Enabling Direct Feedback Between Initial Orbit Determination and Sensor Data Processing for Detection and Tracking of Space Objects

Brad Sease[a], Timothy Murphy[b], Brien Flewelling[c], Marcus J. Holzinger[d], and Jonathan Black[e]

[a]Virginia Tech, 1901 Innovation Dr., Blacksburg, VA, USA;
[b]Georgia Tech, North Ave NW, Atlanta, GA, USA;
[c]Air Force Research Lab, 3550 Aberdeen Ave. SE, Kirtland AFB, NM, USA;
[d]Georgia Tech, North Ave NW, Atlanta, GA, USA;
[e]Virginia Tech, 1901 Innovation Dr., Blacksburg, VA, USA

Abstract

This paper presents an automatic RSO detection and tracking scheme operating at the optical sensor system level. The software presented is a pipeline for processing ground or space-based imagery built from several sub-algorithms which processes raw or calibrated imagery, detects and discriminates non-star objects, and associates observations over time. An orbit determination routine uses an admissible region to start off an unscented particle filter. This preliminary orbit estimate allows prediction of the appearance of the object in the next frame. A matched filter uses this imagery to provide feedback to the initial detection and tracking process.

Keywords: Space Situational Awareness, Image Processing, Orbit Determination

1. INTRODUCTION

There exists a need for improved Space Situational Awareness (SSA) technologies to support US interests in space as described by the 2001 Rumsfeld Report.[1] The Joint Space Operations Center (JSpOC) under U.S. Strategic Command operates The Space Surveillance Network (SSN) which tracks upwards of 17000 space objects of diameters greater than 10 cm in the space object catalog (SOC).[2] The SSN takes around 400,000 observations each day with radar, optical, and space based sensors.[2] Space and ground based optical sensors can be seen as an affordable alternative to radar technologies. They also play a key role in tracking objects outside of Low Earth Orbit (LEO) where radar is less effective. While this technology has been useful in mitigating risk for space missions, the need for better space object tracking technologies can be seen by high profile events such as the 2009 Iridium Cosmos spacecraft collision.

The SSN includes both exquisite and minimalist sensors. Minimalist sensors use cost driven design, allowing larger sensor networks but often lacking the necessary noise requirements for detection of dim objects. Design of exquisite sensors is driven by their excellent noise characteristics, enabling dim object detection, but often leads to restrictive cost. Resident space objects (RSO) also take a wide variety of forms. To an optical sensor a bright unresolved RSO may be anywhere from an extremely reflective but small SO to a large heavy SO with low reflectivity. An optical sensor's ability to make measurements on a RSO is typically characterized by the object signal to noise ratio (SNR), which is roughly the ratio of photons from a RSO to the photons from noise sources. Therefore, exquisite sensors are those that have minimized noise and therefore maximized SNR.

Exquisite sensors are needed to discover new objects but are a limited resource. Minimalist sensors are ideal for discovery and catalog upkeep when available. The threshold for detection is measured by SNR and so any computational methods which can increase or maximize SNR are very relevant to improving the performance of low-cost assets. This motivates concept of operations (ConOps) and processing techniques for enabling low SNR detection for passive sensors. From a ConOps perspective there are two broad categories, active and passive sensing. Rate tracking an object (slewing so that the RSO stays in the same pixel) provides a boost to SNR

by allowing the photons from an objects to all fall in the same few sensor pixels. The catch to tracking is that without good knowledge of the RSO's orbit, active sensing is difficult. Sidereal stare sensing is therefore necessary during RSO discovery, that is, finding RSO not already in the SOC. With sidereal stare sensing, there exist processing solutions to allow low SNR detections. The matched filter (MF) is a signal processing filter which provides optimal SNR gain. It can be applied to images with low SNR signals to increase detections, but requires prior knowledge of signal shape.[3] A MF requires a template for the predicted signal shape which further requires orbital knowledge of some type. The saving grace is that MF implementation can be possible with very limited orbital knowledge.[4]

In current optical SSA applications, tracking, image processing, and orbit determination (OD) are often disjoint processes. In the standard tracking process, measurements are taken on an object without any online orbit update. Then, after tracking and data collection are complete, orbital parameters are estimated via a batch process or multiple hypothesis filter type method. If the entire process of tracking, image processing, and OD are combined, there exist ways to improve this algorithm through real-time feedback. The work presented here serves to demonstrate an algorithm for providing direct feedback between sensor-level analysis and orbit determination processes. Further, we aim to show that these typically distinct processes become more robust with direct feedback.

The Generalized Electro-Optical DEtect, Track, Identify, and Characterize Application (GEODETICA) is a tool that aims to incorporate a variety of SSA image processing, estimation, and tasking tools into one streamlined application. The first goal of GEODETICA is to integrate tracking, orbit determination, and image processing into one application to enable inter-connectivity and feedback. This paper in particular will demonstrate this through a standard tracking process enhanced with orbit determination enabled matched filtering. GEODETICA also aims to explore more general ways to implement SSA software to allow less restrictive ConOps.

The paper will first present an algorithm that merges sensor-level image processing with on-line orbit determination. This will include discussion of an image point tracker built on a bank of Kalman filters with frame-to-frame association enabled by a global nearest neighbor process, an unscented particle filter for orbit determination, and a matched filter process designed to augment the point tracker. The results presented in the paper show that the sensor-level measurement process benefits from an on-line orbit estimate by enabling more measurements. We demonstrate the algorithm is capable of observing difficult-to-track objects such as spinning spacecraft. The algorithm additionally enables us to hand-off to lower fidelity sensors to free up primary sensors for high-priority targets.

2. METHODOLOGY

2.1 Object Tracking & RSO Discrimination

The GEODETICA process begins with a source extraction routine operating on a single image. The source extraction routine relies on a process known as edge detection to detect objects. Edge detectors highlight regions in an image where strong 1-directional gradients exist.[5] This is a powerful tool for space-based imagery because both stars (point-sources) and streaks exhibit a strong edge response. GEODETICA specifically uses the method of phase congruency due to its robustness to image gradients and normalized response range between 0 and 1.[6] We threshold the phase congruency edge response at a user-defined level and centroid all of the contiguous regions remaining.[7] We run this source extraction routine for each successive image and use that data to inform the rest of the GEODETICA routine.

