constrained and known beforehand, we do not know the
distance from the observer to the asteroid. We express the observer to asteroid position vector as \( \mathbf{r}_{ao} \), which has magnitude \( \Delta \) and is equivalent to \( n_{ao} \cdot \Delta \). Last, we denote the unknown heliocentric position vector of the asteroid as \( \mathbf{r}_a \), which has magnitude \( r \).

A fiducial test orbit has heliocentric position vector \( \mathbf{r} \) and velocity \( \mathbf{v} \) given at some epoch, \( t \). The location in space and time of an object on the test orbit can be precisely predicted. Assuming there are other minor planets similar to the test orbit, we can propagate the test orbit to all times where it could have been detected in a survey and gather actual detections near its predicted location (Sections 2.1.1–2.1.2). Nearby detections likely belong to moving objects on similar orbits. If we assume those lie at the same heliocentric distance from the Sun, we have enough information to fully constrain the observing geometry for each individual detection, allowing the unknown quantities (\( \Delta \) and \( r_a \)) to be calculated.

From our assumed observing geometry, we know the direction \( \hat{n}_{ao} \) for every gathered detection. Assuming each detection lies at the distance \( r \) of the test orbit, we can then calculate \( \Delta \):

\[
\Delta = -n_{ao} \cdot r_o + \sqrt{(n_{ao} \cdot r_o)^2 + r^2 - r_o^2}.
\]

Once we know \( \Delta \), we can solve for \( r_a \), where

\[
r_a = r_{ao} + r_v.
\]

At this point, we can attribute a heliocentric position vector to every gathered detection at its specific epoch, again under the assumption that the detection belongs to an orbit similar to the test orbit.

### A.3. Transformation into Frame of the Test Orbit

Once the heliocentric coordinates of each observation have been calculated, the observations of other minor planets can be transformed into the frame of the test orbit. This is accomplished using a double rotation. First, the detections are rotated so that the test orbit lies in the \( x-y \) plane. Second, the detections are rotated so the test orbit lies along the \( x \)-axis.

To perform the first rotation, which we denote as \( R_1 \), we transform the detections so that \( \mathbf{r} \) lies in the \( x-y \) plane. This is equivalent to rotating the vector normal to the plane of the orbit (\( \hat{n} \)) toward the \( z \)-axis as defined in heliocentric space (\( \hat{z} \)). We now follow the method outlined in van den Berg (2016). The rotation axis, \( \hat{v} \), can be found by taking the cross product of \( \hat{n} \) and \( \hat{z} \).

\[
\hat{v} = \hat{n} \times \hat{z}.
\]

The rotation matrix, \( R_1 \), is then

\[
R_1 = I + [\hat{v}]_c + [\hat{v}]^2_c \frac{1}{1 + c},
\]

where \( c = \hat{n} \cdot \hat{z} \) and \([\hat{v}]_c \) is the skew-symmetric cross product matrix of \( \hat{v} \),

\[
[\hat{v}]_c = \begin{bmatrix}
0 & -\hat{v}_3 & \hat{v}_2 \\
\hat{v}_3 & 0 & -\hat{v}_1 \\
-\hat{v}_2 & \hat{v}_1 & 0
\end{bmatrix}.
\]

Finally, we need to rotate the detections so that \( \mathbf{r} \) lies along the \( x \)-axis. We denote this rotation as \( R_2 \). Let \( \hat{r}' \) be a unit vector in the direction of a detection or test orbit rotated into the \( x-y \) plane by \( R_1 \). The angle of rotation toward the \( x \)-axis, \( \alpha \), is equivalent to \(-\hat{x} \cdot \hat{r}' \).

\[
R_2 = \begin{bmatrix}
\cos \alpha & \sin \alpha & 0 \\
-\sin \alpha & \cos \alpha & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]

The complete rotation matrix is \( M = R_2 \cdot R_1 \).

### A.4. Gnomonic Projection

After the observations have been rotated, they can be projected into a gnomonic tangent plane (Weisstein 2001). The gnomonic tangent plane preserves great circle distances and so offers a natural projection in which to discover minor planets relative to a test orbit. The gnomonic projection is also used by both Holman et al. (2018) and Bernstein & Khushalani (2000). The gnomonic projection can be accomplished using either spherical or Cartesian coordinates. For completeness, we list both sets of equations. In our implementation, we directly project the transformed and rotated Cartesian coordinates onto the gnomonic tangent plane.

