Real-time Multi-target Detection & Tracking of Space Objects using FiSSt Methods

Shahzad Virani [*], **Dr. Marcus Holzinger** [†]
University of Colorado - Boulder

ABSTRACT

This paper leverages the likelihood ratio for low SNR problem to motivate the use of target likelihood ratio recursion for detecting and tracking dim targets. A new intensity marginalized likelihood ratio is proposed, which allows for quick detection of low SNR targets in the entire surveillance region. A constant velocity model for targets for short period of time is used to estimate the probability of existence using the intensity marginalized likelihood ratio. Any target tracks that have a probability of existence greater than a chosen threshold are handed-off to image-based Multi Bernoulli filter to account for process noise and for track maintenance. The tracks are handed off after computing the intensity based on the maximum likelihood estimation. A closed form solution for this intensity estimate is also presented in the paper. This method is compared to the well-known synthetic tracking approach for detecting low SNR asteroids. Several cases are shown to compare and contrast the performance of both the methods. Additionally, results from the proposed method are shown on simulated data of multiple low SNR targets. A follow-on paper will discuss the exact implementation details for computing the intensity marginalized likelihood function in real-time as well as present results on real data from various sensors.

1. INTRODUCTION

Space Situational Awareness (SSA) and Space Traffic Management (STM) activities involve the ability to detect, track, and characterize space objects. The current space object catalog contains over 20,000 objects and the database is far from complete. Additionally, the number of space objects are increasing at an alarming rate. Recent developments in space have emphasized the need to detect and track low SNR targets. With new objects being launched into Cislunar space, it it crucial for the current sensors to be able to search and detect targets in very noisy background with poor lighting conditions. Various methods have been proposed to detect low SNR signals such as synthetic tracking / shift & add methods [1], multi-object tracking [2], and matched filter based detection [3], to name a few.

Traditional approaches for target tracking are based on detections from the data as measurements for state estimation. These detections are often produced by thresholding the output from a sensor. These detect-before-track (DBT) methods typically perform well for targets that have high signal-to-noise ratio (SNR) and often significantly reduce computation time by restricting the amount of data needed to be processed. For a target with a given SNR, the probability of detection and the density of false alarm depends on the chosen detection threshold. For low SNR targets, the threshold must be sufficiently low to allow for target detection. However, choosing a low threshold leads to a higher rate of false detections. This increase in false alarm rate in turn affects the complexity of the data association and thereby increases computation time.

In order to lower the computational effort, thresholding techniques simply throw away useful information for low SNR targets. Hence, to track low SNR targets, an alternative approach known as track-before-detect (TBD) method is used, which involves utilizing all of the sensor data with the tracker without any threshold, thereby retaining as much information as possible. However, in these methods, the measurements are highly nonlinear functions of the target state. In addition to estimating the target state, the TBD algorithm also needs to be able to detect the presence and absence of the targets. Several papers have proposed a particle-based track-before-detect filters for tracking single targets [4], [5], [6]. Since the measurement likelihood

[*]Ph.D. Student, NDSEG Fellow, Colorado Center for Astrodynamics Research (CCAR). Email: shez.virani@colorado.edu
[†]Associate Professor, Colorado Center for Astrodynamics Research (CCAR). Email: marcus.holzinger@colorado.edu

depends on the target intensity, which is typically unknown, the intensity is also estimated in addition to the position and velocity. Salmond & Birch augment the state space with a target existence variable, which is modeled as a two-state Markov process. This existence variable is then used to compute the probability of target existence. Alternatively, Rutten et. al separate the probability of existence calculation from state estimation in order to efficiently compute the target existence probability using partice weights [7]. Note that these methods were only considered single target tracking.

An alternative approach to track multiple low SNR targets is the H-PMHT method proposed by Streit et. al [8]. This approach uses a parametric representation of the target pdf to reduce computation load. An expectation-maximization (EM) method is applied to derive a stable tracking algorithm that uses a histogram interpretation of the intensity data with probabilistic multiple hypothesis tracking (PMHT). These methods have also been extended to use phase information to improve the TBD algorithm [9]. Recently, Finite Set Statistics (FiSSt) based filters have been proposed as an alternative for multi-target tracking. In this methodology, the collection of target states are modeled as finite sets and the posterior distribution of the random finite set of states is computed given the sensor data. Vo et. al leveraged the FiSSt framework to develop a joint detection and tracking algorithm using Multi-Bernoulli filters [10], [11].

This paper proposes a new intensity marginalized likelihood function, which when used along with target likelihood ratio recursion, allows for fast detection of low SNR targets within the surveillance region. This method is compared to a simple case of synthetic tracking/shift-and-add methods to illustrate the differences in the underlying principles between the two methods. Additionally, the targets detected above a probability of existence threshold based on a constant velocity assumption are then handed off to a image-based Multi-Bernoulli filter to maintain the track. The tracks are handed off by computing the maximum likelihood estimate for the target intensity along with an estimate of target's kinematic states.

2. THEORY

This section begins with describing the dynamics and measurement model of targets in low SNR problem pertaining to electro-optical images. Then, the measurement likelihood function is shown before any Bayesian theory is discussed. This is used to motivate the target likelihood ratio based recursion. Additionally, an intensity marginalized likelihood ratio is derived along with maximum likelihood estimate for intensity.

2.1 Dynamics Model

This paper will discuss joint detection and estimation of low SNR targets from electro-optical measurements. Consider a target of an unknown brightness/intensity moving in the $x-y$ plane. The dynamics of the target can be modeled in discrete time as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{1}$$

where k denotes the time index, $\boldsymbol{v}_k$ is the discrete-time process noise, $\boldsymbol{x}_k \in \mathbb{R}^5$ is the target state vector defined as follows

$$\boldsymbol{x}_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k & I_k \end{bmatrix}^{\mathrm{T}} \tag{2}$$

Since we are interested in electro-optical data, it is convenient to choose the surveillance region to be defined by the images. Therefore, (x_k, y_k) denote the position of the target in pixel space, $(\dot{x}_k, \dot{y}_k)$ denote the velocity of the target in pixel space, and I_k denotes the target brightness/intensity in pixel counts. Additionally, if the target disappears from the surveillance region, it has a null state represented by ϕ. For convenience, also define the target's kinematic states as

$$\tilde{\boldsymbol{x}}_k = \begin{bmatrix} x_k & y_k & \dot{x}_k & \dot{y}_k \end{bmatrix}^{\mathrm{T}} \tag{3}$$

It is important to note that given the dynamics along with the statistics of the process noise is equivalent to knowing the transition density, $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$. For additional information, see [2].

One key assumption that is made in this paper is that the targets in the electro-optical data appear as two-dimensional circular symmetric Gaussian functions. This is generally the case for when the targets are

rate-tracked or when the targets are moving very slowly relative to the sensor during the exposure time. This is typically the case for rate-tracked targets in the geostationary belt as well as targets in cislunar space relative to ground-based telescopes. The next paper on this topic will extend this work to fast moving targets that appear as streaks.