The standard GEODETICA object tracking code is built on the approximate pin-hole camera model. The projection $T : \mathbb{R}^3 \to \mathbb{R}^2$ is given by the Collinearity equation,[8]

$$\begin{pmatrix} x \\ y \end{pmatrix} = -\frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix} \tag{1}$$

where f is the focal length of the optical system, (X, Y, Z) is the coordinate of an imaged particle in the non-inertial reference frame fixed to the camera boresight, and (x, y) designates the corresponding image-plane coordinate. With this simplified model we are able to examine the expected motion profile for certain objects within a camera's field of view (FOV). Point sources extracted from space-based imagery typically correspond to

either stars or other resident space objects (RSOs). With this knowledge, we make some assumptions about the type of motion we expect for a particular object. The proper motion of a star is on the order of milliarcseconds per year (mas/yr).[9] Therefore, we assume that the motion of a star signal in the image plane is completely defined by the attitude dynamics of the sensor. If we further assume that the angular rate of the sensor is approximately constant over the integration time of the camera, then we are able to project the angular kinematics through the Collinearity equation. With a small field of view assumption we can cancel terms to arrive at an approximate motion model of the form

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \approx \begin{pmatrix} 0 & \omega_3 \\ -\omega_3 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + f \begin{pmatrix} \omega_2 \\ -\omega_1 \end{pmatrix} \tag{2}$$

where $(\omega_1, \omega_2, \omega_3)$ is the inertial angular rate of the sensor. Note that, for a constant angular rate, this is a linear motion model expressed entirely in the image plane. Now, for any star we localize in the image plane, our measurement model is simply

$$h(x, y) = \begin{pmatrix} x \\ y \end{pmatrix} \tag{3}$$

In other words, with our motion model expressed entirely in the image plane, the corresponding measurement model is a direct measurement of the system states. It is trivial to verify that this system is fully observable. A similar derivation follows for the motion model corresponding to RSOs. For a particular RSO, the apparent motion of the object is not necessarily negligible, as it was in the case of stars. Therefore we require a model that captures a broader range of motion profiles. For RSOs, then, we consider a translational motion model in the body-fixed reference frame of the sensor. With the proper choice of acceleration inputs this model perfectly captures orbital motion. When we project this dynamic model into the image plane through Equation 1, however, we find that, for distant objects, these accelerations can be approximated as zero. Thus we arrive at a motion model for RSOs, described as the "general" motion model, given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{4}$$

which describes translational motion in the image plane. As in the case of the star motion model, we have obtained an expression for linear dynamics expressed fully within the image plane. For a point source localized in the image plane, our measurement model for this system is

$$h(x, y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix} \tag{5}$$

In this motion model, the state contains image-plane velocities in addition to the (x, y) coordinates. Thus for the general model we do not measure the full state – only the position substates. It is simple to verify, however, that for the general system the observability matrix is full-rank and the system is fully observable. It should be noted that because a typical measurement is taken over an non-zero exposure time, information can be collected on the rates of the object. This information is currently not used by the GEODETICA algorithm but could be incorporated in future improvements.

The heart of the GEODETICA process is a bank of multiple-hypothesis Kalman filters – one for each detected object – which filter the image-plane states according to these motion models.[8] For each sequential image frame, any uncorrelated detection initiates a new track. Each new track contains two possible hypotheses: the dynamics of the corresponding object evolve according to either the star motion model or the general motion model. Initial uncertainty, measurement noise, and process noise are all user-defined parameters based on the expected performance of the optical system. Currently, objects are initiated with a positional uncertainty defined by the expected localization performance of our algorithm on the particular optical system it is operating on. For the translational model, we assume an initial estimate of zero velocity with a large uncertainty, due to the fact that we have no velocity knowledge until a secondary detection has been made. The choice of initial covariance

for the translational model is a trade-off between the ability to detect fast moving objects and prevention of false associations in crowded fields. A small user-defined process noise is added to both models to account for approximation error and other small disturbances.

We propagate all tracks forward for each frame-to-frame time interval and then perform a global nearest-neighbor (GNN) data association process.[10] This means that each track is associated with the measurement nearest to its predicted location. Further, no two tracks can share a measurement, so the closest track to a given measurement "wins" while the other is considered unassociated. Only multiple hypotheses for the same object can associate with a measurement simultaneously.

Hypotheses are pruned with the first successful measurement association after the track is initiated. The algorithm conservatively assumes that if the star motion model makes a successful association, then the corresponding object must follow that model. Even if the general model simultaneously associates with a measurement, the algorithm accepts only the star motion hypothesis. On the other hand, if the general model makes an association while the star motion model does not, then the algorithm accepts the general model for that object. We perform general pruning of outdated tracks according to an uncertainty threshold. When the position uncertainty of a track grows beyond the bounds of a user-defined threshold, that track is considered "lost" and discarded.

Throughout this process, we have some indication as to which objects are stars and which are RSOs, however some ambiguity remains. Though we assume that objects associated with the star motion model are stars, it is possible for a star to correctly match with the general motion model. We see from Equation 2 that for $\omega_3 = 0$, a star's motion is perfectly captured by the general motion model. So, while we can say that an object following the star motion model is not an RSO, we cannot say that an object following the general model is an RSO. Thus we perform a further discrimination process to separate objects with apparent motion relative to the stellar background. This secondary process utilizes an estimate of the frame-to-frame rotation of the sensor.[11] This can either be provided by an on-board gyroscope or estimated on-line from the ensemble motion of the objects in the frame. The latter option relies on the assumption that most of the objects in any particular image are stars. From an estimate of the frame-to-frame rotation, we are able to compute the residuals for each object as

$$\epsilon_i = \left\| \boldsymbol{b}_i - \hat{A}\boldsymbol{a}_i \right\|_2 \tag{6}$$

where $\boldsymbol{b}_i \in \mathbb{R}^3$ denotes the body-fixed unit vector of the ith object in the previous image, $\boldsymbol{a}_i \in \mathbb{R}^3$ is the body-fixed unit vector of the ith object in the current image, and $\hat{A}$ is the estimated rotation of the sensor from the last frame to the current.[12] Outlying residuals point to objects which may not be inertially-fixed, indicating that the object may not be a star. The tracking algorithm uses a voting process based on these residuals to discriminate non-star objects. Beginning at zero, for each time step that an object produces an outlying residual, its "RSO score" is incremented by 1. For each step that an object appears to follow the ensemble motion, its RSO score is decremented by 1. Any object with a positive RSO score is considered to be an RSO score, while any object with an RSO score ≤ 0 is considered a star.

