AAS 16-266

HARDWARE-IN-THE-LOOP COMPARISON OF SPACE OBJECT DETECTION AND TRACKING METHODOLOGIES

Jared Lee[*], Lubna Zubair[†], Shahzad Virani[‡], Timothy Murphy[§], Marcus J. Holzinger[¶]

Space Domain Awareness relies on a network of surveillance architectures to catalog passive and active space object data. This paper analyzes the performance of several detection and tracking algorithms using different hardware-in-the-loop test configurations. Several image sources are tested, including simulated images generated with the Hipparcos and Space Object catalogs, and real images taken from a Nocturn XL camera. The Moving Target Indicator (MTI), Multiple Hypothesis Tracking, and Finite Set Statistics algorithms are tested, implemented, and compared in this study.

INTRODUCTION

Space Domain Awareness (SDA) is the ability to detect, track, and categorize space objects [1]. Though the current catalog houses over 20,000 objects, the database is far from complete and the number of space objects in orbit is increasing. New objects need to be detected, and known objects in the catalog need to be maintained and updated as they drift due to orbital perturbations [2][3]. SDA shifts the focus of Space Situational Awareness (SSA) methods to obtain data using a network of collaborators from various surveillance architectures. To successfully integrate the data, SDA relies on sensor development, algorithm development, network development, tasker development, and data sharing [4]. Together, these allow the tracking and characterization of passive and active space objects [2].

Space object location and orbit parameters can be tracked using radar or optical methods [4]. This paper focuses on ground-based and space-based Electro-Optical sensors to detect and track resident space objects (RSOs): active satellites and space debris. Image processing algorithms analyze the images to classify objects as stars, objects of interest (in our case RSOs), or artifacts (hot or dead pixels). The objects take the form of either a point or a streak depending on the type of sensor, exposure time, and relative angular velocities of the objects [5]. After an object is detected, the orbital parameters can be calculated, most classically with the method of Gauss using three observations of the same object at different moments in time [6].

Detect-before-track methods threshold images and test detected objects and corresponding positions with velocity matching methods [7]. Dim objects can be extracted from an image through comparisons with brighter stars. For example, the average range of photons received by a CCD sensor for objects and stars is 20-50 and 1000-4000 respectively. Mohanty applies a maximum likelihood algorithm to estimate the changing mean and covariance of the assumed Gaussian background to detect RSOs [8]. Object shape, either characterized for each individual object, or by applying a filter to seek a specific shape, can be used on individual frames to identify stars or RSO streaks. Using a star catalog, a mask can be applied to a single frame leaving dimmer stars and potential RSOs behind [5]. Levesque applies a match filtering technique to

[*]Undergraduate Student, The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332
[†]Undergraduate Student, The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332
[‡]Graduate Student, Space Systems Design Lab, The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332
[§]Graduate Student, Space Systems Design Lab, The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332
[¶]Assistant Professor, The Guggenheim School of Aerospace Engineering, Atlanta, GA 30332

identify streaks during a re-acquisition of data where the image astrometry is known. In this situation, the sensor accesses an RSO database housing a description of the objects' mean orbital parameters to retrieve the RSOs angular rates and direction. Once the stars in an image are eliminated, the match filter is applied to identify streaks and obtain the RSOs exact location [9].

Though detect-before-track methods are generally computationally efficient, they do not perform well in dimly lit conditions [3]. Alternatively, Track-Before-Detect Methods evaluate multiple frames and uses the raw measurements to simultaneously detect objects and evaluate their motions [3]. A Binary Hypothesis Tracking (BHT) Method compares two positions of an object between images: if the object is in a pixel or not. Three successive images can be used to extract the the track of the object and predict the object's next position [10]. The Multiple Hypothesis Tracking (MHT) Method works similarly by assessing the probability of the object's position [11]. Richards presents a similar method using Hough transformations with the dimensions of range and azimuth to compare the returns of the object's track dynamics. Using the track-before-detect method, the frames can be integrated using a binary or a summing method [11]. Other methods include using stereo imaging and 3-D coordinates to further eliminate false positives [6].

The U.S. Space Surveillance Network (SSN) hosts the largest SSA catalog[6]. Ground-based systems such as telescopes and radar observations have the capability of detecting and tracking space objects. Ground-based optical systems that use visible light spectrum are limited by atmospheric conditions and night operations [12]. Objects in LEO provide a challenge for ground based systems, as the objects have a greater angular rate unlike objects in GEO which have a slower angular rate and remain in the same area of the sky for a longer duration [6]. Raven-class telescope refer to a design using commercial off the shelf hardware and software components to complete a mission [13]. MIT Lincoln Laboratory (MITLL) and DARPA developed the Space Surveillance Telescope (SST) with a customized CCD array and 3.5 m f/1.0 telescope to search and detect space debris in GEO*.

On-orbit sensors are able to operate at all times of the day and are not dependent on weather conditions. The U.S. Air Force launched the Space Based Space Surveillance (SBSS) Satellite in September 2010 and two Geosynchronous SSA Program (GSSAP) spacecraft in July 2014 to increase surveillance capabilities [14]. SBSS operates from a 630km orbit above Earth 24 hours, 7 days a week detecting objects as small as 1 m^3 in GEO†. GSSAP are at a higher orbit at 35,970 km above Earth and have rendezvous and proximity operation capabilities to maneuver around RSOs ‡.

The paper presents tests intended for implementation on a Cubesat architecture, but the algorithms can be applied to space-based and ground-based architectures. A comparison of multiple Electro-Optical detection and tracking methods are presented. The algorithms are tested using real data taken from a Nocturn XL camera on flight-like hardware.

BACKGROUND THEORY

Object Detection in Images

To detect and track RSOs, detections must first be extracted from imaging data provided by the camera then transformed to reference frames useful for later processing stages. This process is typically referred to as feature detection and is a classic problem in computer vision with many solutions of varying degrees of complexity [15]. The OpenCV computer vision library [16] provides in its `SimpleBlobDetector` class an implementation of Lindeberg's watershed-based grey-level blob detection algorithm [17], described below.

Let the intensity distribution of an image be described by some discrete mapping $I(x, y)$ whose maxima indicate the presence of stars or RSOs. Thresholding the image at some intensity I_t yields a binary image whose non-zero regions are assumed to be features of interest. Taking the contour of the image outlines any non-zero regions, the area centroids of which approximate the locations of any objects in the image. The

*`http://www.au.af.mil/au/awc/awcgate/usspc-fs/space.htm`, last accessed Oct. 5 2015

†"Space Based Space Surveillance (SBSS) Factsheet," February 2015.

‡"Geosynchronous Space Situational Awareness Program (GSSAP) Factsheet," April 2015.

threshold-contour-centroid sequence is applied multiple times at thresholds greater than I_t. The centroids are then averaged to approximate the mass centroid of the nonzero regions, resulting in more accurate detections.

The chosen value of I_t is critical to the performance of the system. By prescribing a desired signal-to-noise threshold ratio $\text{SNR}_t = I_t/\sigma_n$, one can select a thresholding value based on the standard deviation of the background noise σ_n in a typical dark frame image.

Once centroids have been calculated, they must be transformed from the image plane, measured relative to the top left corner of an image, to the sensor frame, measured relative to the boresight of the camera. This is done using Equation (1).

$$^s\begin{bmatrix} x \\ y_d \\ z_d \end{bmatrix} = \begin{bmatrix} f \\ N_x/2 - x_d^c \\ N_y/2 - y_d^c \end{bmatrix} \tag{1}$$

Here, f is the effective focal length of the lens and N_x and N_y are the width and height of the image, respectively. The subscript d indicates that the centroids extracted from the image are distorted as a result of the light passing through the lens before reaching the sensor. The problem of lens distortion is discussed in the following section.