2.2 Measurement Model

Since we are interested in electro-optical sensors, the data collected are sequences of two-dimensional images of the surveillance region, where each consists of $n \times m$ pixels of size $\Delta_x \times \Delta_y$. Let (i,j) $1 \le i \le n$, $1 \le j \le m$ denote the location of the center of each pixel. Additionally, let $\boldsymbol{y}_k = \left\{ y_k^{(i,j)} : i = 1,\ldots,n,\ j = 1,\ldots,m \right\}$ denote the image observation received at time k and $\boldsymbol{Y}_{1:k} = \{\boldsymbol{y}_b,\ b = 1,\ldots,k\}$ denote the set of measurements received up until time k. The intensity measured in pixel (i,j) can be modeled as follows:

$$y_k^{(i,j)} = \begin{cases} h_k^{(i,j)}(\boldsymbol{x}_k) + w_k^{(i,j)} & \text{if target present} \\ \\ w_k^{(i,j)} & \text{if target absent} \end{cases} \tag{4}$$

where $h_k^{(i,j)}(\boldsymbol{x}_k)$ denotes the intensity in the pixel (i,j) contributed by a target with state $\boldsymbol{x}_k$, $w_k^{(i,j)}$ is the measurement noise in pixel (i,j). In words, the measurement received in a pixel will contain some signal + noise if a target exists, but only noise if no target exists. This paper also assumes independence of measurement noise from pixel to pixel and image to image, just as it is done in the literature. However, it is important to note that the results of this paper hold even if this assumption is not made, as long as the noise statistics are known for each pixel (i,j). For this discussion, the noise is assumed to be Gaussian, i.e. $w_k^{(i,j)} \sim \mathcal{N}\left(0, \sigma^2\right)$. It is also important to note that unresolved imagery of space objects typically contains sky background and bright stars and are assumed to be removed from the image before detecting low SNR targets [12].

In the case of unresolved imagery, a target appears as a point source in the image and the photons from it are dispersed based on the point spread function (PSF) of the imaging system. This PSF is often approximated as a two-dimensional Gaussian function with circular symmetry [4]. To be exact, $h_k^{(i,j)}(\boldsymbol{x}(k))$ should be the integral of the PSF over the pixel (i,j). Therefore, the pixel intensity in (i,j) contributed by a target at position (x_k, y_k) with intensity I_k is

$$h_k^{(i,j)}(\boldsymbol{x}_k) \approx \frac{\Delta_x \Delta_y I_k}{2\pi\Sigma^2} \exp\left[-\frac{(i\Delta_x - x_k)^2 + (j\Delta_y - y_k)^2}{2\Sigma^2} \right] \tag{5}$$

where Σ is the blur factor, which is a function of the optics and seeing conditions and is assumed to be known. Additionally, for convenience, also define the intensity normalized pixel contribution as

$$\begin{aligned} \tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k) &= \frac{\Delta_x \Delta_y}{2\pi\Sigma^2} \exp\left[-\frac{(i\Delta_x - x_k)^2 + (j\Delta_y - y_k)^2}{2\Sigma^2} \right] \\ h_k^{(i,j)}(\boldsymbol{x}_k) &= I_k \cdot \tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k) \end{aligned} \tag{6}$$

Since we asssumed that a target appears as a two-dimensional circularly symmetric Gaussian function, it is evident from Eq. (5) that the pixel intensity contribution is only a function of the target's intensity and its positional states, (x_k, y_k) but not its velocity states. If the targets are moving fast relative to the sensor, then this is no longer true and this case will be addressed in the next follow-on paper.

2.3 Measurement Likelihood Ratio

As we assumed independence of measurement noise from pixel to pixel, the likelihood for the entire image at time t_k can be computed by the product of the individual pixel likelihoods. However, the individual pixel likelihood depends on whether a target exists in its vicinity or not. Hence, using Eq. (4), the measurement likelihood can be written as

$$\begin{aligned} p\left(y_k^{(i,j)} \middle| \boldsymbol{x}_k \right) &= \mathcal{N}\left(y_k^{(i,j)}; h_k^{(i,j)}, \sigma^2 \right) \qquad \text{(if a target is in vicinity of pixel } (i,j)) \\ p\left(y_k^{(i,j)} \middle| \phi_k \right) &= \mathcal{N}\left(y_k^{(i,j)}; 0, \sigma^2 \right) \qquad \text{(if no target is in vicinity of pixel } (i,j)) \end{aligned} \tag{7}$$

If the measurement likelihood is to be computed for a pixel (i,j) that is far from the target's hypothesized position (x_k, y_k), then the target's intensity contribution, i.e. $h_k^{(i,j)}$ will approach zero. Therefore, it is trivial to see that the existence of a target will only affect the measurement likelihood in the pixels that are close to the target's state. Let $\mathcal{C}_x(\boldsymbol{x}_k) = \{j : \|j\Delta_x - x_k\| \leq r\}$ and $\mathcal{C}_y(\boldsymbol{x}_k) = \{i : \|i\Delta_y - y_k\| \leq r\}$ define the vicinity around a target with state $\boldsymbol{x}_k$ of distance r, which is a user defined parameter based on the known PSF of the optics and seeing condition. Therefore, the total likelihood for the entire image at time t_k can be written as:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \prod_{i\in\mathcal{C}_y(\boldsymbol{x}_k)}\prod_{j\in\mathcal{C}_x(\boldsymbol{x}_k)} p\left(y_k^{(i,j)}\middle|\boldsymbol{x}_k\right) \prod_{i\notin\mathcal{C}_y(\boldsymbol{x}_k)}\prod_{j\notin\mathcal{C}_x(\boldsymbol{x}_k)} p\left(y_k^{(i,j)}\middle|\phi_k\right) \tag{8}$$

This equation can be further simplified as follows:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \prod_{i\in\mathcal{C}_y(\boldsymbol{x}_k)}\prod_{j\in\mathcal{C}_x(\boldsymbol{x}_k)} \frac{p\left(y_k^{(i,j)}\middle|\boldsymbol{x}_k\right)}{p\left(y_k^{(i,j)}\middle|\phi_k\right)} \underbrace{\prod_{i=1}^{n}\prod_{j=1}^{m} p\left(y_k^{(i,j)}\middle|\phi_k\right)}_{p(\boldsymbol{y_k}|\phi_k)} \tag{9}$$