2.2 Orbit Determination Method

The GEODETICA tracking algorithm described above can operate in a stand-alone manner. The following discussion is a supplemental code which uses an unscented particle filter to perform real time orbit determination. Consider a space object with position and velocity $\mathbf{r}$ and $\mathbf{v}$, seen by an optical observer at position and velocity $\mathbf{o}$ and $\dot{\mathbf{o}}$ along observation vector, $\boldsymbol{\rho}$. The general orbit determination problem is illustrated in Figure 1.

For a general optical observer, measurements will consist of a unit vector $\hat{\boldsymbol{\rho}}$, co-aligned with the vector $\boldsymbol{\rho}$. A variety of estimator techniques can be used to approximate $\mathbf{r}$ and $\mathbf{v}$ from a series of data points. For real time OD problems, an extended Kalman filter is often used, but recently more advanced nonlinear methods are becoming more popular.[13,14] One such method, the particle filter (PF) has the advantage of modeling the random variable probability density function (PDF) as a cloud of virtual particles (VP). The virtual particles each have a weight associated with them which allows a PDF to be approximated by the weighted density of particles. This takes away the Kalman filter assumption that the estimate is a Gaussian random variable, allowing the estimate PDF to take on the shape dictated by the dynamics. VP are discrete points and so the transformation of a PDF to a different space, such as orbital elements is relatively straight forward. In orbital element space

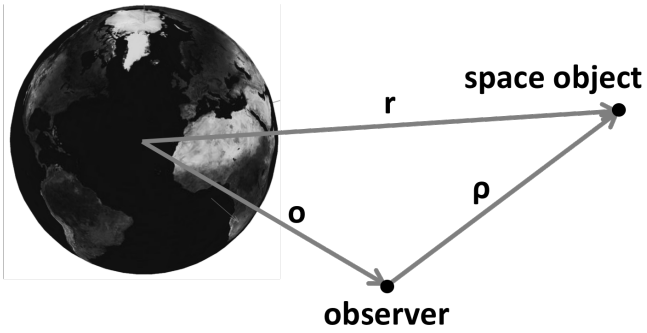

Figure 1. Setup of general observer problem.

specifically, the estimate PDF exhibits highly non-Gaussian properties. The disadvantage of using a PF is that the implementation requires creation of thousands of VP each of which must be propagated through the dynamics being used. The number of required VP is dependent on the dimensionality of the problem. This problem is refered to as the "Curse of Dimensionality". This makes a PF computationally costly and unattractive for orbit determination applications. However, the discrete nature of a particle filter makes it parallelizable, reducing computational costs. The use of methods such as admissible regions (AR) mitigate the need for large numbers of VPs by allowing for an initial sampling in a two dimensional subspace of $\mathbb{R}^6$.[15] These improvements have made PF methods tractable, facilitating their rise in popularity.

For uncorrelated tracks, the initial distribution is created by sampling an admissible region created from the first and second measurements. An admissible region allows a single uncorrelated short arc measurement consisting of the unit vector and its rate, $[\hat{\boldsymbol{\rho}}\ \dot{\hat{\boldsymbol{\rho}}}]^T$, to be used to create a constrained subspace of $\mathbb{R}^6$. This subspace is typically modeled with a uniform PDF, assuming that all possible orbits are of equal probability. For applications like a PF, this uniform PDF is sampled, creating the initial cloud of VP.[16] This paper uses an Unscented Particle Filter (UPF), which is a revision to the original PF which includes an internal estimator acting on each particle.

The admissible region will be defined more rigorously here. An optical measurement taken by an observer consists of right ascension, declination, and their corresponding rates. These have a one-to-one and onto transform to the observation unit vector and its rate of change, $\hat{\boldsymbol{\rho}}$ and $\dot{\hat{\boldsymbol{\rho}}}$, or equivalently two angles and two angle rates. The angular states that can be represented as $\mathbf{x}_d = [\alpha\ \delta\ \dot{\alpha}\ \dot{\delta}]^T$. Then, the unknown states can be written as $\mathbf{x}_u = [\rho\ \dot{\rho}]^T$. The full state can then be written as

$$\mathbf{x}' = \left[\mathbf{x}_d^T\ \ \mathbf{x}_u^T\right]^T \tag{7}$$

It is easy to see that a one-to-one and onto mapping, $\mathbf{m}$, exists between $\mathbf{x}'$ and $\mathbf{x}(t) = [\mathbf{r}^T\ \ \mathbf{v}^T]^T$.

$$\mathbf{x}' = \mathbf{m}(\mathbf{x}; \mathbf{k}) \tag{8}$$

where $\mathbf{k}$ is a parameter vector containing $\mathbf{o}$ and $\dot{\mathbf{o}}$. It should be noted that all measurements and representations of orbits discussed here are time dependent but the explicit dependence will not be shown. Also note that $\mathbf{x}$ is defined as position and velocity but could also be written as orbital elements or any other full characterization of an orbit. The transform between these two is in essence a reference frame change between geocentric Cartesian frame and the observer-centric spherical frame. This mapping requires knowledge of $\mathbf{o}$ and $\dot{\mathbf{o}}$, which will be modeled as Gaussian random variable. Uncertainties in observer position, measurement error and timing error can also be taken into account.[17]

The basis of the admissible region is that the unknown parameters, $\mathbf{x}_u$, can be sampled instead of a full sampling of the state space. An AR is then created by enforcing a series of constraints on $\mathbf{x}_u$ of the form

$$g_i(\mathbf{x}_d, \mathbf{x}_u; \mathbf{k}, t) \leq 0 \tag{9}$$

The AR is then the space where all hypotheses are true. Next, we define an admissible region set, $\mathcal{AR}_i \in \mathbb{R}^2$, predicated on g_i

$$\mathcal{AR}_i := \left\{ \mathbf{x}_u \mid g_i(\mathbf{x}_d, \mathbf{x}_u; \mathbf{k}, t) \leq 0 \right\} \tag{10}$$

In practice, an admissible region predicated on n constraints is used.

$$\mathcal{AR} = \bigcap_{i=1}^{n} \mathcal{AR}_i \tag{11}$$

The most commonly used constraints, g_i, are on orbital energy[18] and minimum radius of periapse[19] and omit hyperbolic and ballistic trajectories respectively. In practice, this gives a set in $\mathcal{AR} \in \mathbb{R}^2$ which defines all orbits which are consistent with the single measurement obtained. The practical use of an admissible region is the ability to use it as a prior distribution for a variety of filters and methods. $\mathcal{AR}$ is sampled through some methodology which could include but is not limited to random sampling, some predetermined grid, Delauney triangulation, or Gaussian mixture models. In general, an even spacing of points within the AR is desirable.[20,15] As mentioned, these samples will become the initial distribution of VP for the particle filter.