Distortion Correction

Due to the nature of optics and the limitations of lens design and manufacturing, light passing through the lens onto the sensor is distorted, displacing the centroids of detections in the resulting image, ultimately producing erroneous vectors. The Brown distortion model [18], given in Equation (2), relates distorted coordinates $(\cdot)_d$ to undistorted coordinates $(\cdot)_u$, using four distortion parameters, K_1, K_2, P_1, P_2, the two former for radial distortion and the two latter for tangential distortion.

$$\begin{bmatrix} y_d \\ z_d \end{bmatrix} = \begin{bmatrix} y_u \\ z_u \end{bmatrix} + \begin{bmatrix} y_u r^2 & z_u r^4 & 2y_u z_u & r^2 + 2y_u^2 \\ z_u r^2 & z_u r^4 & r^2 + 2z_u^2 & 2y_u z_u \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \\ P_1 \\ P_2 \end{bmatrix} \tag{2}$$

$$r^2 = y_u^2 + z_u^2 \tag{3}$$

Since Equation (2) is not invertible in y_u and z_u, the Newton-Raphson method is used to calculate the undistorted coordinates. Letting $\mathbf{f}(y_u, z_u)$ be the right hand side of Equation (2), define a new function $\mathbf{g}(y_u, z_u) = \mathbf{f}(y_u, z_u) - [y_d, z_d]^T$. The sequence given in Equation (4) quickly converges, yielding the undistorted coordinates, where $\mathbf{J}$ is the Jacobian matrix of $\mathbf{g}$.

$$\begin{bmatrix} y_u \\ z_u \end{bmatrix}_n = \begin{bmatrix} y_u \\ z_u \end{bmatrix}_{n-1} - \mathbf{J}^{-1}\mathbf{g} \tag{4}$$

Camera Calibration

Calculating the undistorted location of detections of course requires knowledge of the effective focal length and distortion parameters of the camera. Consider the angular separation θ of a pair of stars projected onto the celestial sphere. Using the Hipparcos star catalog [19][§] or similar, this angle can be calculated to a high degree of accuracy. Next, consider an image of the same pair of stars taken by a camera. Using Equations (1) and (2), the angular separation θ^s of the pair can be calculated in the sensor frame. The residual of the two angles $\gamma = \theta - \theta^s$ is then a function of θ^s which is in turn a function of the effective focal length and distortion parameters of the camera, $\boldsymbol{\beta} = [f, K_1, K_2, P_1, P_2]$.

[§]Available at `http://tdc-www.harvard.edu/catalogs/hipparcos.html`, last accessed Feb. 3 2016

$$\gamma(\boldsymbol{\beta}) = \theta - \theta^s(\boldsymbol{\beta}) \tag{5}$$

Given a sufficiently large set of star pairs, an approximation of the effective focal length and distortion parameters of the camera can be found by minimizing $\sum_{i=1}^{n} \gamma_i^2$. Equation (5) is nonlinear in $\boldsymbol{\beta}$, so the Gauss-Newton method of least squares is used, as shown in Equation (6), where $\mathbf{J}$ is the Jacobian matrix of $\boldsymbol{\gamma}(\boldsymbol{\beta})$, the vector of residuals computed using Equation (5).

$$\boldsymbol{\beta}_{n+1} = \boldsymbol{\beta}_n - \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\boldsymbol{\gamma} \tag{6}$$

Ideally, once the distortion parameters have been determined, any remaining residuals are due to ignored higher-order terms in the distortion model and measurement error in the feature detection stage of processing. Further, the distribution should be approximately Gaussian with zero mean.

Star Identification and Subtraction

Depending on the sensitivity of the camera and SNR chosen for blob detection thresholding, it is quite possible to have hundreds of detections in a single image, the majority of which will be stars. Identifying any stars present in a frame is useful for a number of reasons. In the context of RSO tracking, two in particular stand out: the ability to easily place detections in a common reference frame, and the ability to remove stars from consideration when tracking. The former is desirable as it allows the kinematic state of a detection to be meaningfully analyzed over a series of frames, while the latter aids in tracking by reducing the total number of objects to be tracked.

Consider the following assumptions: RSOs are free to move according to orbital mechanics; stars are fixed in the inertial reference frame and greatly outnumber RSOs; and, the satellite is fixed at the inertial origin with only rotational degrees of freedom. Under these restrictions, any movement of the satellite in the inertial frame results in pure rotation of stars in the sensor frame.

Next, consider two sets of detections $\mathcal{D}_{k-1}$ and $\mathcal{D}_k$ expressed in the sensor frame, recorded at times t_{k-1} and t_k, during which the camera undergoes some rotation $\mathbf{R}$. For small rotations, there exist subsets $\mathcal{S}_{k-1} \in \mathcal{D}_{k-1}$ and $\mathcal{S}_k \in \mathcal{D}_k$ consisting of stars common to both frames such that $\mathbf{S}_k = \mathbf{R}\mathbf{S}_{k-1}$. When expressed in a common reference frame, detections of stars will align, allowing them to be easily found by taking the angular separation between elements of each set. Those whose separation is near zero after correcting for rotation are likely to be stars and are removed, leaving behind only RSOs and spurious detections for tracking.

Determining the frame-to-frame rotation can be done in a variety of ways depending on available sensors and system sophistication. If only the ability to detect and track is required, rotation rates from on onboard IMU may be sufficient for coarse alignment of frames. For many applications, such as orbit determination or pointing control, knowledge of the boresight attitude is required, thus image-based star identification algorithms should be considered.

Star Identification

Determining the frame-to-frame rotation of the boresight can be done by identifying the stars common to both frames. Given a set of detections expressed in the body frame, the Kolomenkin geometric voting algorithm[¶] is used to pair each with a star listed in the Hipparcos star catalog. Once the association is known, the frame-to-frame rotation can be calculated to a high degree of accuracy.

Catalog Preprocessing The algorithm requires a preprocessing stage to generate a great-circle distance table `DT`. For every star pair in the Hipparcos star catalog, $(\mathbf{s}_i, \mathbf{s}_j) \in \mathcal{S}_H$, the angular separation between the pair is calculated, $\theta = \angle(\mathbf{s}_i, \mathbf{s}_j)$. Pairs whose angular separation is within the angle of view of the camera are stored in a table, as shown in Table (1), where k indicates the k^{th} row of the table.

[¶]M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum. Geometric voting algorithm for star trackers, IEEE Transactions on Aerospace and Electronic Systems, 44(2):441âĂŞ456, April 2008.

θ	i	j
$\vdots$	$\vdots$	$\vdots$
θ_k	i_k	j_k
$\vdots$	$\vdots$	$\vdots$

Table 1. Distance table format.

To improve performance, DT is sorted such that $\theta_{k-1} \leq \theta_k \leq \theta_{k+1}$, allowing the use of a binary search for quicker indexing. Using this construction, any pair of stars measured by the sensor should exist within DT, provided that the pair is contained within the Hipparcos catalog.

Geometric Voting From the feature detection phase, a set of detections expressed in the sensor frame is available. For each pair $(\mathbf{d}_i, \mathbf{d}_j)$, the angular separation θ^s between the two is calculated. The distance table is then searched for all DT[k] such that $|\theta_k - \theta^s| \leq \gamma$, where γ accounts for detection uncertainty and is based on the results of the calibration procedure.

Each star pair $(\mathbf{s}_{k1}, \mathbf{s}_{k2})$ found in this search potentially corresponds to the measured pair $(\mathbf{d}_i, \mathbf{d}_j)$, so a voting table VT is constructed, where the value of VT[i, j] is the number of votes supporting a correspondence between the i^{th} detection and the j^{th} catalog star. For the potential correspondence of the pairs $(\mathbf{d}_i, \mathbf{d}_j)$ to $(\mathbf{s}_{k1}, \mathbf{s}_{k2})$, two combinations of pairings are possible, so VT is incremented as shown in Table (2).

VT	$\mathbf{s}_{k1}$	$\mathbf{s}_{k2}$
$\mathbf{d}_i$	+1	+1
$\mathbf{d}_j$	+1	+1

Table 2. Voting procedure for a potential correspondence.

Once the voting procedure is complete, each row i of VT is searched to find the column j with the most votes. This is considered to be the pairing of $\mathbf{d}_i$ with $\mathbf{s}_j$ with the highest likelihood of being correct. To detect and remove any false pairings, a second round of voting occurs in which angles are taken between all pairs of detections and between the stars they have been matched with. If the angles agree within $\pm\gamma$, the pairing is considered true. Otherwise, the pair is no longer considered.

Point Alignment Assuming at least three correct pairings of detections to stars is known, the attitude of the satellite can be calculated by finding the rotation that minimizes distances between each detection and its corresponding star.

Let $\mathbf{D}$ be the matrix containing all detections and $\mathbf{S}$ be the matrix containing all stars such that each column in the former corresponds to the same in the latter. The attitude of the satellite is then given by the rotation matrix $\mathbf{R}$ such that Equation (7) is true.

$$\mathbf{S} = \mathbf{R}\mathbf{D} \tag{7}$$

$\mathbf{R}$ can be found by taking the singular value decomposition of the covariance matrix $\mathbf{H}$ of $\mathbf{D}$ and $\mathbf{S}$.

$$\mathbf{H} = \mathbf{D}\mathbf{S}^T \tag{8}$$

Next, the singular value decomposition of $\mathbf{H}$ is computed.

$$\text{svd}(\mathbf{H}) = [\mathbf{U}, \mathbf{s}, \mathbf{V}^T] \tag{9}$$

Finally, $\mathbf{R}$ is computed by taking the product of $\mathbf{U}$ and $\mathbf{V}$.

$$\mathbf{R} = \mathbf{V}\mathbf{U}^T \tag{10}$$

Knowing $\mathbf{R}$, all measurements in the body frame can also be expressed in the inertial frame via a simple rotation by $\mathbf{R}$. If $\mathbf{R}$ is known for two frames, detections in each can be expressed in the inertial frame, aligning stars common to each, allowing them to be subtracted as discussed before.