In Bayesian framework, the update equation has a normalizing constant in the denominator as shown in Eq. (14). Hence, the measurement likelihood is only required to be known up to a proportionality constant. We can leverage this property and define the first term in the above equation as the pixel-wise measurement likelihood ratio which only depends on the vicinity of the hypothesized target state:

$$l\left(y_k^{(i,j)}\middle|\boldsymbol{x}_k\right) \triangleq \frac{p\left(y_k^{(i,j)}\middle|\boldsymbol{x}_k\right)}{p\left(y_k^{(i,j)}\middle|\phi_k\right)} \tag{10}$$

Therefore, the measurement likelihood up to a proportionality constant is related to the measurement likelihood ratio as follows:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) \propto \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_k)}{p(\boldsymbol{y}_k|\phi_k)} = l(\boldsymbol{y}_k|\boldsymbol{x}_k) = \prod_{i\in\mathcal{C}_y(\boldsymbol{x}_k)}\prod_{j\in\mathcal{C}_x(\boldsymbol{x}_k)} \frac{p\left(y_k^{(i,j)}\middle|\boldsymbol{x}_k\right)}{p\left(y_k^{(i,j)}\middle|\phi_k\right)} \tag{11}$$

Substituting the appropriate pdfs from Eq. (7) and simplifying results in

$$\begin{aligned} l(\boldsymbol{y}_k|\boldsymbol{x}_k) &= \prod_{i\in\mathcal{C}_y(\boldsymbol{x}_k)}\prod_{j\in\mathcal{C}_x(\boldsymbol{x}_k)} \exp\left\{-\frac{1}{2\sigma^2}h_k^{(i,j)}(\boldsymbol{x}_k)\left(h_k^{(i,j)}(\boldsymbol{x}_k) - 2y_k^{(i,j)}\right)\right\} \\ &= \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i\in\mathcal{C}_y(\boldsymbol{x}_k)}\sum_{j\in\mathcal{C}_x(\boldsymbol{x}_k)} h_k^{(i,j)}(\boldsymbol{x}_k)\left(h_k^{(i,j)}(\boldsymbol{x}_k) - 2y_k^{(i,j)}\right)\right]\right\} \end{aligned} \tag{12}$$

This measurement likelihood ratio is convenient since it *only* depends on the pixels in the vicinity of the target state. This property is crucial in extending this method for multi-target tracking under the assumption that no two target PSFs overlap. Additionally, since Bayesian recursion only requires the measurement likelihood to be known up to a proportionality constant, this measurement likelihood ratio allows us to incorporate the data into the filter without having to compute the actual likelihood. This also motivates us to reformulate the Bayesian recursion in terms of the *target likelihood ratio density* as opposed to computing both the posterior target and null pdfs.

2.4 Bayesian Estimation of Low SNR Target

In the Bayesian framework, the goal is to compute the posterior pdf at time $k+1$, i.e. $p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k+1})$, given the posterior at time k, i.e. $p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})$, and the measurement received, $\boldsymbol{y}_{k+1}$. This recursive estimation generally involves using the Chapman-Kolmogorov equation to compute the predicted posterior pdf for a

discrete dynamical system with a known transition density, $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)$, and incorporating the measurements to compute the posterior using Bayes' rule, as shown below:

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \int_{\mathcal{X}} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\ p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k \tag{13}$$

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \frac{p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})}{\int_{\mathcal{X}} p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_{k+1}} \tag{14}$$

2.4.1 Prediction Equation for Target Likelihood Ratio

However, for general single/multi-target tracking, the target can appear into the surveillance region or target state-space, $\mathcal{X}$ and disappear, i.e. the birth and death processes [13]. If the target disappears, its state is not defined and we can denote it using the null state, ϕ, and we augment it to the target state-space: $\mathcal{X} \cup \phi$. Accounting for the null state, the prediction equation (13) becomes

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \int_{\mathcal{X}} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\ p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k\ +\ p(\boldsymbol{x}_{k+1}|\phi_k)\ p(\phi_k|\boldsymbol{Y}_{1:k}) \tag{15}$$

$$p(\phi_{k+1}|\boldsymbol{Y}_{1:k}) = \int_{\mathcal{X}} p(\phi_{k+1}|\boldsymbol{x}_k)\ p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k\ +\ p(\phi_{k+1}|\phi_k)\ p(\phi_k|\boldsymbol{Y}_{1:k}) \tag{16}$$

where $p(\boldsymbol{x}_{k+1}|\phi_k)$ and $p(\phi_{k+1}|\boldsymbol{x}_k)$ represent the birth and death of a target, respectively. If birth and death processes are modeled such that null state probability does not change [13], i.e. the probability mass exchange to and from ϕ and $\mathcal{X}$ is balanced, the predicted null state probability integral simplifies to

$$p(\phi_{k+1}|\boldsymbol{Y}_{1:k}) = p(\phi_k|\boldsymbol{Y}_{1:k}) \tag{17}$$

Instead of keeping track of the target null state pdf, it is more convenient to compute the ratio of the target state pdf to the null state pdf, i.e. let the target likelihood ratio density be defined as

$$\Lambda(\boldsymbol{x}_k|\cdot) = \frac{p(\boldsymbol{x}_k|\cdot)}{p(\phi|\cdot)} \tag{18}$$

It is important to clarify that the predicted likelihood ratio density is denoted as

$$\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})}{p(\phi_{k+1}|\boldsymbol{Y}_{1:k})} \tag{19}$$

and the posterior likelihood ratio density is denoted as

$$\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k+1}) = \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k+1})}{p(\phi_{k+1}|\boldsymbol{Y}_{1:k+1})} \tag{20}$$

Additionally, note that the this target likelihood ratio density is different than the measurement likelihood ratio, which is defined in Eq. (12). Substituting the target likelihood ratio in the prediction equation and rewriting the predicted target likelihood ratio at t_{k+1} in terms of the posterior target likelihood ratio at t_k

results in

$$\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})}{p(\phi_{k+1}|\boldsymbol{Y}_{1:k})} \tag{21}$$

$$= \frac{\int\limits_{\mathcal{X}} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\ p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k + p(\boldsymbol{x}_{k+1}|\phi_k)\ p(\phi_k|\boldsymbol{Y}_{1:k})}{\int\limits_{\mathcal{X}} p(\phi_{k+1}|\boldsymbol{x}_k)\ p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k + p(\phi_{k+1}|\phi_k)\ p(\phi_k|\boldsymbol{Y}_{1:k})} \tag{22}$$

$$= \frac{\int\limits_{\mathcal{X}} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\ p(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k + p(\boldsymbol{x}_{k+1}|\phi_k)\ p(\phi_k|\boldsymbol{Y}_{1:k})}{p(\phi_k|\boldsymbol{Y}_{1:k})} \tag{23}$$

$$\implies \Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \int\limits_{\mathcal{X}} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\ \Lambda(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k + p(\boldsymbol{x}_{k+1}|\phi_k) \tag{24}$$

2.4.2 Update Equation for Target Likelihood Ratio

Accounting for the null state, the Bayesian update equation (14) transforms into the following:

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \frac{p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})}{\int\limits_{\mathcal{X}} p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_{k+1}\ +\ p(\boldsymbol{y}_{k+1}|\phi_{k+1})\ p(\phi_{k+1}|\boldsymbol{Y}_{1:k})} \tag{25}$$

$$p(\phi_{k+1}|\boldsymbol{Y}_{1:k}) = \frac{p(\boldsymbol{y}_{k+1}|\phi_{k+1})\ p(\phi_{k+1}|\boldsymbol{Y}_{1:k})}{\int\limits_{\mathcal{X}} p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_{k+1}\ +\ p(\boldsymbol{y}_{k+1}|\phi_{k+1})\ p(\phi_{k+1}|\boldsymbol{Y}_{1:k})} \tag{26}$$

Note that denominator for both the posterior pdfs is identical. Using the definition for the posterior likelihood ratio density from Eq. (20) and dividing the two equations results in the following:

$$\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k+1}) = \frac{p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k+1})}{p(\phi_{k+1}|\boldsymbol{Y}_{1:k+1})} = \frac{p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ p(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})}{p(\boldsymbol{y}_{k+1}|\phi_{k+1})\ p(\phi_{k+1}|\boldsymbol{Y}_{1:k})} \tag{27}$$

$$\implies \Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k+1}) = l(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ \Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) \tag{28}$$

Therefore, the Bayesian recursion shown in Eqs. (13) and (14) reformulated in terms of target likelihood ratio densities becomes as follows:

$$\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \int\limits_{\mathcal{X}} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\ \Lambda(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k\ +\ p(\boldsymbol{x}_{k+1}|\phi_k) \tag{29}$$

$$\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k+1}) = l(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1})\ \Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) \tag{30}$$

Several sensors have multiple detectors that scan a region of the sky. In these instances, the total measurement likelihood ratio is just the product of the measurement likelihood ratio of the individual images from different detector, after being aligned using the celestial coordinates.

2.5 Intensity Marginalized Likelihood Ratio

An important property to note from Eq. (12) is that the likelihood ratio depends on both the target's position as well as the intensity. Using the intensity as a state in the filter adds another dimension to the states space, which can become computationally intensive. Additionally, it is difficult to jointly detect & track a low SNR target and estimate its brightness, especially for multiple targets in a wide field-of-view. To circumvent this problem, the target intensity can be marginalized, resulting in a likelihood ratio that only depends on the kinematic states of the target. By definition, based on Eq. (3):

$$l\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right) = l\left(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_k, I_k\right) \tag{31}$$

Therefore,

$$l\left(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_k\right) = \int_{\mathcal{I}} l\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right) \cdot p(I_k|\tilde{\boldsymbol{x}}_k)\ dI_k \tag{32}$$

Typically, the target intensity is not modeled a function of the kinematic states. However, one could choose to do so based on the application. The general problem deals with detecting targets between two bounds of SNR, which does not depend on where the target is located in the surveillance region. Hence, the pdf of the intensity can be considered to be independent of the target's kinematic states, i.e.

$$p(I_k|\tilde{\boldsymbol{x}}_k) = p(I_k) \tag{33}$$

The target intensity can also vary significantly between frames, especially for a low SNR object. Instead of choosing to model the intensity as a Markov process, this paper assumes a uniform distribution, i.e.

$$p(I_k) = \begin{cases} \dfrac{1}{I_{\max} - I_{\min}}, & I_{\min} \leq I_k \leq I_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{34}$$

Therefore, computing the integral results in the intensity marginalized likelihood ratio as follows:

$$l\left(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_k\right) = \frac{1}{I_{\max} - I_{\min}} \int_{I_{\min}}^{I_{\max}} \exp\left\{ -\frac{1}{2\sigma^2} \left[\sum_{i \in \mathcal{C}_y(\boldsymbol{x}_k)} \sum_{j \in \mathcal{C}_x(\boldsymbol{x}_k)} h_k^{(i,j)}(\boldsymbol{x}_k) \left(h_k^{(i,j)}(\boldsymbol{x}_k) - 2y_k^{(i,j)} \right) \right] \right\} dI_k \tag{35}$$

For clarity, define:

$$\Omega_1\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}_k\right) = \sum_{i \in \mathcal{C}_y(\tilde{\boldsymbol{x}}_k)} \sum_{j \in \mathcal{C}_x(\tilde{\boldsymbol{x}}_k)} 2y_k^{(i,j)} \cdot \tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k) \tag{36}$$

$$\Omega_2\left(\tilde{\boldsymbol{x}}_k\right) = \sum_{i \in \mathcal{C}_y(\tilde{\boldsymbol{x}}_k)} \sum_{j \in \mathcal{C}_x(\tilde{\boldsymbol{x}}_k)} \left(\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k) \right)^2 \tag{37}$$

$$\Omega_3\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}_k\right) = \frac{\sqrt{\pi\sigma^2}}{(I_{\max} - I_{\min})\sqrt{2\Omega_2\left(\tilde{\boldsymbol{x}}_k\right)}} \exp\left\{ \frac{\left[\Omega_1\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}_k\right)\right]^2}{8\sigma^2\Omega_2\left(\tilde{\boldsymbol{x}}_k\right)} \right\} \tag{38}$$

$$l\left(\boldsymbol{y}_k|\tilde{\boldsymbol{x}}_k\right) = \Omega_3\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}_k\right) \left[\operatorname{erf}\left(\frac{2I_{\max}\Omega_2\left(\tilde{\boldsymbol{x}}_k\right) - \Omega_1\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}_k\right)}{\sqrt{8\sigma^2\Omega_2\left(\tilde{\boldsymbol{x}}_k\right)}} \right) - \operatorname{erf}\left(\frac{2I_{\min}\Omega_2\left(\tilde{\boldsymbol{x}}_k\right) - \Omega_1\left(\tilde{\boldsymbol{x}}_k, \boldsymbol{y}_k\right)}{\sqrt{8\sigma^2\Omega_2\left(\tilde{\boldsymbol{x}}_k\right)}} \right) \right] \tag{39}$$

This equation can then be used to compute the *intensity marginalized likelihood ratio* for an image $\boldsymbol{y}_k$ given a hypothesized target's kinematic state $\tilde{\boldsymbol{x}}_k$. This equation might seem difficult to compute in real-time since it only allows for calculating the measurement likelihood ratio for each hypothesized target state $\tilde{\boldsymbol{x}}_k$. However, careful implementation results in fast real-time computation of this likelihood ratio for the states in the surveillance region. Exact implementation details will be further discussed in another follow-on paper.