The following will begin the discussion of the particle filter. In,[21] the Unscented Particle Filter (UPF) is developed and proposed as a better suited method for non-linear systems. The UPF represents each VP as a Gaussian random variable instead a just a particle. This requires the propagation of covariance for each particle which is done with the unscented transform. This transformation models a Gaussian random variable by choosing $1 + 2N$ "Sigma Points".[22] These points can be propagated through any mapping. Third order accuracy is obtained by this estimators for Gaussian inputs. Computationally, the UKF requires that each sigma point for each VP be propagated forward for each time step giving an increase of $1 + 2N$ computational cost. Typical covariance propagation requires a costly Jacobian calculation at each time step, so use of the unscented transformation is justified. The UPF is used over a standard PF for its better modeling of non Gaussian distributions and naturally parallelizable construction.

The UPF is initialized with VP from an admissible region with the k^{th} particle having an associated weight, w_i. Particle weights can be thought of as the statistical likelihood that a particle is "correct" and allow the PDF to be modeled as a weighted particle density rather than just particle density. The weights should always be normalized such that they sum to 1. For each iteration of the particle filter, the following occurs. A new measurement at a time t_k is obtained. Each particle is then run through one iteration of an unscented Kalman filter[21] using this new measurement to update, giving an updated state $x_i(t_k)$. The updated VP are mapped to the measurement space via a measurement function $y = h(x(t_k)) + \epsilon$ and are then compared to the measurement via a multivariate Gaussian distribution. The weight of each particle is then updated as the product of the previous weight and the new probability. This allows time history to be incorporated into the weights.

$$p(\mathbf{z}(t_k)|\mathbf{x}_i(t_k)) = \frac{1}{\sqrt{(2\pi)^l |\mathbf{P}_i(t_k)|)}} \exp\left(-\frac{1}{2} (h(\mathbf{x}_i(t_k)) - \mathbf{z}(t_k)) \mathbf{P}_i^{-1}(t_k) (h(\mathbf{x}_i(t_k)) - \mathbf{z}(t_k))^T \right) \tag{12}$$

$$\mathbf{P}_i(t_k) = \mathbf{R}(t_k) + \mathbf{H}(\mathbf{P}_i(t_k)) \tag{13}$$

$$w_i(t_k) = w_i(t_{k-1}) p(\mathbf{z}(t_k)|\mathbf{x}_i(t_k)) \tag{14}$$

where the measurement has l dimensions, $\mathbf{x}_i(t_k)$ is the i^{th} VP at time step k, $\mathbf{z}(t_k)$ is the new measurement at timestep k, $\mathbf{R}(t_k)$ is the measurement noise covariance at time step k, $\mathbf{P}_i(t_k)$ is the covariance of a the i^{th} VP, and w_k^i is the weight assigned to the i^{th} particle at time step k. $\mathbf{H}(\mathbf{P}_i(t_k))$ is a mapping of the covariance of a VP to the measurement space and can be done with the unscented transform. The weights are then re-normalized such that the total distribution sums to 1.

VP deleting and resampling occur next. A threshold is set to determine which particles to delete. When $w_k^i < w_{min}$ is satisfied, a particle is deleted. The threshold is highly dependent on the desired distribution size; larger distributions require lower thresholds. Resampling occurs when the number of particles falls below a

certain threshold at which point it is considered too few particles to accurately represent the distribution. Each particle is resampled based on the following

$$N_i(t_k) = \frac{w_i(t_k)}{C} \tag{15}$$

where $N_i(t_k)$ is the number of new VP to be sampled from the i^{th} parent VP, and C is some given threshold based on the size of the desired distribution. N_i is rounded down, and N_i new VP are sampled from the Gaussian distribution of the i^{th} VP. Note that this assures that no current VP is deleted but instead only new VP are created. Finally, the weights $w_{i+1}(t_k)...w_{i+N_i}(t_k)$ of the newly sampled VP are re-normalized based on the parent VP weight, $w_i w_{i+1}(t_k)$, by making each VP weight the following

$$w_{\hat{i}}(t_k) = \frac{w_i(t_k)}{N_i + 1} \tag{16}$$

where $\hat{i}$ are the indexes for only the VP sampled from $w_i(t_k)$, including $w_i(t_k)$ itself.

Ultimately, the output of the particle filter is a cloud of Gaussian random variables which can represented via the unscented transformation in any well defined representation of orbits. How to use these VP is a matter of preference, as normal statistical measures such as mean and variance may not be useful.

2.3 Sensor Feedback

The GEODETICA process and the orbit determination routine are two typically disparate processes. Here, we combine them on-line to produce a more robust algorithm. Figure 2 contains a visual representation of the feedback algorithm. The sensor feedback process begins with the GEODETICA point tracking process. Initially, the algorithm detects an object and begins tracking it frame-to-frame. After sufficient measurements have been taken to confidently (by a user-defined threshold) classify the object as an RSO, the algorithm passes RA, DEC, and rate data to the orbit determination process. The orbit determination process then primes a particle filter as described in the previous section. This particle filter data is stored in the track data for that particular object and updated as additional measurements are made. This orbital knowledge allows us to predict the appearance of the object in successive frames, enabling the use of a matched filter to make detections even when the SNR of the object decreases significantly.

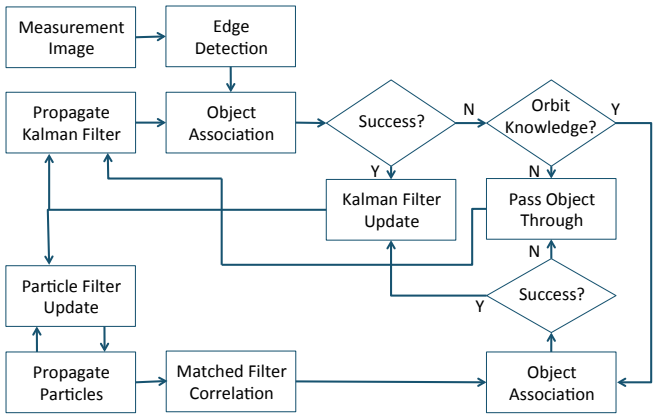

Figure 2. Control loop for matched filter sensor feedback.