Data Association Methods

Given two sets of detections, there exists a mapping between some, all, or none of the detections in the first set to those in the second, shown graphically in Figure (1). Finding this mapping is another classical problem in computer vision, typically referred to as the data association or correspondence problem, also with many solutions of varying degrees of complexity.

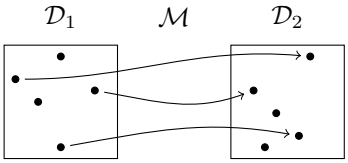

Figure 1. Associations between detections across frames.

Methods of association can often be placed into one of two categories: immediate or deferred assignment. In the former, associations are made on a per frame basis. That is, given two sets of detections $\mathcal{D}_1$ and $\mathcal{D}_2$, a single mapping $\mathcal{M} : \mathcal{D}_1 \rightarrow D_2$ is formed and assumed to be true. In the latter, a set of mappings $\mathcal{M} = [\mathcal{M}_1, \mathcal{M}_2, \cdots, \mathcal{M}_n]$ is generated from which a single mapping is selected after taking into account a specified number of prior frames.

Moving Target Indicator Tracking

Moving target indicator (MTI) tracking methods use nearest neighbor criteria to associate detections in sequential frames [20]. Star detections are first aligned such that each frame is expressed in a common reference frame. Stars are then subtracted from the frame in question. Any remaining detections cannot be inertially fixed and are thus candidates for tracking. MTI typically falls into the immediate assignment category and has extensive flight heritage.

Nearest neighbor association is among the simplest of data association methods. It relies on the assumption that detection sets $\mathcal{D}_1$ and $\mathcal{D}_2$ are taken in rapid enough succession that corresponding detections form small angles relative to the spatial density of detections in the sets. In its simplest form, nearest neighbor associates detections in sequential frames by finding pairs of detections across frames whose distance is a minimum. In the context of RSO tracking, distance is best defined as the angular separation or great circle distance between two detections. Nearest neighbor pairings are immediately assumed to be correct and no further processing of either element is necessary. For trivial applications involving very few simultaneous and non-intersecting tracks, this approach is usually sufficient.

In non-ideal scenarios where detections cross paths or associations are incorrect, nearest neighbor will often produce nonsensical and kinematically impossible tracks, sometimes multiple detections assigned to a single track. Cost minimization algorithms can sometimes be used to mitigate the incorrect pairing problem [21]. Such methods are often computationally expensive and may not be suitable in applications.

Multiple Hypothesis Tracking

Multiple hypothesis tracking (MHT) attempts to solve the multiple assignment problem by deferring the assignment until enough information is available to make a decision with certainty. Over a series of i frames, MHT uses an Extended Kalman Filter (EKF) to generate a tree of hypotheses, each with an associated certainty based off the estimated state and covariance measurements for each measurement and track. For each frame, MHT is initiated by considering three cases for each measurement in a frame: 1. The observation starts a new track; 2. The observation updates an existing track; 3. The observation is following a skipped observation for an existing track [22]. The presented implementation of MHT recomputes the hypotheses with each new frame, maintaining only the tracks and scores from frame $i - 1$.

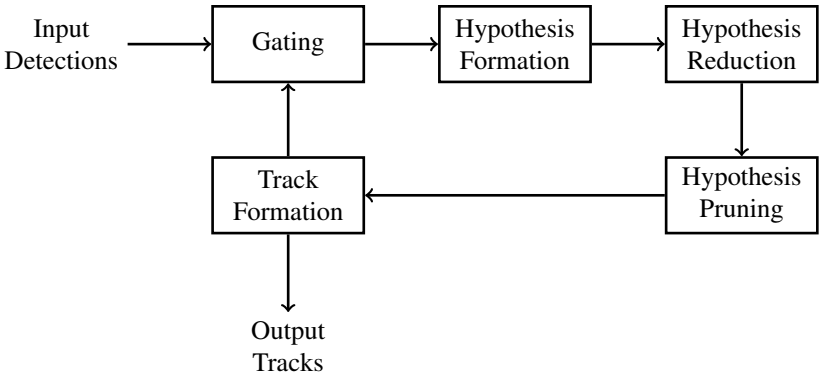

Figure 2. Track Oriented MHT Logic Diagram

Gating The EKF is used to provide the probability of the updated state of a track based on its dynamics and the new measurement. The prediction covariance $\mathbf{P}_k^i$ outputted from the EKF is used to calculate the residual covariance $\mathbf{S}_k^i$ defined in Equation (11) where $\mathbf{H}$ is the linearized measurement matrix and $\mathbf{R}$ is the measurement noise covariance.

$$\mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R} \tag{11}$$

A hypothesis is formed after gating a new observation with each track using the Mahalanobis distance, calculated using $\mathbf{S}$ and $\mathbf{r}_k$ to only consider observations within a predetermined gating threshold, where $\mathbf{r}_k$ is the residual vector defined by Equation (12) and $h(\mathbf{x})$ is the nonlinear observation function.

$$\mathbf{r}_k = \mathbf{z}_k - \mathbf{h}(\mathbf{x}_i|\mathbf{x}_{i-1}) \tag{12}$$

Hypothesis Formation In addition to potentially appending measurements in a current frame to the current tracks, additional hypotheses are made for the scenario of a track skipping an observation in the frame, and for a current measurement beginning an independent track. For a skipped measurement in frame $i+1$, an incremented dt value is calculated and inputted into the EKF along with $\mathbf{x}^i$ to provide a state estimate for frame $i+2$. The formation step also controls the number of skipped frames allowed, further reducing the number of hypotheses created.

Reduction Once the hypotheses are formed, the track score and observation probabilities are calculated. In the track oriented MHT approach, only the track score from the frame $k-1$ is required to be saved. The track score is calculated using the log likelihood ratio (LLR), Equation (13), which allows for easy computation and conversion to the target's true probability [23].

$$\text{LLR} = L(i-1) + \Delta L(i) \tag{13}$$

Assuming a Gaussian distribution, Equation (14), ΔL is used to calculate the object probability based on the probability of detection P_D, the density of false alarms β_{FA}, and $\mathbf{S}$ for a new observation [24]. This value is used to update the track scores for each hypothesis and is used during hypothesis reduction as well as track formation. Each LLR (track score) is weighted based on the number of measurements as the total score is a sum of each individual object's probability score.

$$\Delta L(i) = \begin{cases} \ln(1-P_D) & \text{No track update on frame } i \\ \ln\left(\dfrac{P_D}{2\pi\beta_{FA}|\mathbf{S}|^{1/2}}\right) & \text{Track update on frame } i \end{cases} \tag{14}$$

After gating, the hypotheses are further reduced if the track score falls underneath a predetermined track score threshold. The reduction threshold must be set to allow the hypotheses containing new measurements to survive for at least a few frames, as these hypotheses are generally the lowest scoring.

Track Formation After all the hypotheses are formed they are sorted in descending order of LLR score, and the algorithm selects the most probable track for each measurement based on i frames. There are two main methods of implementing the track formation step. The first is to use Reid's Algorithm and select the $m-$best hypotheses and merge the tracks if necessary [23]. The second method, which is implemented in this paper is to continue selecting tracks in order of probability, adding and merging tracks as they are compatible. Both implementations assign measurements to a current track, or identify a new track. This step also maintains the tracks, deleting any tracks which do not have a desired number of consecutive measurements.

Pruning After track formation, the hypotheses are pruned by deleting any hypotheses which conflict with the newly identified tracks. Figure (3) demonstrates how on frame i, the decision for a detection $\mathcal{D}_n$ or skip in frame $i-2$ is selected based on the track's probability score in frame i. Any other hypotheses in frame $i-2$ are pruned from track 1 and deleted from the iteration in the next frame [22]. Detection 1, $\mathcal{D}_1$ in frame i-2 is also pruned from all the other tracks to maintain track compatibility.