2.6 Maximum Likelihood Estimate for Intensity

Since the new likelihood is based on marginalization of the target intensity, it is crucial to compute the intensity that maximizes the likelihood so as to pass the tracks to the MBF. This can easily be done using maximum likelihood estimation (MLE). It is important to note that the maximum likelihood estimate for the intensity is derived from the intensity based likelihood function shown in Eq. (12)

$$l\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right) = \exp\left\{ -\frac{1}{2\sigma^2} \left[\sum_{i \in \mathcal{C}_y(\boldsymbol{x}_k)} \sum_{j \in \mathcal{C}_x(\boldsymbol{x}_k)} h_k^{(i,j)}(\boldsymbol{x}_k) \left(h_k^{(i,j)}(\boldsymbol{x}_k) - 2y_k^{(i,j)} \right) \right] \right\} \tag{40}$$

Using the intensity normalized pixel contribution defined in Eq. (6), the above equation can be rewritten as follows:

$$l\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right) = \exp\left[-\frac{I_k^2}{2\sigma^2} \sum_{i \in \mathcal{C}_y(\tilde{\boldsymbol{x}}_k)} \sum_{j \in \mathcal{C}_x(\tilde{\boldsymbol{x}}_k)} \left(\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k) \right)^2 + \frac{I_k}{\sigma^2} \sum_{i \in \mathcal{C}_y(\tilde{\boldsymbol{x}}_k)} \sum_{j \in \mathcal{C}_x(\tilde{\boldsymbol{x}}_k)} y_k^{(i,j)} \cdot \tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k) \right] \tag{41}$$

The maximum likelihood estimate of intensity can then be computed by using the first order necessary conditions of optimality, i.e.:

$$\hat{I}_k = \underset{I_k}{\operatorname{argmax}}\ l(\boldsymbol{y}_k|\boldsymbol{x}_k) \tag{42}$$

However, this is identical to minimizing the negative log likelihood, $\mathcal{L}(\boldsymbol{y}_k|\boldsymbol{x}_k)$ as shown below

$$\mathcal{L}(\boldsymbol{y}_k|\boldsymbol{x}_k) = \left[\frac{I_k^2}{2\sigma^2}\sum_{i\in\mathcal{C}_y(\tilde{\boldsymbol{x}}_k)}\sum_{j\in\mathcal{C}_x(\tilde{\boldsymbol{x}}_k)}\left(\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k)\right)^2 - \frac{I_k}{\sigma^2}\sum_{i\in\mathcal{C}_y(\tilde{\boldsymbol{x}}_k)}\sum_{j\in\mathcal{C}_x(\tilde{\boldsymbol{x}}_k)} y_k^{(i,j)}\cdot\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k)\right] \tag{43}$$

Computing the appropriate partial derivative and setting it equal to zero results in the following:

$$\frac{\partial\mathcal{L}(\boldsymbol{y}_k|\boldsymbol{x}_k)}{\partial I_k} = 0 \tag{44}$$

$$\frac{I_k}{\sigma^2}\sum_{i\in\mathcal{C}_y(\tilde{\boldsymbol{x}}_k)}\sum_{j\in\mathcal{C}_x(\tilde{\boldsymbol{x}}_k)}\left(\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k)\right)^2 - \frac{1}{\sigma^2}\sum_{i\in\mathcal{C}_y(\tilde{\boldsymbol{x}}_k)}\sum_{j\in\mathcal{C}_x(\tilde{\boldsymbol{x}}_k)} y_k^{(i,j)}\cdot\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k) = 0 \tag{45}$$

$$\hat{I}_k = \frac{\sum_{i\in\mathcal{C}_y(\tilde{\boldsymbol{x}}_k)}\sum_{j\in\mathcal{C}_x(\tilde{\boldsymbol{x}}_k)} y_k^{(i,j)}\cdot\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k)}{\sum_{i\in\mathcal{C}_y(\tilde{\boldsymbol{x}}_k)}\sum_{j\in\mathcal{C}_x(\tilde{\boldsymbol{x}}_k)}\left(\tilde{h}_k^{(i,j)}(\tilde{\boldsymbol{x}}_k)\right)^2} \tag{46}$$

From Eq. (46), an interesting property to note is that the maximum likelihood estimate for the intensity only depends on the pixel measurements and the blur factor and it *does not* directly depend on the statistics of the background noise.

3. APPROACH

3.1 Data Processing

This section discusses an overall approach that can be used to detect low SNR targets in real time. In order to use the theory discussed above, it is important that the images are pre-processed such that the noise in the image is zero mean with known standard deviation. Typically, the sensor has hot pixels and bias. These effects can be removed using dark frame subtraction. Additionally, the variations in pixel-to-pixel sensitivity of the detector can also be mitigated using flat field correction. The brightness of the sky for ground-based sensors varies in an image and it must be estimated and subtracted as it causes a bias in the pixel measurements. Once these effects are mitigated, the image must only contain zero-mean noise and signals from point sources, including stars. It is not required to subtract the stars from the frame before detecting low SNR targets. However, this will result in many detections as stars are typically much brighter than the targets.

3.2 Track Hand-off to a Multi-Bernoulli Filter

The likelihood ratio recursion along with the intensity marginalized likelihood ratio can be implemented in a particle-filter based framework. However, FiSSt based methodologies are more appropriate since they are inherently designed to track multiple targets [11]. These methods are typically robust and allow for modeling target dynamics of choice. However, they suffer from curse of dimensionality due to the particle-based implementations. Specifically, the ability to detect low SNR targets heavily depends on the birth model. Since the targets can appear anywhere in the frame, this can be quite expensive computation especially for blind detection of targets using a wide field-of-view sensor.

As it was shown in the previous section, the intensity marginalized likelihood ratio only depends on the position of the targets in the surveillance region. This property can be leveraged to discretize the position states

in the surveillance region and compute the intensity marginalized likelihood ratio in *parallel*. Additionally, under constant velocity assumption, the integral in Eq. (29) is trivial to compute. This essentially becomes a point-mass filter implementation of the likelihood ratio recursion, which allows for quick identification of possible targets. Any target track in the surveillance region with a likelihood ratio above a set threshold can then be passed on to a Multi-Bernoulli Filter operating on image data. Several methods to compute this threshold have been proposed, including thresholding based on minimizing Bayes' risk, confidence level, and Neyman-Pearson hypothesis testing. Ultimately, this methodology allows the point-mass filter to act a temporary detector for the full Multi-Bernoulli filter. Therefore, under the assumptions made in this paper, the computational efficiency of the likelihood recursion is combined with the sensitivity of the Multi-Bernoulli filter giving a real-time implementation.

4. RESULTS

4.1 Comparison to Synthetic Tracking / Shift-and-Add Methods

Synthetic tracking involves computationally searching for an appropriate velocity vector, shifting successive frames relative to each other and then co-adding the shifted frames in post-processing to create a long-exposure image as if a telescope were tracking the object. We know that the signal improves linearly when images are added and at the same time, the noise increases by the square root of the number of images (N). Therefore, the SNR gain from synthetic tracking is approximately $\sqrt{N}$. However, we must also note that synthetic tracking inherently assumes that the target is present in all of the images that are being shifted and added. If, for example, the target appears after 10 frames have been collected, the sum of the first 10 frames have no signal but the standard deviation of the background has increased by $\sqrt{10}$, and therefore significantly degrades the performance of this method.