It should be noted that these results do not require a particle filter to obtain orbital estimates, only online orbital knowledge. Sensor feedback is performed with a matched filter (MF). Matched filtering originates in image processing and signal processing as a general method for processing signals to allow low SNR detections. The original formulation dates back to the 1960s as a way to detect a signal in a high noise environment.[23] It was then adapted to general image processing and moving target detection.[24]

First, define a measurement as $\mathbf{Y}(t_1, t_0) \in \mathcal{Y} \subset \mathbb{R}^{d_1, d_2}$ which is a matrix in the case to be observed, but could in a more general derivation be of a variety of forms. $\mathbf{Y}(t_1, t_0)$ can be thought as a matrix of the pixel counts from a measurement made by an optical sensor with (t_1, t_0) representing the beginning and end time of the observer exposure. $\mathbf{Y}(t_1, t_0)$ is composed of a signal of interest $\mathbf{S}(t_1, t_0) \in \mathcal{Y}$ and a Gaussian white noise signal as $\mathbf{W}(t_1, t_0) \in \mathcal{Y}$ as shown in Equation 17. $\mathbf{S}(t_1, t_0)$ is the pixel photon counts directly from the SO, while $\mathbf{W}(t_1, t_0)$ is the pixel photon counts from a combination of sensor noise and other objects in the sky. The MF is predicated on a template $\mathbf{S}_0(t_1, t_0) \in \mathcal{T}$, which is typically an approximation of the signal contained in $\mathbf{S}(t_1, t_0)$. $\mathbf{S}_0(t_1, t_0)$ is a smaller or equal dimension matrix than $\mathbf{S}(t_1, t_0)$, that is $\mathcal{T} \subset \mathcal{Y}$, due to the fact that the signal of interest does not always span the entire optical frame. It should be noted that $\mathbf{S}_0(t_1, t_0)$ is not always a function of the exposure time, but could be predicated on any number of factors. Typically though, for optical observation, the signal is dependent of the timing on the exposure making $\mathbf{S}_0(t_1, t_0)$. A matched filter can then be defined as a mapping $f_{MF} : \mathcal{Y} \times \mathcal{T} \to \mathcal{Y}$, shown in Equation 18.

$$\mathbf{Y}(t_1, t_0) = \mathbf{S}(t_1, t_0) + \mathbf{W}(t_1, t_0) \tag{17}$$

$$\mathbf{Y}'(t_1, t_0) = f_{MF}(\mathbf{Y}(t_1, t_0), \mathbf{S}_0(t_1, t_0)) \tag{18}$$

In practice, the function f_{MF} is a correlation of $\mathbf{S}_0(t_1, t_0)$ centered on every pixel in $\mathbf{Y}(t_1, t_0)$ resulting in $\mathbf{Y}'(t_1, t_0)$. The SNR of $\mathbf{Y}(t_1, t_0)$ and $\mathbf{Y}'(t_1, t_0)$ at a certain coordinate $\mathbf{z}$ are defined as:

$$SNR_{Y,z} = \frac{\mathbb{E}[\mathbf{Y}(\mathbf{z}, t)]^2}{\mathbb{E}[(\mathbf{Y}(\mathbf{z}, t) - \mathbb{E}[\mathbf{Y}(\mathbf{z}, t)])^2]} \tag{19}$$

$$SNR_{Y',z} = \frac{\mathbb{E}[\mathbf{Y}'(\mathbf{z}, t)]^2}{\mathbb{E}[(\mathbf{Y}'(\mathbf{z}, t) - \mathbb{E}[\mathbf{Y}'(\mathbf{z}, t)])^2]} \tag{20}$$

From here on the dependence on exposure time, (t_1, t_0), will be assumed. It has been shown that the correlation-based filter which is predicated on an accurate template at the correct coordinate, will produce an optimal SNR gain $K_{SNR} = SNR_{Y',z}/SNR_{Y,z}$.[25] This problem is solved and built into MATLAB, making implementation of a MF simple. A good discussion of this with respect to optical systems is presented by Dragovic.[26] The utility of the MF is enabling low SNR measurements in $\mathbf{Y}(s)$ to be used if the signal shape is known.

Previous work has shown the utility of an orbital mechanics based MF.[4] Assume there exists a representative hypothesis orbit of a SO, $\mathbf{x}(t)$ is a given frame. This hypothesis can then be mapped to the measurement frame of the relevant observer. This distribution can also be mapped forward in time using the known flow function, or integration the orbit.

By varying t, $\mathbf{x}(t)$ can be mapped over the integration time of the observer. A hypothesis with such a mapping forms an arc A,

$$A = \{r \in \mathbb{R}^3 : r = \boldsymbol{\phi}_r(t, t_0, \mathbf{x}(t_0)), t \in [t_0, t_f]\} \tag{21}$$

where $\boldsymbol{\phi}_r$ is a truncated version of $\boldsymbol{\phi}$, providing only position. Similarly, A_m can be defined as an arc through the measurement frame,

$$A_m = \{x_m \in \mathbb{R}^2 : x_m = \mathbf{m}_i(\boldsymbol{\phi}_r(t, t_0, \mathbf{x}(t_0))), t \in [t_0, t_f]\} \tag{22}$$

This arc, A_m can be discretized and mapped into pixels forming an arc through an image, creating a template $\mathbf{S}_0(t)$. The Collinearity equations or another sensor model can be used to map a point or arc in space into an optical sensor image frame.[27] This process allows the simulation of a measurement image, which can be used as $\mathbf{S}_0(t)$.

The matched filter allows us to make detections which would typically be dropped by a tracking algorithm. The algorithm uses the standard GEODETICA tracking algorithm as the default mode. When the tracking algorithm fails to find an object in a new frame, and some orbital knowledge is available, a MF is attempted. The mean orbital estimate from the particle filter is propagated forward and used to create a template. The MF scans the entire image with the template and attempts to find correlation peaks. These correlation peaks are treated in the same fashion as the edge detector results; if a peak is found close to the predicted object location, it is used to update the Kalman filter. This methodology does not rely on the MF to make detections but rather falls back on it, enabling detections that would otherwise be impossible.

3. SIMULATED SCENARIOS

Now we will demonstrate the benefits of the proposed algorithm in three potential use cases. In the first section we demonstrate that, with the proposed algorithm, we are capable of generating preliminary orbit estimates on-line, at the sensor level. Further, these estimates allow us to make in increased number of detections. In Section 3.2, we present a scenario where the observed object is tumbling and has a dynamically varying visible magnitude. In the simulated case, the point tracking algorithm alone will fail when the RSO intensity falls below its detection threshold. We will show that a matched filter built from the on-line orbit estimate allows us to detect and track the object despite its widely varying signal to noise ratio (SNR). In the final use case, in Section 3.3, we examine the possibility of generating initial orbital knowledge from a high-quality sensor and then "handing off" to a lower quality sensor. We will show that the matched filter process allows us to detect much dimmer objects in the low quality sensor, freeing up more sensitive equipment for other targets.