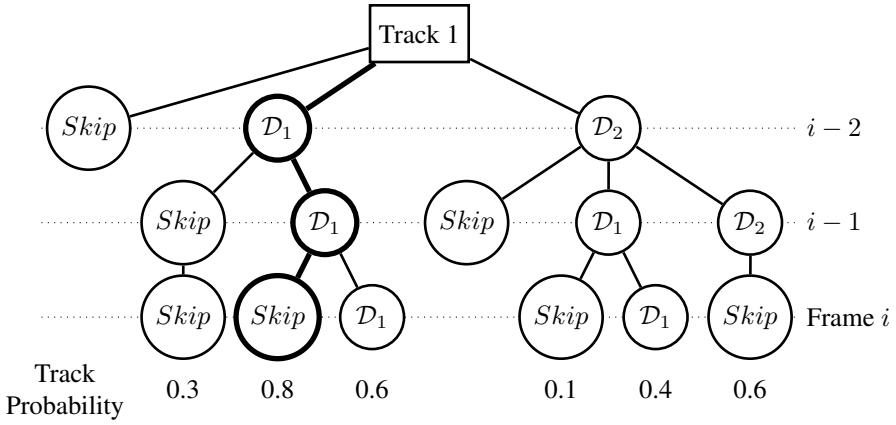

Figure 3. Example of Pruning for k=3

A track is defined as compatible if no two tracks share the same measurement. For example, if `0` represents a skipped observation, `T1 = [1 4 3 2 0]` and `T2 = [1 4 3 2 2]` can be merged into `T3 = [1 4 3 2 2]`, as no observations conflict in any given scan for both tracks. The pruning and merging process minimizes the number of hypotheses that are passed along to frame $i+1$ [22].

Output Tracks The resulting tracks are outputted in a frame ordered list of the pixel coordinates originally inputted into the algorithm (O_x, O_y), estimated states outputted from the EKF x, , covariance matrices $\mathbf{P}$, and object probabilities p.

$$\text{Outputs} = \{O_i, x_i, \mathbf{P}_i, p_i\}$$

Probability Hypothesis Density Filter Approach

Multi-target tracking is a class of dynamic state estimation in which the object of interest is a set that is random in the number of elements as well as the values of individual elements. The representation of these multi-target states can be achieved using finite random sets. Although rigorous algorithms have been developed in the form of Finite Set Statistics, optimal Bayesian multi-target tracking is not yet practical [25]. One of the alternatives is to use a Probability Hypothesis Density (PHD) filter which propagates the PHD of the full multi-target posterior.

Vo et al. proposed a Sequential Monte Carlo (SMC) method to approximate the PHD using a large set of weighted random particles. A probabilistic interpretation of the PHD recursion is implemented that is sufficient to handle the non-linear Gaussian dynamics, unlike the EKF. The detailed derivations of the following equations are presented in [26].

Given a set of states, X_{k-1} at time, t_{k-1}, each $x_{k-1} \in X_{k-1}$ moves to the next state x_k with a transition probability density $f_{k|k-1}(x_k|\xi)$ and continues to exist at time k with a probability of survival $e_{k,k-1}(x_{k-1})$. Alternatively, the particle can disappear with a probability of $1 - e_{k,k-1}(x_{k-1})$ [26].

At time k, the probability of a given target being detected is quantified as $p_{D,k}(x_k)$. Each x_k also generates an observation $z_k \in Z_k$ with a likelihood of $g_k(z_k|x_k)$ [27].

The first step in the algorithm is the prediction step. $D_{k|k-1}$ and $D_{k|k}$ denote the PHD associated with the multi-target prior and posterior density. The PHD associated with the prior is calculated using the following equation:

$$D_{k|k-1} = \int \phi_{k|k-1}(x,\xi) D_{k-1|k-1}(\xi) d\xi + \gamma_k(x) \tag{15}$$

where

$$\phi_{k|k-1}(x,\xi) = e_{k,k-1}(x_{k-1}) f_{k|k-1}(x_k|\xi) + b_{k|k-1}(x|\xi) \tag{16}$$

In the above equation, $b_{k|k-1}(x|\xi)$ denotes the PHD of the RFS spawned by a target with a previous state ξ, and $\gamma_k(x)$ denotes the PHD of the spontaneous birth RFS.

In the update step, the PHD is updated using the following equation:

$$D_{k|k} = \left[\nu(x) + \sum_{z \in Z_k} \frac{\psi_{k,z}(x)}{\kappa_k(z) + C_k(z)} \right] D_{k|k-1}(x) \tag{17}$$

where

$$\nu(x) = 1 - p_D(x) \tag{18}$$

$$\psi_{k,z}(x) = p_D(x) g_k(z|x) \tag{19}$$

$$C_k(z) = \sum_{j=1}^{L_{k-1}+J_k} \psi_{k,z}(x_k^{(j)}) w_{k|k-1}^{(j)} \tag{20}$$

In the above equations, κ_k represents the clutter intensity and $g_k(\cdot|\cdot)$ represents the likelihood of individual targets. Once the PHD of the particles are updated, the particles are resampled based on their weights. New particles are initiated around the particles with highest weights.

IMPLEMENTATION ON FLIGHT-LIKE HARDWARE

Multiple Target Indicator

The input to the MTI tracking algorithm is list of RSO detections and noise remaining after the star subtraction phase of processing. To find the frame-to-frame association, nearest neighbors are found by measuring the angular separation between pairs of detections across frames.

It is desirable for a multitude of reasons to track these correspondences over a series of images. To this end, a list of tracks is carried over the set of images being processed. In each new frame, the remaining

detections are first assumed to be extensions of a currently existing track. To which track each detection belongs is determined via a nearest neighbor association between the detections of the current frame and the most recent detection in each track. If the separation of the nearest neighbors is less than a given distance, the association is made and the RSO is appended to the track. If an RSO has no nearest neighbor within the specified distance, it becomes the beginning of a new track. For the image sets considered in this paper, a the limiting separation was set to twenty times the instantaneous field of view of the camera.

Under this scheme, if detection or subtraction perform sub-optimally, excessive track generation may occur, causing the list of active tracks to grow prohibitively large. After the association stage, each track has either been updated or not. Those that have not been updated for three images are considered inactive or dead and are written to a file then removed from the active tracks list. Each track written to the output file will contain the position of the detection for each frame in which it was detected.

Multiple Hypothesis Tracking

The datasets passed through the MHT algorithm not only include pixels of RSOs, but also image noise pixels identified during object detection and stars which were not subtracted. In order to improve computation time and efficiency, measurements are discarded if the state outputted from the EKF is above a predetermined maximum velocity, indicating a measurement is too far away in the image to be considered inside of the track, or below a minimum velocity indicating a measurement is possibly stationary noise or a star which fell through the object detection phase.

The values for P_D and β_{FA} were obtained experimentally by analyzing MHT's performance on the real datasets. A P_D of 0.99 and β_{FA} of 0.01 allowed the algorithm to detect the track best. These values also tuned the object and track probability thresholds used for track reduction. In order to allow potentially new observations, the object probability was set to a maximum of $log(1 - P_D)$, representing the ΔL score for an unupdated track.

The hypotheses are maintained between frames as arrays, carrying the measurements from a window of $i - skipthreshold$ to $i + (k - 1)$ to minimize the computation time and memory space. The window size provides enough information about the hypothesis measurements to check for compatibility and merging with the declared tracks. Each hypothesis array also has an associated $\mathbf{P}_k^i$ matrix used to updated the state using the EKF. Track information regarding the number of skips, track length, frame origin, and track score are maintained to formulate hypotheses.

To further reduce computation and memory space, skips were only considered if a track was declared (had 3 measurements in consecutive frames), and tracks were discarded if it encountered 4 skipped frames.

The Extended Kalman Filter used the following process noise $\mathbf{R}$, measurement noise $\mathbf{Q}$, and initial uncertainty covariance matrices $\mathbf{P}_0$. On the datasets, N = 10 provided the best results for the amount of noise present in the images.

$$\mathbf{R} = \begin{bmatrix} N & 0 \\ 0 & N \end{bmatrix} \tag{21}$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{v}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\theta}}^2 \end{bmatrix} \tag{22}$$

where $\sigma_{\dot{v}}^2$ was determined by $\left(\frac{N \cdot \text{IFOV}}{dt^2}\right)^2$ and $\sigma_{\dot{\theta}}^2$ by $\left(\frac{N \cdot \text{IFOV}}{dt}\right)^2$.

$$\mathbf{P}_0 = \begin{bmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & 0 & N \end{bmatrix} \tag{23}$$

EXPERIMENTAL SETUP AND SCENARIO DESCRIPTION

Overview

Several image datasets were captured in Atlanta, Georgia using a 35mm lens attached to a Nocturn XL CMOS camera. These datasets each contain one or more visible objects in low earth orbit, including several SL-16 rocket bodies and the SAC-D and COSMO-SkyMed satellites.

Each dataset was first underwent by an object detection to extract the centroids of any visible objects. For each image set, several detection sets were generated by varying the blob-detection threshold. This paper is concerned primarily with the speed of various tracking methodologies, so the detection portion was conducted on a personal computer. For completeness, timing data was recorded and normalized to approximate the expected runtime of the Raspberry Pi, however these results will not be discussed in detail.

The resulting detection sets were transferred to the Raspberry Pi then given as inputs to the star subtraction algorithm, the output of which was then passed on to each of the three tracking methodologies under consideration. The per-image runtime of each dataset for each tracking method was measured and the output tracking data was evaluated qualitatively.