To project a set of spherical coordinates \((\lambda, \phi)\) onto a gnomonic plane \((\theta_X, \theta_Y)\) tangent about point \((\lambda_0, \phi_0)\) (in our case, the location of the test orbit at each epoch in which detections were found and gathered):

\[
\begin{align*}
\theta_X &= \frac{\cos \phi \sin (\lambda - \lambda_0)}{\sin \phi_0 \sin \phi + \cos \phi_0 \cos \phi \cos (\lambda - \lambda_0)}, \\
\theta_Y &= \frac{\cos \phi_0 \sin \phi - \sin \phi_0 \cos \phi \cos (\lambda - \lambda_0)}{\sin \phi_0 \sin \phi + \cos \phi_0 \cos \phi \cos (\lambda - \lambda_0)}.
\end{align*}
\]

The same projection can be achieved using the rotated Cartesian coordinates described in the previous section. In this case, by convention, the test orbit lies at \((\lambda_0, \phi_0) = (0, 0)\), making the projection simply

\[
\begin{align*}
\theta_X &= \frac{x''}{y''}, \\
\theta_Y &= \frac{z''}{y''}.
\end{align*}
\]

The double prime (‘‘) is used to denote the transformed Cartesian coordinates after they have been rotated as described in Section A.3. After completing the procedure outlined in Sections A.1–A.4, we now have a set of detections and their locations in the corotating reference frame of a test orbit. From this reference frame, other orbits can be recovered as described in Section 2.1.4.

### ORCID iDs

Joachim Moeyens @ https://orcid.org/0000-0001-5820-3925
Mario Juric @ https://orcid.org/0000-0003-1996-9252
Jes Ford @ https://orcid.org/0000-0002-2946-3776
Dino Bektešević @ https://orcid.org/0000-0002-1312-5529
Andrew J. Connolly @ https://orcid.org/0000-0001-5576-8189
Siegfried Eggl @ https://orcid.org/0000-0002-1398-6302
Željko Ivezic @ https://orcid.org/0000-0001-5250-2633
R. Lynne Jones @ https://orcid.org/0000-0001-5916-0031
J. Bryce Kalmbach @ https://orcid.org/0000-0002-6825-5283
Hayden Smotherman @ https://orcid.org/0000-0002-7895-4344
References

Annex, A. M., Pearson, B., Seignovert, B., et al. 2020, JOSS, 5, 2050
Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 132
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Ballard, D. 1981, PatRe, 13, 111
Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019, PASP, 131, 018002
Bernstein, G., & Khushalani, B. 2000, AJ, 120, 3323
Berthier, J., Vachier, F., Thuillot, W., et al. 2006, in ASP Conf. Ser. 351, Astronomical Data Analysis Software and Systems XV, ed. C. Gabriel et al. (San Francisco, CA: ASP), 367
Bokeh Development Team 2018, Bokeh: Python library for interactive visualization, https://bokeh.org/
Denneau, L., Jedicke, R., Grav, T., et al. 2013, PASP, 125, 357
Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. 1996, in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining, ed. E. Simoudis, J. Han, & U. Fayyad (Cambridge, MA: AAAI Press), 226
Gorski, K. M., Hivon, E., Banday, A. J., et al. 2005, ApJ, 622, 759
Graham, M. J., Kulkarni, S. R., Bellm, E. C., et al. 2019, PASP, 131, 078001
Granvik, M., Virtanen, J., Oszkiewicz, D., & Muinonen, K. 2009, M&PS, 44, 1853
Harris C. R., Millman K.J., van der Walt S.J., et al. 2020, Natur, 585, 7825
Herget, P. 1965, AJ, 70, 1
Holman, M. J., Payne, M. J., Blankley, P., Janssen, R., & Kuindersma, S. 2018, AJ, 156, 135
Hunter, J. D. 2007, CSE, 9, 90
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Jones, R. L., Slater, C. T., Moeyens, J., et al. 2018, Icar, 303, 181
Kubica, I., Denneau, L., Grav, T., et al. 2007, Icar, 189, 151
Larson, S., Beshore, E., Hill, R., et al. 2003, AAS Meeting, 35, 36.04
Mainzer, A. K. & NEOCam Science Team 2016, AAS Meeting, 48, 327.01
Masci, F. J., Laher, R. R., Rosholme, B., et al. 2019, PASP, 131, 018003
McKinney, W. 2010, in Proc. 9th Python in Science Conference, ed. S. van der Walt & J. Millman (Austin, TX: SciPy), 56
Milani, A., Gronchi, G. F., Farnocchia, D., et al. 2008, Icar, 195, 474
Minton, D. A., & Malhotra, R. 2009, Natur, 457, 1109
Patterson, M. T., Bellm, E. C., Rosholme, B., et al. 2019, PASP, 131, 018001
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, J. Mach. Learn. Res., 12, 2825
Plotly Technologies Inc. 2015, Collaborative data science, https://plot.ly
Vallado, D. A. 2013, Fundamentals of Astrodynamics and Applications (4th ed.; Hawthorne, CA: Microcosm Press)
van den Berg, J. 2016, Calculate Rotation Matrix to align Vector A to Vector B in 3d?, https://math.stackexchange.com/q/476311
Veres, P., & Chesley, S. R. 2017a, AJ, 154, 12
Veres, P., & Chesley, S. R. 2017b, AJ, 154, 13
Warner, B. D., Harris, A. W., Vokrouhlický, D., Nesvorny, D., & Bottke, W. F. 2009, Icar, 204, 172
Waskom, M. 2021, JOSS, 6, 3021
Weisstein, E. W. 2001, Gnomonic Projection From MathWorld – A Wolfram Web Resource, http://mathworld.wolfram.com/GnomonicProjection.html