In order to compare the likelihood ratio recursion, let's assume that we have discretized the target position space such that each state corresponds to a single pixel in the image. Using a uniform prior over these discretized positions, computing the measurement likelihood ratio for each of these states, and updating them using Eq. (30) will result in a $n \times m$ grid that is the same size as the image. Let's refer to this as the *target likelihood ratio map*. As mentioned earlier, synthetic tracking assumes that the target already exists in each frame and will not disappear from the surveillance region. Making the same assumption in the likelihood ratio recursion (Eq. 29) gives the following equation:

$$\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k}) = \int_{\mathcal{X}} p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k)\ \Lambda(\boldsymbol{x}_k|\boldsymbol{Y}_{1:k})\ d\boldsymbol{x}_k \cancel{+ p(\boldsymbol{x}_{k+1}|\phi_k)} \tag{47}$$

Synthetic tracking also assumes constant velocities for the target. It is trivial to see that making the same assumptions in the above equation will result in the integral computation to be identical to shifting and adding the *target likelihood ratio map*. That is, if we are interested in v velocity hypotheses, the integral boils down to shifting $\Lambda(\boldsymbol{x}_{k+1}|\boldsymbol{Y}_{1:k})$ by each velocity hypothesis and generating v *target likelihood ratio maps*. The update equation, requires that target likelihood ratio map be multiplied by the measurement likelihood map. However, if we instead keep track of the target log likelihood ratio map, then update equation simplifies to point-wise adding the measurement log likelihood map. Therefore, we can see the analogy between synthetic tracking and target likelihood ratio recursion.

To summarize, under constant velocity and no target birth process assumption, synthetic tracking involves shifting and adding "data". Under the same assumptions, the target likelihood ratio recursion simplifies to shifting and adding "target log likelihood ratios". However, there are few differences to note. Synthetic tracking is not able to use sub-pixel velocity hypotheses without interpolating data. This is easily handled in the target likelihood ratio recursion since the refinement of discretization of the surveillance region is up to the user. Another important concept to note is that synthetic tracking cannot handle birth and death of targets without analyzing different combinations of stacking the frames or compromising the SNR gain. In other words, given a stack of 5 frames, if a target was born after 3 frames, synthetic tracking would have to shift-and-add frames 1 through 5, 2 through 5, 3 through 5, and so on to gain optimal SNR improvement. This combinationorial process would have to be performed for all the different velocity hypothesis. If this

is not performed, then shifting-and-adding frames without signal can severaly degrade the performance. However, the birth and death processes are handled in likelihood ratio recursion by default. We can show the similarities and differences between these two methods by an simple simulation.

Let us only consider the case of one stationary target with its location directly in the center of a pixel. This guarantees that there are no losses while shift-and-adding. In this case, synthetic tracking therefore simplifies to naive stacking of frames. Additionally, let the target intensity be fixed throughout both the simulation once it enters the surveillance region. Now, let's analyze the performance of both the methods in the following two scenarios:

- Scenario 1: Set of 30 frames are collected. The target is known to be in the surveillance region in all of the 30 frames. In the real world, this could represent the case of where a sensor was tasked to detect the target by rate-tracking.
- Scenario 2: Set of 50 frames are collected. However, the target is birthed at the 30^{th} frame. This could represent the case where a sensor is tasked to perform blind detection in cislunar space. This could also represent a case where a sensor was tasked to detect a target by rate-tracking, but the target was eclipsed for the first 29 frames.

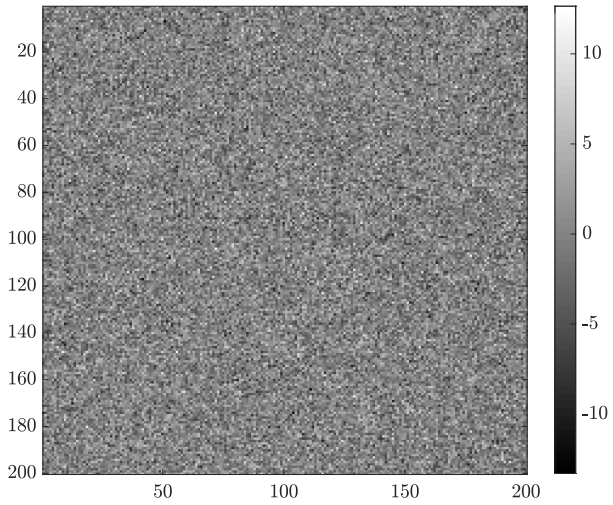
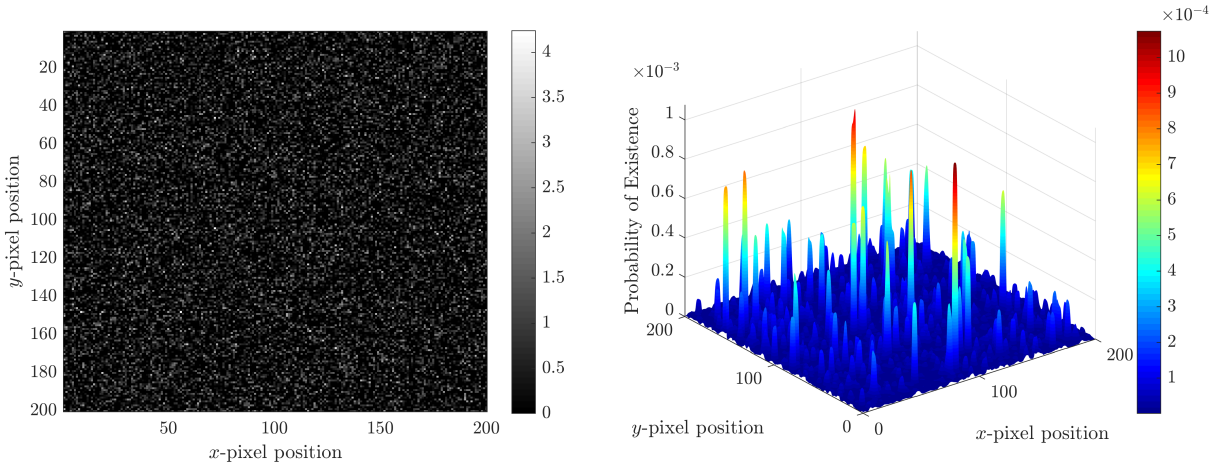


Fig. 1: Sample image from the simulation showing the target with noise.