Hardware Description

Processors Each tracking methodology will be tested on Raspberry Pi 2B processor. This particular processor was chosen for its ease of use and similarity to actual space-rated processors, such as the Tyvak Intrepid or Innoflight CFC-300, all of which utilize the ARM architecture and have clock speeds in the 400 MHz to 1 GHz range. Relevant specifications of each are listed in Table (3). Due to these similarities, it is expected that runtimes on the space-rated processors will be within an order of magnitude of the Raspberry Pi.

	Raspberry Pi 2B	Tyvak Intrepid	Innoflight CFC-300
Architecture	ARMv7	ARM926	ARMv7
Processor	Cortex-A7 Quad-Core	AT91SAM9G20	Cortex-A9 Dual-Core
Clock Speed	900 MHz	400 MHz	1 Ghz
SDRAM	1 GB	128 MB	256 MB

Table 3. Processor specifications

Camera The images of the night sky were taken using the Nocturn XL Low-Light CMOS camera. The imaging sensor has a rolling shutter, image resolution of 1280x1024 pixels with 9.7 μm pixel pitch, and a 10-bit video output with adjustable frame rate of 50, 60, or 100 Hz. The sensor read noise is $<$4e- median at 60 Hz and has an SNR of 42 dB. Attached to the sensor was a 35 mm f/1.7 lens, for a field of view of approximately 20.1 x 16.2 degrees and an instantaneous field of view (IFOV) of approximately 57 arcseconds.

Detection thresholds were determined based on dark frame and environmental noise present in each dataset. Figure (4) shows the intensity distribution of a typical image from one of the datasets. Although the camera outputs a 10 bit video feed, the drivers used to store individual frames were unable to correctly read the data stream, resulting in a loss of bit depth in the resulting image. In future work, this problem will be corrected.

Due to the wide field of view of the lens, a non-negligible amount of distortion is present in the images. Further, in order to express image centroids in the body frame, an accurate measure of the effective focal length of the camera in pixel units is necessary. This can be approximated by taking the ratio of the lens focal length to sensor pixel size, but experience has shown that value can sometimes be off by 1-2% as the focus and f-stop are adjusted.

To correct for these, the camera is calibrated by taking a series of images of various parts of the sky then identifying the imaged stars using tools and data provided by Astrometry.net [28]. The calibration

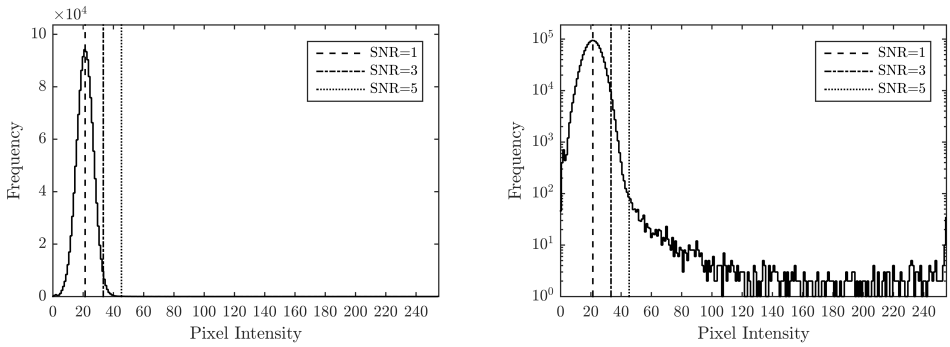

Figure 4. Pixel intensity histogram in linear and log scale.

procedure described earlier is then applied. Figure (5) shows the change in residual distribution before and after calibration is applied to a subset of one the datasets.

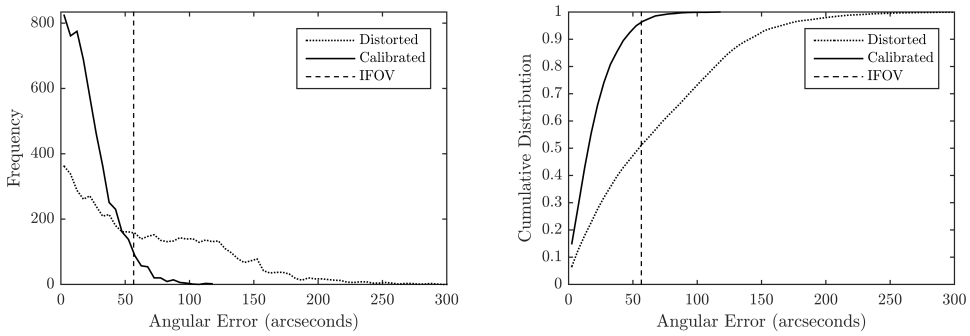

Figure 5. Residuals of the angular separation between star catalog and sensor-frame vectors before and after calibration.

The subset contained a total of six images in which approximately 40 stars were identified per image. Approximately 5500 unique star pairs were taken into consideration and convergence was reached in under 10 iterations of the Gauss-Newton method. After calibration, approximately 95% residuals lie within the sensor IFOV, indicating that most detections now fall within a one pixel radius of their expected position.

Table 4. Calibration results with 95% confidence interval.

Parameter	Value	Confidence
f	3646.4310	$\pm$ 3.2e-3
K_1	1.37535e-1	$\pm$ 8.2e-5
K_2	-4.126e-1	$\pm$ 1.7e-3
P_1	1.01285e-3	$\pm$ 8.4e-7
P_2	-1.68341e-3	$\pm$ 6.5e-7

Scenario Description

Three image sets were tested, each consisting of 100 or more images taken by the Nocturn camera at a frame rate of approximately 2.5 images per second. The image sets were taking in Georgia, USA at

different dates and regions of the sky. The SAC-D and SL-16 016B datasets each had one RSO detectable by human eye after dark frame subtraction. The SL-16 093B dataset had two RSOs moving across the frame in approximately the same direction. A tumbling rocket body is also recognizable in the top left corner of the frame, with fluctuating brightness levels.

RESULTS AND DISCUSSION

MTI Results

Multiple target indicator tracking results are given in Figures (6) and (7). Figure (6) shows the runtime performance with respect to the chosen thresholding signal to noise ratio. As SNR threshold value is increased, few spurious detections occur due to less background noise passing the threshold filter. As a result, fewer detections remain after the subtraction phase, meaning fewer nearest neighbor comparisons per frame thus shorter runtimes.

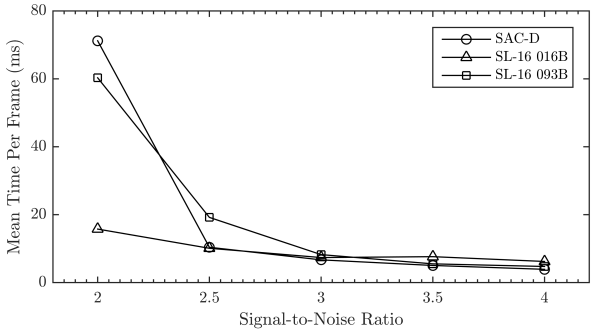

Figure 6. MTI tracking time per frame with respect to SNR thresholding value.

Figure (7) shows the per frame runtime with respect to the total number of detections remaining after the subtraction stage of processing. As expected, poor performance of the subtraction phase causes poor run time performance. Further, because nearest neighbor association is quadratic in its limiting behavior, greater numbers of spurious detections causes an approximately quadratic increase in computation time.

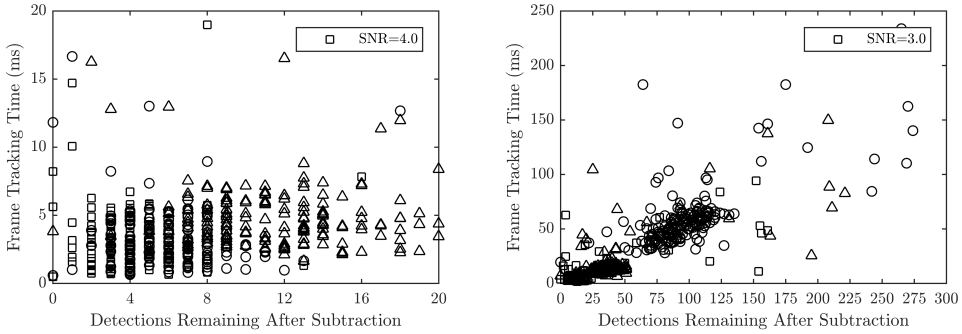

Figure 7. MTI tracking time with respect to detections remaining after subtraction.

Figures (8) and (9) show the results of running the MTI algorithm on the SL-16 093B dataset. Triangles and squares are used to symbolize the beginning and end of a track, respectively. Hexagons are used to symbolize detected stars along with their respectively catalog number. As expected, the results for an SNR 2.0 yielded more detected tracks than with the SNR 4.0 dataset.