In the following examples we will use a resident space object with the orbital parameters shown in Table 1. The spacecraft is in a medium-Earth, near-circular orbit.

RSO Orbital Elements	
Semimajor Axis	9000 km
Eccentricity	0.01
Inclination	30°

Table 1. Orbit parameters for all simulated scenarios.

We also maintain consistent optical system parameters for each of the simulations discussed below. These parameters are shown in Table 2. The Tycho-2 star catalog provided a simulated star field.[28] The density of the catalog led to the choice of a 2° field of view to ensure that sufficient star signals were contained in each frame.

Optical System	
Field of View	2°
Focal Length	293 mm
Pixel Pitch	20 μm
Resolution	512 pixels

Table 2. Optical system parameters for all simulated scenarios.

3.1 Typical Tracking Case

In this section, we outline a simple operating scenario where the main GEODETICA code tracks an object frame-to-frame while the orbit determination component produces an on-line estimate of the orbital elements. For the purposes of this paper we focus on the detection rates of the standard method compared with our augmented method. In this first scenario, we initialize an object with the orbital parameters shown in Table 1. The optical system follows the spacecraft with sequential step-stare maneuvers, producing a streak near the center of the image in each frame. As the spacecraft approaches, its apparent motion increases. This causes the streak to elongate, reducing it signal-to-noise ratio (SNR). This SNR behavior is plotted as the solid black line in Figure 3. The background shading in Figure 3 highlights the types of detections that were made at each time step. Light gray columns indicate that the detection made at that time step was from the "standard" detection method, which refers to the edge detection method described in Section 2.1. Dark gray columns indicate a detection provided by the matched filter process explained in Section 2.3. Recall that matched filter detections are only used when the standard process fails to produce a measurement. White columns indicate that no detection was made by either the standard or the matched filter process during that time step.

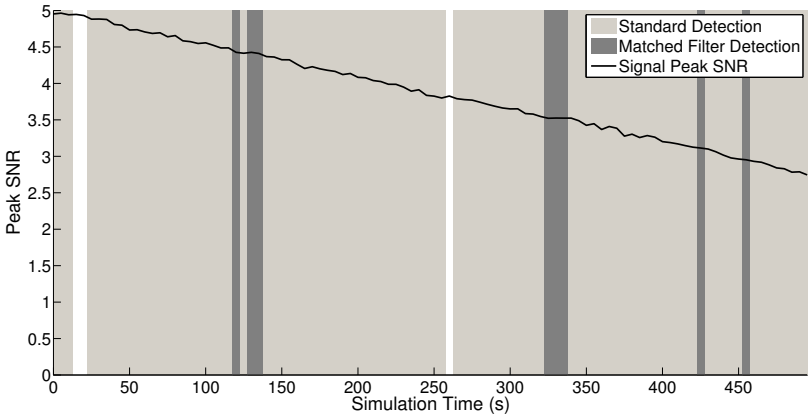

Figure 3. RSO signal-to-noise ratio and detection method.

We see that, for the most part, the standard detection process is able to detect and track the object through the imagery. In some cases the object may overlap with another celestial object or become otherwise corrupted do to image noise. An example of an overlapping signal event is shown in Figure 4. In these cases, the matched filter detection process supplements the algorithm to make the detection. Each matched filter detection in Figure 3 corresponds to a measurement that otherwise would have been missed by the standard process.

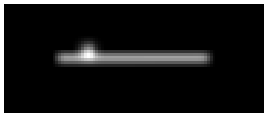

Figure 4. Example of an observation corrupted by overlapping signals.

In Figure 5 we examine the localization performance of the two processes. This plot shows the localization error of each measurement, in pixels, for both standard and matched filter detections. As in the previous plot, the background shading for a particular time step indicates which detection process provided the measurement. Localization error is computed from the true RSO position at the midpoint of the image integration time as projected into the image plane. This provides a set of image (x, y) coordinates that coincide with the ideal centroid of the overall streaked signal. Localization error is presented as the Euclidean distance of the measurement from the truth. We see that the largest sources of error are the three missed detections, where the algorithm is forced to propagate based on prior knowledge. Average error is on the order of 1 pixel, however a slight increase is apparent as the SNR of the signal decreases over time. One matched filter detection produces an outlying measurement, however the remaining produce similar localization accuracy to the standard process.

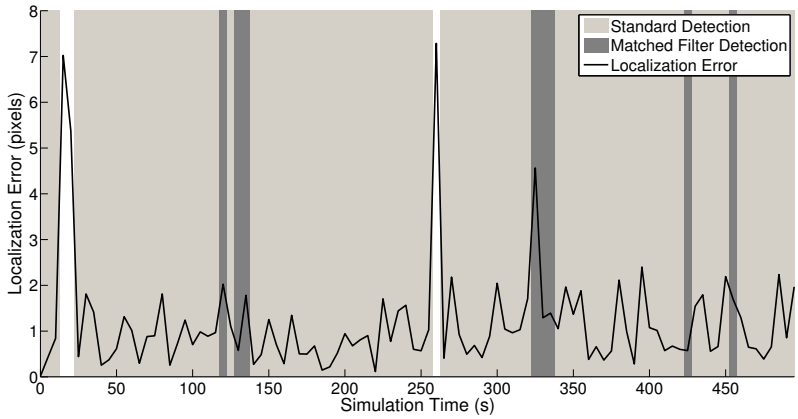

Figure 5. Pixel localization error and detection method.

3.2 Tracking a Rotating Object

Now we consider a scenario where the benefits of the augmented algorithm are more apparent. In this case, we have an RSO with dynamically varying brightness. This is modeled as a sinusoidally-varying visible magnitude, M_v, given by the relation

$$\begin{aligned} M_v(t) &= M_v(0) + A\cos\omega t \\ &= 7.2 - \cos\left(0.0077t\right) \end{aligned} \tag{23}$$

which represents the photometric behavior of a spin-stabilized spacecraft. The spacecraft is initially a magnitude 7.2 object, varying with a range of ± 1 and a period of approximately 130 seconds. As in the previous example, a ground-based observer tracks the object with sequential step-stare maneuvers, creating a streak in the center of each image. The peak SNR profile for this case is shown in Figure 6.