4.1.1 Scenario 1 Results

The results are shown after processing the 1^{st} and the 8^{th} frames. Fig. 2 shows the due to the low SNR of the target, neither synthetic tracking nor likelihood ratio recursion is able to confirm the existence of the target. Fig. 3 shows that stacking method was able to increase the peak pixel SNR to $\sim$5.5 after 8 frames. However, it is important to note that there are several other pixels in the frame that have comprable pixel SNR. The likelihood recursion method estimates the probability of detection to be $\sim$0.13 after 8 frames. However, all other target states have probability of existence ≤ 0.01.

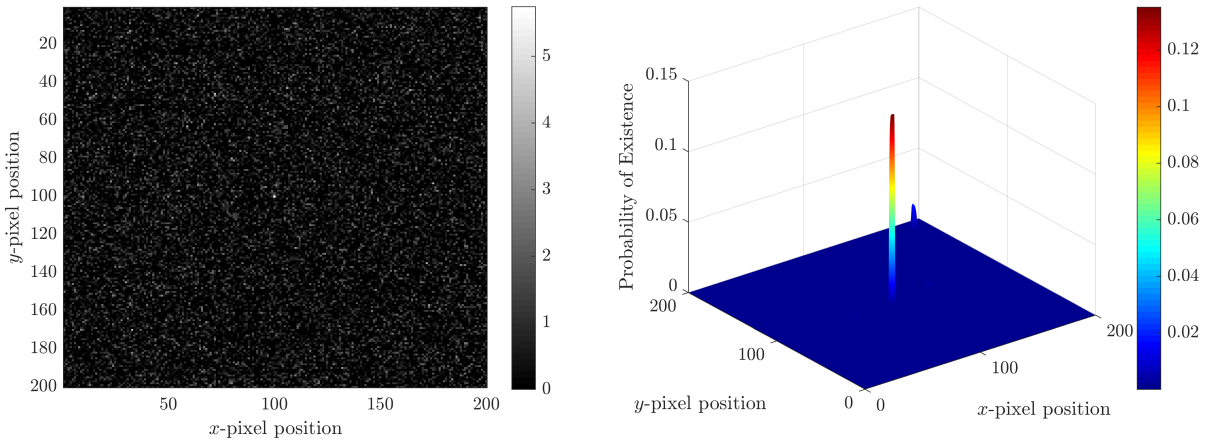
The paper on Synthetic tracking by Shao. et al discusses the false alarm rate associated with low pixel SNR thresholds and uses a threshold of $\sim$7 [1]. From Fig. 4, it is important to note that in order to reach a pixel SNR of 7, theory suggests that at least 11 frames are needed, for this simulation. However, the likelihood ratio recursion confirms the target's existence well under 10 frames based on the Monte Carlo runs.



(a) Pixel SNR after 1st frame.

(b) Probability of Existence after 1st frame.

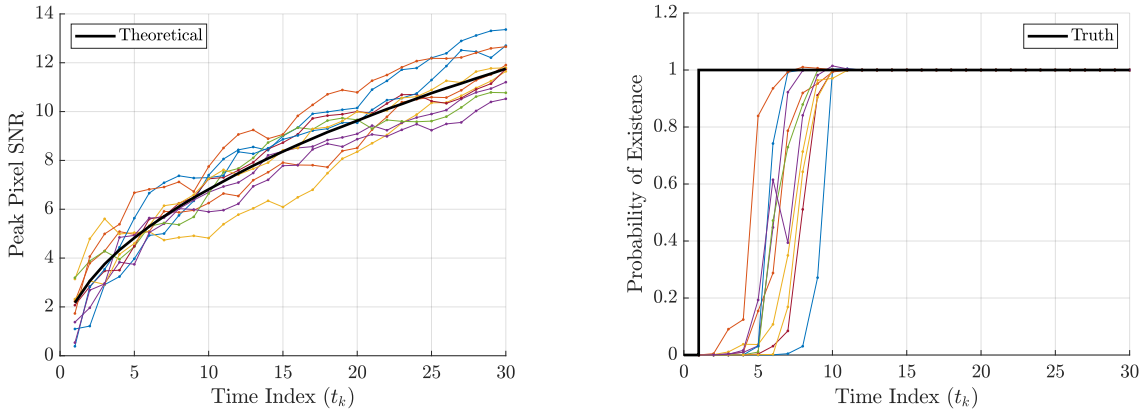
Fig. 2: Scenario 1 - Pixel SNR from stacking & existence probability from likelihood ratio recursion after frame 1.



(a) Pixel SNR after 8th frame.

(b) Probability of Existence after 8th frame.

Fig. 3: Scenario 1 - Pixel SNR from stacking & existence probability from likelihood ratio recursion after frame 8.



(a) Pixel SNR over time index.

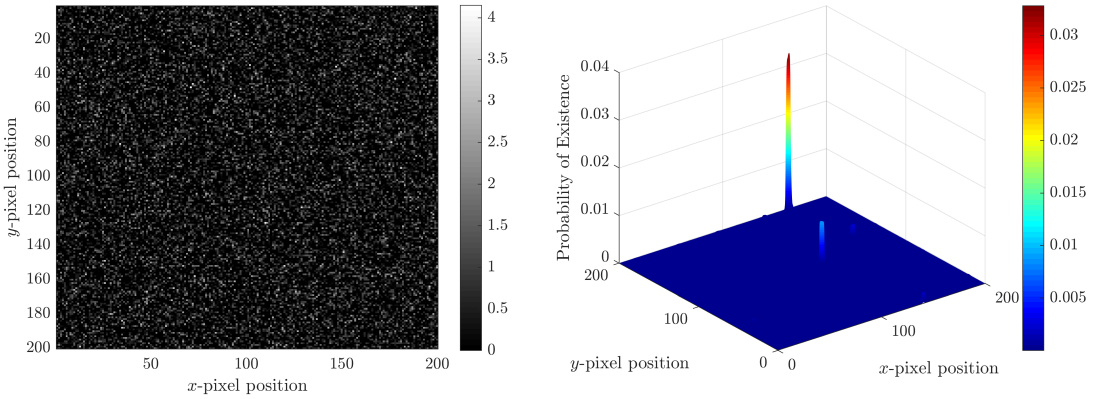
(b) Probability of Existence over time index.