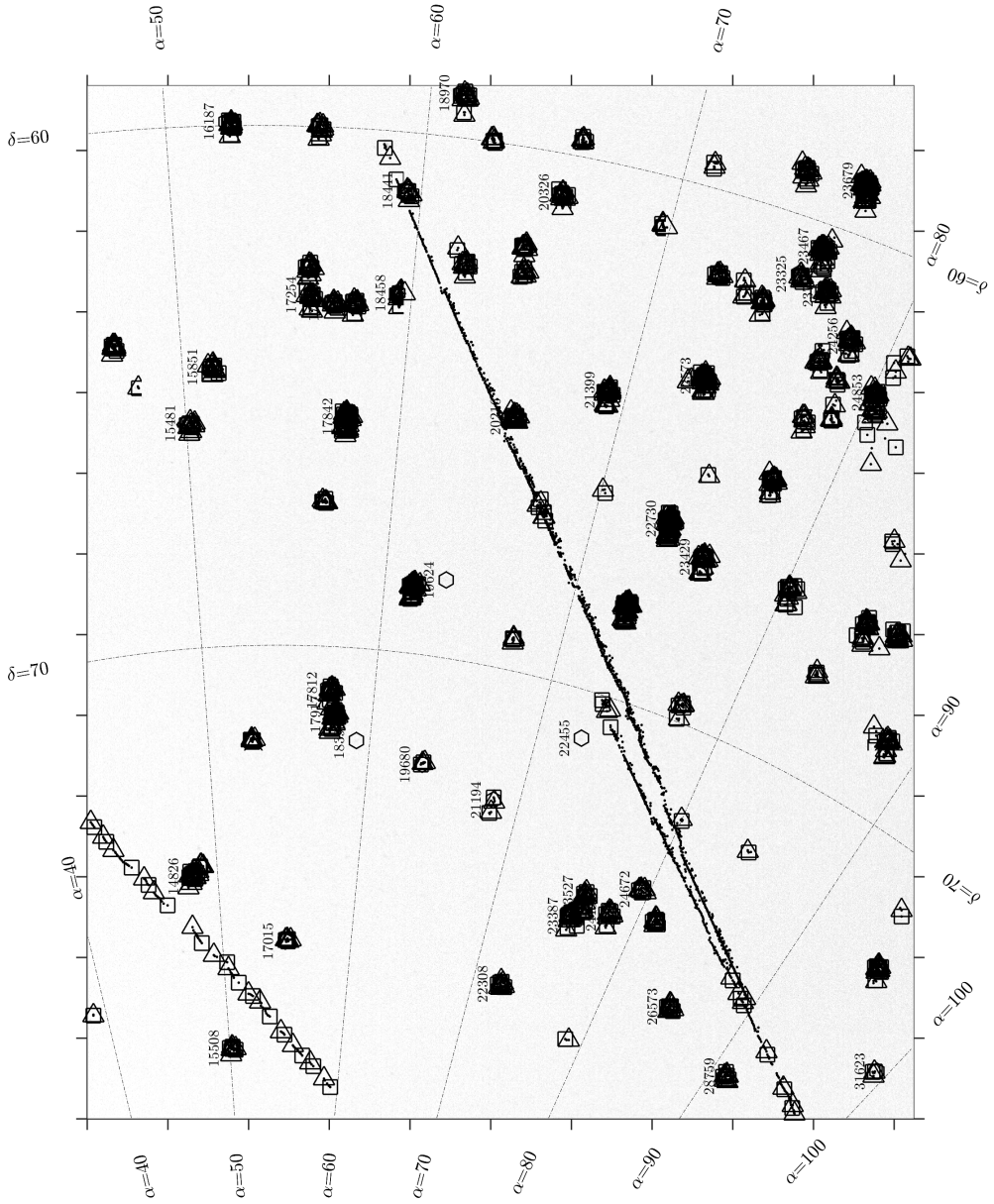

Figure 8. MTI: SL-16 016B SNR 2.0

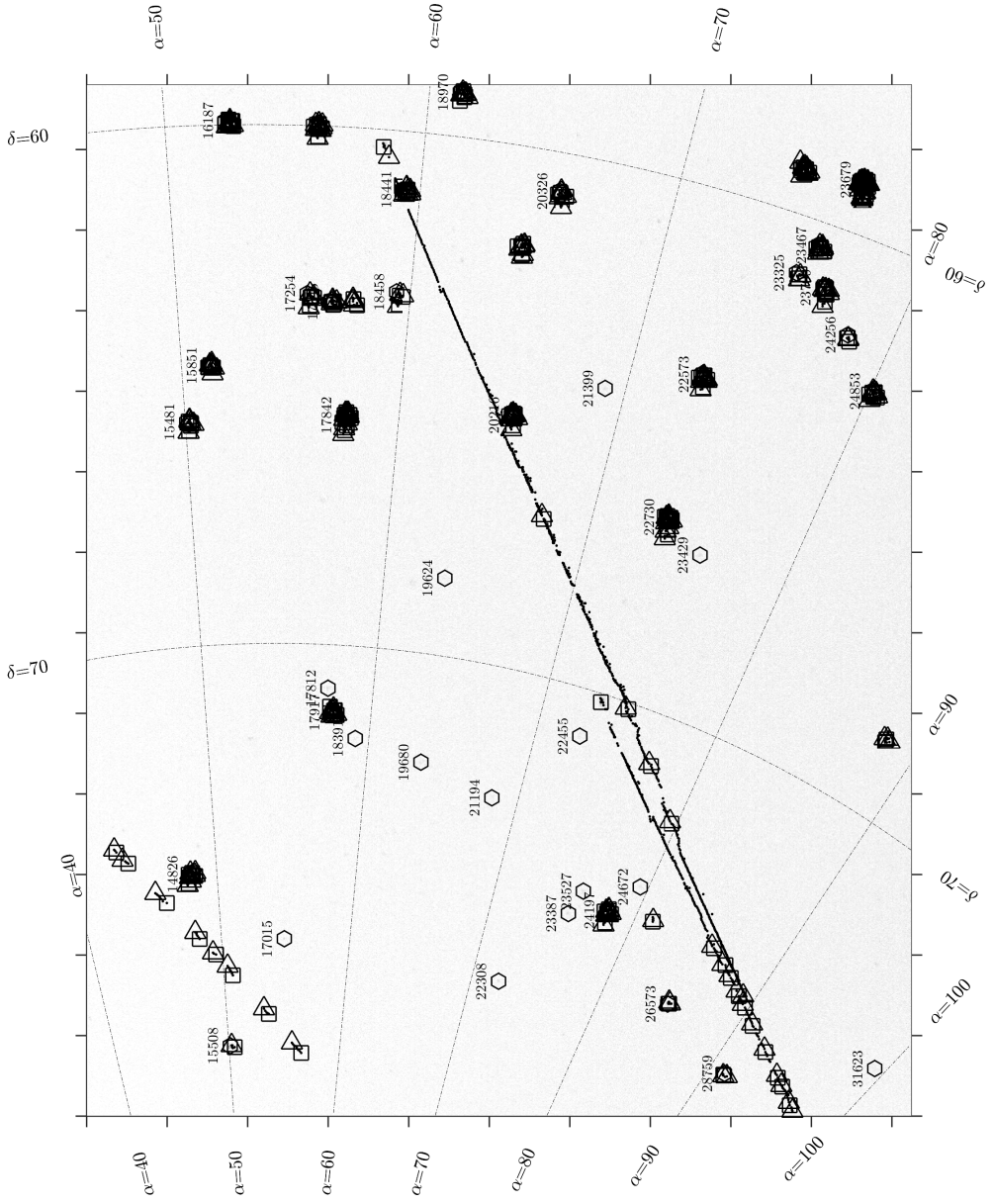

Figure 9. MTI: SL-16 016B SNR 4.0

MHT Results

The computational time for MHT was considerably larger than the MTI results, as expected. The SNR 2.0 datasets took on average 4-6 seconds between frames, but produced higher quality track data than the MTI results, yielding less tracks due to stars and noise. Figure (10) , shows how the computational time decreases rapidly as the SNR increases, producing results with less than 1 second per frame. MHT did not detect as many stars and random noise as MTI, because the tracks were selected based on the probability of the the predicted states propagated by the EKF.