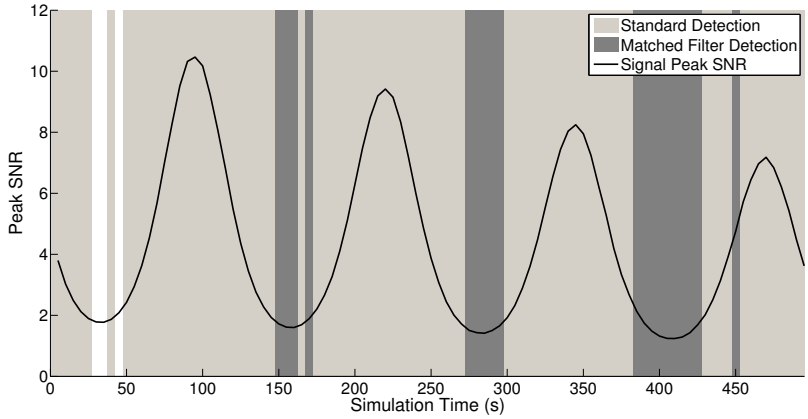

Figure 6. RSO signal-to-noise ratio and detections method.

We see that the standard detection method loses the object prior to initiation of the particle filter. In the first several time steps, the GEODETICA algorithm works to track the object and successfully classify it as an RSO. The first SNR minimum occurs after the object has been detected, but before the particle filter has been generated. Since there is no associated orbital knowledge for the object, a failure in the standard detection method means that no detection is made for that time step. For subsequent dips in SNR, the matched filter generated from the particle filter successfully augments the standard method. We also observe a similar linear

decrease in peak SNR as in the previous example – as the object approaches the observer, its apparent motion increases and the resulting streak lengthens. This effectively spreads the finite signal over a larger pixel area, reducing the signal strength in any particular pixel. As the simulation time advances, each subsequent drop is SNR is more pronounced, requiring the matched filter process to pick up more detections with each cycle.

The localization error for this case is shown in Figure 7. As before, localization error is measured as the Euclidean distance of the measured centroid from the true centroid. We see an average localization error of approximately 1.5 pixels, with the peak error corresponding to a missed detection at $t = 45$ s. The important thing to note in Figure 7 is that both methods produce a measurement of similar precision. Thus the augmented method enables us to accurately observe a spinning space object with widely varying SNR. All measurements made by the matched filter process correspond to detections that would have otherwise been missed with a standard tracking algorithm alone.

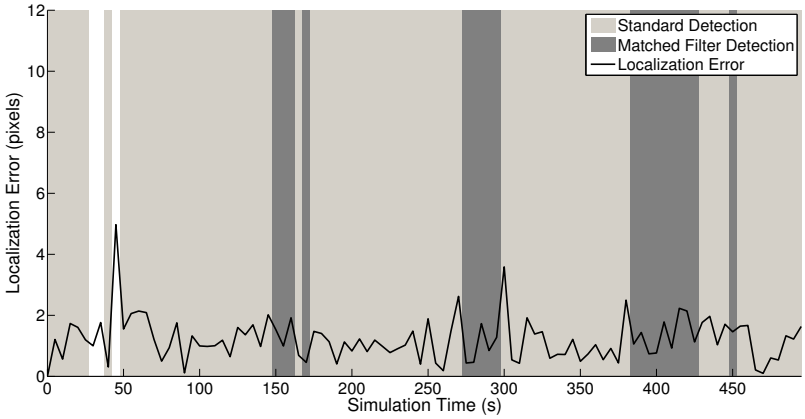

Figure 7. Pixel localization error and detection method.

3.3 Noisy Sensor Hand-Off

In the final example we consider a "step input" to the sensor noise, where the noise increases drastically halfway through the simulation time. This scenario is intended to represent a process where a high-fidelity sensor tracks an object to a desired certainty and then "hands off" to a lower-fidelity sensor, however a similar case occurs when an observed object initially passes into eclipse. We model this with two identical optical systems with different noise properties. The first system is assumed to contain a low-noise CCD while the secondary sensor utilizes a much lower quality CCD. In this simulation, the object undergoes approximately a 70% drop in SNR from the high-quality sensor to the low-quality. The peak SNR profile for the simulation is shown in Figure 8. We see that the algorithm tracks the object with the standard method throughout the first 145 seconds of the simulation. At the 150 second mark the SNR drops from approximately 7.7 to 2.4. After this drop, the standard method fails to make any detections and the matched filter process takes over entirely for the rest of the simulation.

The localization performance for this scenario is seen in Figure 9. It is apparent that there is some increase in the variance of our estimates with the matched filter process, due in part to the low SNR of the signal. The average error increases from approximately 1 pixel in the first 150 seconds to 1.5 pixels in the last 150 seconds. The important thing to note here is that, despite the reduction in precision, we are able to make measurements of an object that would not otherwise be possible. This potentially allows us to free up low-noise sensors for higher-priority targets, while lower priority ones are passed off to a lower-fidelity sensor.

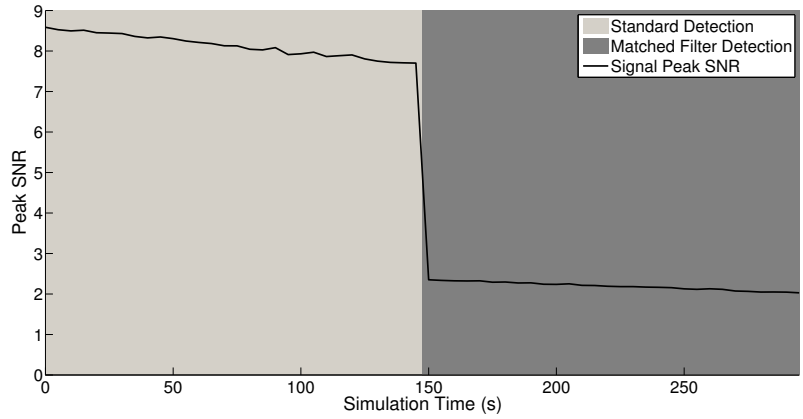

Figure 8. RSO signal-to-noise ratio and detection method.