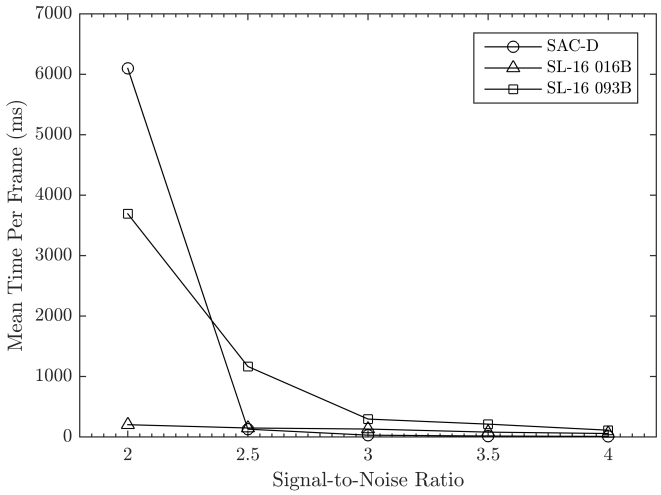

Figure 10. MHT tracking time per frame with respect to SNR thresholding value

Figure (11) shows an example of how the error covariance decreases with each consecutive frame in a track from the SAC-D dataset. At frame $i = 5$, the track is initiated with pixel coordinates, therefore no associated covariance is assigned, and from frame $i = 6$ to $i = 11$ the track's probability increases from 73.7% to 99.8%.

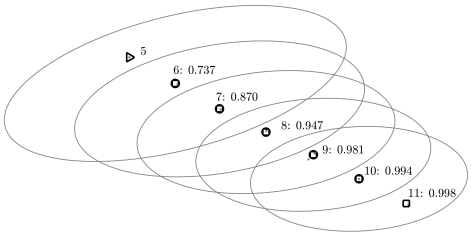

Figure 11. Track Probability vs. Frame

Figures (12) and (13) show the results of running the MHT algorithm on the SL-16 093B dataset. The pruning process of the algorithm allows for a higher higher skip threshold, resulting in longer track lengths when compared to the MTI results. The results displayed were calculated with a skip allowance threshold of 10 frames. In Figure (12), the tumbling rocket body in the top left corner is tracked for a longer amount of frames than the SNR 4.0 dataset due to the higher frequency of detections. The MHT algorithm was also able to account of the skipped frames and associate more of the rocket body detections to a single track.

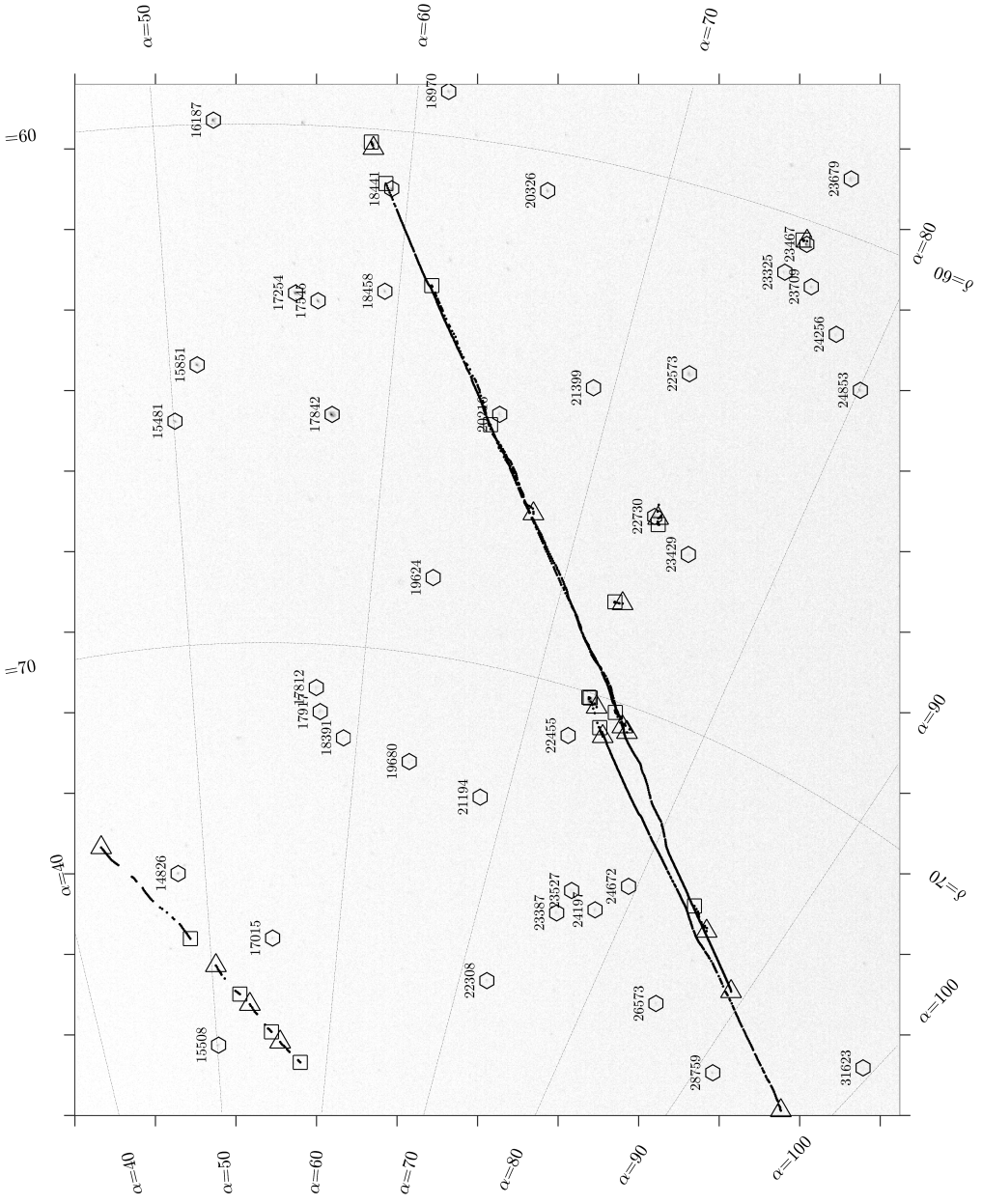

Figure 12. MHT: SL-16 016B SNR 2.0

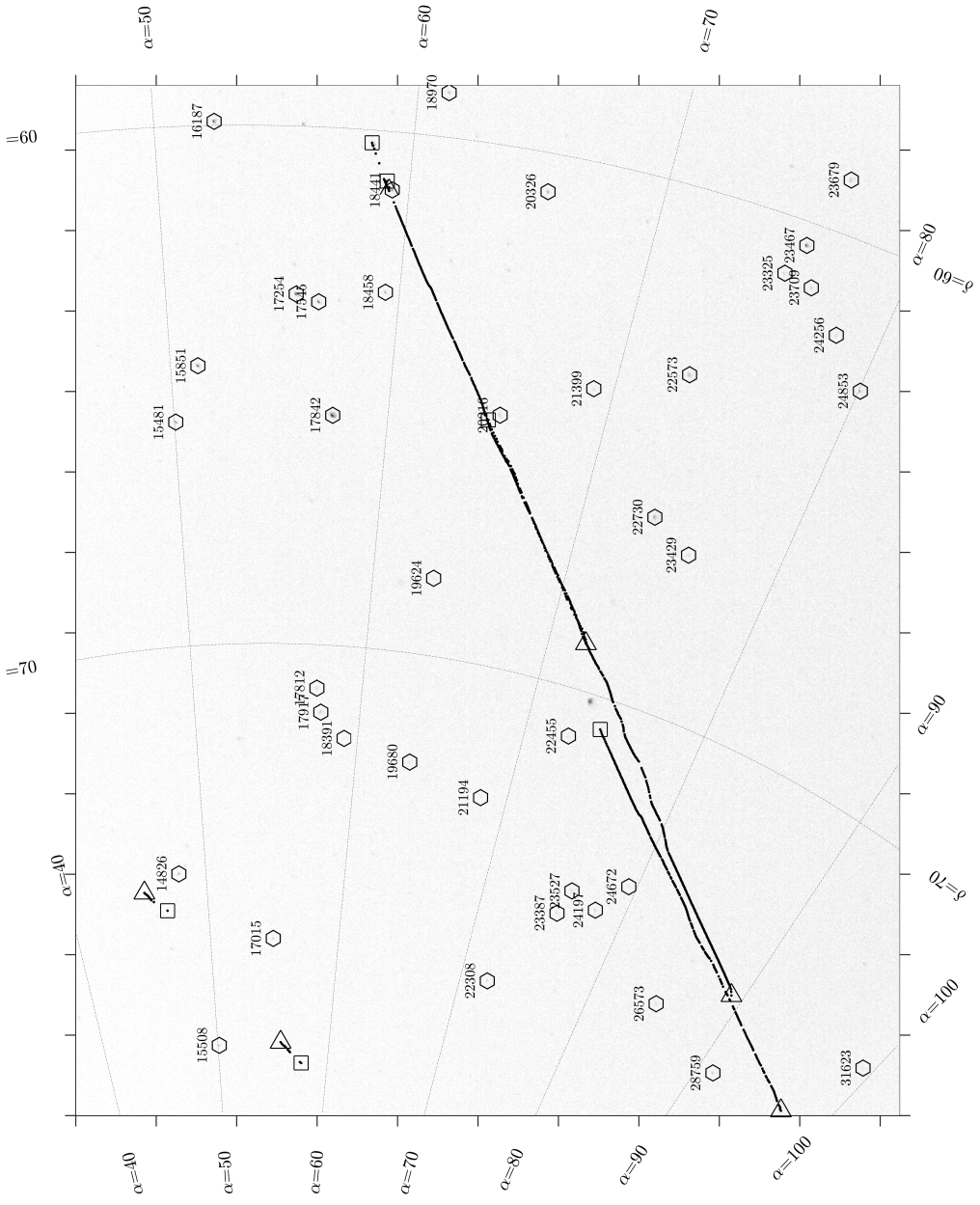

Figure 13. MHT: SL-16 016B SNR 4.0

Algorithm Benchmarking

Table 5. Algorithm Performance

		Average Runtime per Frame (ms)		95% Confidence Interval (ms)	
Scenario	SNR	MTI	MHT	MTI	MHT
SAC-D	2.0	71.2	6100	8.28	834
	3.0	6.68	29	0.72	5.43
	4.0	3.90	8.6	0.34	2.35
SL-16 016B	2.0	10.1	202	0.51	30.3
	3.0	7.37	130	1.38	48.9
	4.0	7.62	57	0.45	7.32
SL-16 093B	2.0	60.3	3700	4.66	1900
	3.0	8.23	300	1.88	262
	4.0	4.74	110	1.01	90.5

The MTI algorithms performed faster than the MHT results as expected. Both algorithms had a significantly higher run time for the SAC-D and SL-16 093B SNR 2.0 datasets which had frames with over 100 detections after star subtraction. All other datasets were able to be processed with < 1 sec per frame. This performance is favorable for real time mission implementation which is expected to have a frame rate of about 2 fps.

Like other particle filters, the runtime of current SMC implementation of PHD Filter depends on the number of particles chosen. Due to the high resolution of the images, 50,000 particles were used which takes significant amount of processing time. Additionally, unlike MHT, the current FISST approach does not provide track information for the objects. Hence, data association between frames was performed after the filter generated the PHD for all objects. Therefore, it was concluded that the current implementation of FISST approach is not suitable for real-time detection and tracking of RSOs with the chosen hardware.

FUTURE WORK

To improve the performance of the algorithms, the false detections need to be minimized. This will be accomplished by improving the object detection and star tracking/subtraction algorithms. To further improve the performance, the algorithms will be made to skip analyzing frames that have significantly large number of detections in order to keep the processing time between frames consistent, and feasible to run in a real-time application. To better compare the processing times between MHT and FISST algorithms, alternate implementations of FISST, such as Adaptive Entropy-based Gaussian-mixture Information Synthesis (AEGIS-FISST) and Gaussian Mixture PHD Filters, will be used [29] [30]. These algorithms will also be benchmarked on the Innoflight CFC-300 processor to provide a comparison with the flight hardware used for the RECONSO cubesat mission.

REFERENCES

[1] T. J. Hardy and S. C. Cain, "Characterizing Point Spread Function (PSF) fluctuations to improve Resident Space Object detection (RSO)," Vol. 9469, 2015.

[2] T. Blake, M. Goergen, M. Sanchez, S. Sundbeck, and J. Krassner, "DARPA Space Domain Awareness," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Sept. 2012, p. 39.

[3] M. Fanaswala and V. Krishnamurthy, "Syntactic Models for Trajectory Constrained Track-Before-Detect," *IEEE Transactions on Signal Processing*, Vol. 62, Dec 2014, pp. 6130–6142, 10.1109/TSP.2014.2360142.

[4] A. Ciurte, A. Soucup, and R. Danescu, "Generic method for real-time satellite detection using optical acquisition systems," *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sept 2014, pp. 179–185, 10.1109/ICCP.2014.6936971.

[5] T. S. Edith Stoveken, "Algorithms for the optical detection of space debris objects.," *Proceedings of 4th European Conference on Space Debris*, 2005.

[6] T. V. A. Danescu R, Ciurte A, "A low cost automatic detection and ranging system for space surveillance in the medium Earth orbit region and beyond.," , 10.3390/s140202703

[7] P. S. Gural, J. A. Larsen, and A. E. Gleason, "Matched Filter Processing for Asteroid Detection," *The Astronomical Journal*, Vol. 130, No. 4, 2005, p. 1951.

[8] N. C. Mohanty, "Computer Tracking of Moving Point Targets in Space," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 3, May 1981, pp. 606–611, 10.1109/TPAMI.1981.4767153.

[9] M. P. Levesque, "Automatic Reacquisition of Satellite Positions by Detecting Their Expected Streak in Astronomical Images," tech. rep., Defence R&D Canada, 2009.

[10] J. C. Zingarelli, E. Pearce, R. Lambour, T. Blake, C. J. R. Peterson, and S. Cain, "Improving the Space Surveillance Telescope's Performance Using Multi-Hypothesis Testing," *The Astronomical Journal*, Vol. 147, No. 5, 2014, p. 111.

[11] G. Richards, "Application of the Hough transform as a track-before-detect method," *IEE Colloquium on Target Tracking and Data Fusion*, Nov 1996, pp. 2/1–2/3, 10.1049/ic:19961349.

[12] M. Mallick, S. Rubin, and B.-N. Vo, "An introduction to force and measurement modeling for space object tracking," *16th International Conference on Information Fusion (FUSION)*, July 2013, pp. 1013–1020.

[13] R. D. Coder and M. J. Holzinger, "Autonomy Architecture for a Raven-Class Telescope with Space Situational Awareness Applications," 2013.

[14] J. Rendleman and S. Mountin, "Responsible SSA cooperation to mitigate on-orbit space debris risks," *7th International Conference on Recent Advances in Space Technologies (RAST)*, June 2015, pp. 851–856, 10.1109/RAST.2015.7208459.

[15] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufman, third ed., 2005.

[16] G. Bradski *Dr. Dobb's Journal of Software Tools*, 2000.

[17] T. Lindeberg and J.-O. Eklundh, "Scale detection and region extraction from a scale-space primal sketch," *Third International Conference on Computer Vision Proceedings*, Dec 1990, pp. 416–426, 10.1109/ICCV.1990.139563.

[18] D. C. Brown, "Decentering Distortion of Lenses," *Photometric Engineering*, Vol. 32, No. 3, 1966, pp. 444–462.

[19] M. Perryman, *The Making of History's Greatest Star Map*. Springer Berlin, Heidelberg, 2010.

[20] D. F. Kostishack, B. E. Burke, and G. J. Mayer, "Continuous-Scan Charge-Coupled Device (CCD) Sensor System With Moving Target Indicator (MTI) For Satellite Surveillance," 1980, 10.1117/12.959482.

[21] H. W. Kuhn, "The Hungarian Method for the assignment problem," *Naval Research Logistics Quarterly*, Vol. 2, 1955, pp. 83–97.

[22] A. Amditis, G. Thomaidis, P. Maroudis, P. Lytrivis, and G. Karaseitanidis, "Multiple Hypothesis Tracking Implementation," *Laser Scanner Technology*, InTech, March 2012, 10.5772/33583.

[23] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, Vol. 19, Jan 2004, pp. 5–18, 10.1109/MAES.2004.1263228.

[24] R. D. Palkki, A. Lanterman, and W. Blair, "Addressing Track Hypothesis Coalescence in Sequential K -Best Multiple Hypothesis Tracking," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 47, July 2011, pp. 1551–1563, 10.1109/TAES.2011.5937249.

[25] R. Mahler, "Statistics 102 for Multisource-Multitarget Detection and Tracking," *Selected Topics in Signal Processing, IEEE Journal of*, Vol. 7, June 2013, pp. 376–389, 10.1109/JSTSP.2013.2253084.

[26] B. N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 41, Oct 2005, pp. 1224–1245, 10.1109/TAES.2005.1561884.

[27] T. Li, S. Sun, M. Bolic, and J. M. Corchado, "Algorithm design for parallel implementation of the SMC-PHD filter," *Signal Processing*, Vol. 119, 2016, pp. 115 – 127, 10.1016/j.sigpro.2015.07.013.

[28] J. Barron, C. Stumm, D. W. Hogg, D. Lang, and S. Roweis, "Cleaning the Usno-B Catalog Through Automatic Detection of Optical Artifacts," Vol. 135, Jan. 2008, pp. 414–422, 10.1088/0004-6256/135/1/414.

[29] R. E. e. a. Hussein, Islam, "Stochastic Optimization for Sensor Allocation Using AEGIS-FISST," *24th AAS/AIAA Space Flight Mechanics Meeting, Santa Fe, NM*, Jan 2014.

[30] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, Vol. 54, No. 11, 2006, pp. 4091–4104.