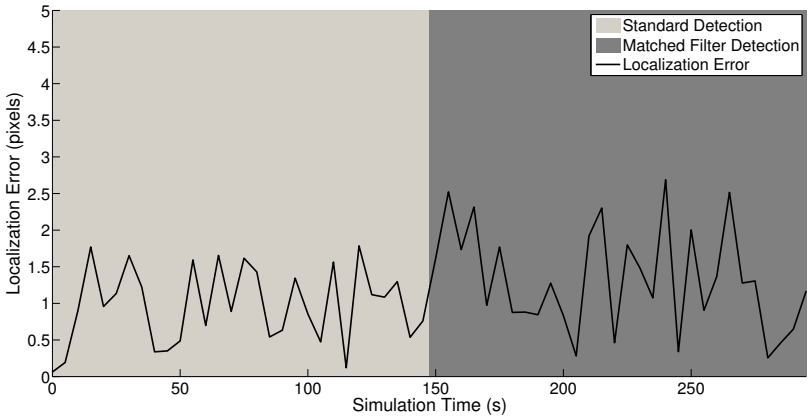

Figure 9. Pixel localization error and detection method.

4. CONCLUSIONS

In this paper we have presented a robust method for processing optical imagery for SSA. The algorithm is broken down into two components: a linearized point tracking method and an on-line orbit determination routine. The first component is capable of tracking sufficiently bright objects frame-to-frame. These measurements are reported to the orbit determination routine, which then generates an estimate of the orbital parameters. Once some orbital knowledge has been generated, it is possible to leverage that knowledge in order to produce a more sensitive detector. With this knowledge we can predict what the object will look like in successive frames and invoke a matched filter process, enabling detection of objects at a much lower SNR.

We have shown that this algorithm is robust to spontaneous signal corruption, dynamically-varying visible magnitudes, and rapid decreases in SNR. In each of these cases the matched filter process supplemented the standard procedure, filling in detections where the standard method failed. The algorithm provided an increased number of detections for all cases. Thus we have shown that we can benefit from processing data and providing real-time feedback between tracking and orbit determination processes at the sensor level.

This work immediately motivates incorporation of closed loop feedback into the RSO tracking problem. With on-line orbit determination, sensor-level tracking can be mtaintained over long periods without new measurements. Complex hand-offs are another avenue of future work. With on-line orbit determination, passing off tracks to geometrically diverse observers can easily be explored. Another topic of interest is passing off matched filter templates to geometrically diverse sensors. There are a number of interesting topics in matched filtering

such as the exploration of methodologies for using a matched filter with a large prior distribution. Modeling, estimating, and exploiting attitude dynamics should enable a more sophisticated matched filter.

REFERENCES

1. Donald Rumsfeld, D Andrews, R Davis, H Estes, R Fogleman, J Garner, W Graham, C Horner, D Jeremiah, T Moorman, et al. Report of the commission to assess united states national security space management and organization. Technical report, 2001.
2. T. S. Kelso. Celestrak. Available: http://www.celestrak.com/SpaceTrack/.
3. Robert S Caprari. Generalized matched filters and univariate neyman-pearson detectors for image target detection. *Information Theory, IEEE Transactions on*, 46(5):1932–1937, 2000.
4. Timothy S. Murphy, Brien Flewelling, and Marcus J Holzinger. Particle and matched filtering using admissible regions. In *AAS/AIAA Space Flight Mechanics Meeting*, 2015.
5. C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.
6. Peter Kovesi. Phase congruency detects corners and edges. *The australian pattern recognition society conference: DICTA 2003*, 2003.
7. D. Mortari, C. Bruccoleri, S. La Rosa, and J. L. Junkins. CCD data processing improvements. In *International Conference on Dynamics and Control of Systems and Structures in Space*, 2002.
8. J. Crassidis and J. Junkins. *Optimal Estimation of Dynamic Systems*. CRC Press, 2nd edition, 2012.
9. M. A. C. Perryman et al. The HIPPARCOS Catalogue. *Astronomy and Astrophysics*, 323:L49–L52, July 1997.
10. Samuel S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, (January), 2004.
11. D. Mortari, F. Markley, and P. Singla. An optimal linear attitude estimator. *Journal of Guidance, Control, and Dynamics*, 30(6):1619–1627, 2007.
12. G. Wahba. Problem 65—1: A least squares estimate of spacecraft attitude. *SIAM Review*, 7(3):409, July 1965.
13. Kumar Vishwajeet, Puneet Singla, and Moriba Jah. Nonlinear uncertainty propagation for perturbed two-body orbits. *Journal of Guidance, Control, and Dynamics*, pages 1–11, 2014.
14. Kyle J DeMars and Moriba K Jah. Probabilistic initial orbit determination using gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 36(5):1324–1335, 2013.
15. Kyle J DeMars, Moriba K Jah, and Paul W Schumacher. Initial orbit determination using short-arc angle and angle rate data. *Aerospace and Electronic Systems, IEEE Transactions on*, 48(3):2628–2637, 2012.
16. Kohei Fujimoto and Daniel J Scheeres. Applications of the admissible region to space-based observations. *Advances in Space Research*, 52(4):696–704, 2013.
17. Johnny L Worthy III and Marcus J Holzinger. Incorporating uncertainty in admissible regions for uncorrelated detections. In *AIAA/AAS Astrodynamics Specialist Conference*, 2014.
18. Giacomo Tommei, Andrea Milani, and Alessandro Rossi. Orbit determination of space debris: admissible regions. *Celestial Mechanics and Dynamical Astronomy*, 97(4):289–304, 2007.
19. Davide Farnocchia, Giacomo Tommei, Andrea Milani, and Alessandro Rossi. Innovative methods of correlation and orbit determination for space debris. *Celestial Mechanics and Dynamical Astronomy*, 107(1-2):169–185, 2010.
20. Andrea Milani, Giovanni F Gronchi, Mattia de'Michieli Vitturi, and Zoran Knežević. Orbit determination with very short arcs. i admissible regions. *Celestial Mechanics and Dynamical Astronomy*, 90(1-2):57–85, 2004.
21. Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric Wan. The unscented particle filter. In *NIPS*, pages 584–590, 2000.
22. Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

23. D. North. An analysis of the factors which determine signal/noise discrimination in pulsed-carrier systems. *Proceedings of the IEEE*, 51(7):1016–1027, 1963.
24. IS Reed, RM Gagliardi, and HM Shao. Application of three-dimensional filtering to moving target detection. *Aerospace and Electronic Systems, IEEE Transactions on*, (6):898–905, 1983.
25. GEORGE L TURINT. An introduction to matched filters. 1960.
26. Matthew Dragovic. *Velocity filtering for target detection and track initiation*. PhD thesis, 2003.
27. John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.
28. E. Høg et al. The Tycho-2 catalogue of the 2.5 million brightest stars. *Astronomy and Astrophysics*, 355:L27, 2000.