Fig. 4: Scenario 1 - Pixel SNR & existence probability compared over 10 Monte Carlo runs. **It is important to note that in order to reach a pixel SNR of 7, theory suggests that at least 11 frames are required. However, the likelihood ratio recursion confirms the target's existence well under 10 frames based on the Monte Carlo runs.**

4.1.2 Scenario 2 Results

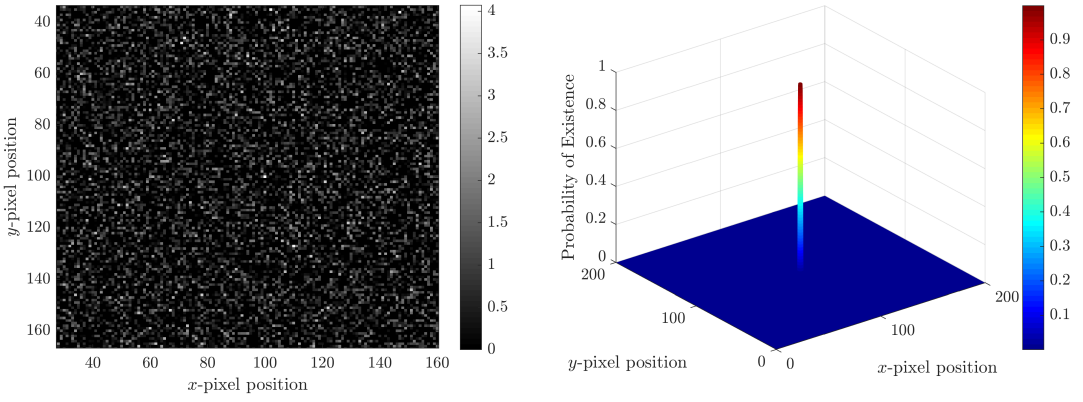
The same simulation is run again, but this time, the target appears in frame 30. Fig. 5 shows the pixel SNR from stacking 29 frames before the target appears. It is evident that there are several pixels that already have an SNR of 4 and the standard deviation of the noise has also increased significantly. The probability of existence after 29 frames for all the states is under 0.05. As expected, neither method confirms the existence of a target.

Fig. 6 again shows the pixel SNR and probability of existence estimate from both the methods. Likelihood ratio recursion estimates the probability of existence to be 0.99 only after 8 frames from when the target initially appeared. However, as expected, stacking method is not able to increase the SNR more than ∼4 due to the increase in standard deviation of the background from first 29 frames. From Fig. 7, in order to reach a pixel SNR of 7, theory suggests that more than 50 frames are required, i.e. more than 20 frames from when the target initially appears. However, the likelihood ratio recursion confirms the target's existence well under 40 frames, i.e. under 10 frames from when the target initially appears, just as it did in Scenario 1.



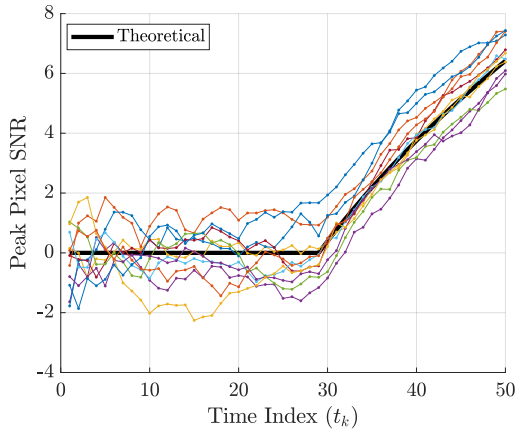
(a) Pixel SNR after 29$^{\text{th}}$ frame. (b) Probability of Existence after 29$^{\text{th}}$ frame.

Fig. 5: Scenario 2 - Pixel SNR from stacking & existence probability from likelihood ratio recursion after frame 29. Note that the target appears in frame 30.

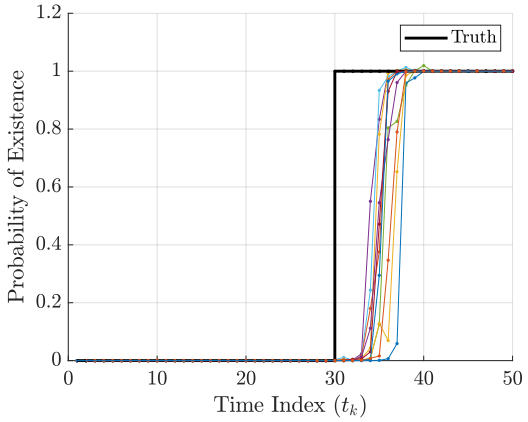


(a) Pixel SNR after 37$^{\text{th}}$ frame. (b) Probability of Existence after 37$^{\text{th}}$ frame.

Fig. 6: Scenario 2 - Pixel SNR from stacking & existence probability from likelihood ratio recursion after frame 37.



(a) Pixel SNR over time index.



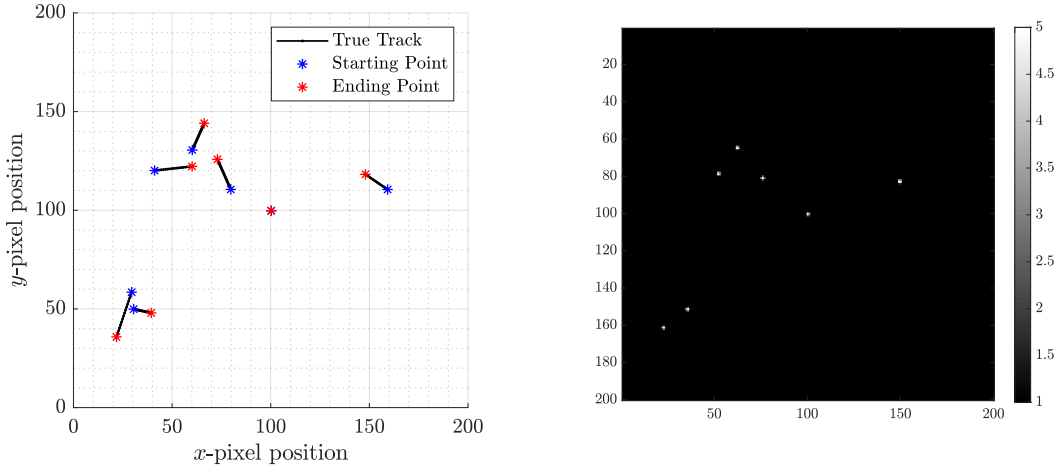
(b) Probability of Existence over time index.

Fig. 7: Scenario 2 - Pixel SNR & existence probability compared over 10 Monte Carlo runs. Note that the target appears in frame 30. **It is important to note that in order to reach a pixel SNR of 7, theory suggests that more than 50 frames are required, i.e. more than 20 frames from when the target initially appears. However, the likelihood ratio recursion confirms the target's existence well under 40 frames, i.e. under 10 frames from when the target initially appears, based on the Monte Carlo runs.**

4.2 Multi-Target Tracking Results using Simulated Data

This section shows the results from using target likelihood ratio recursion on images in which 7 targets with different intensities appear and disappear at various different times. A stack of 45 frames with 200×200 pixel grid is considered for this simulation. The intensity for targets vary from 15 pixel counts to 20 pixel counts. The noise is sampled from a Gaussian with zero mean and $\sigma = 3$. The blur factor, Σ, was chosen to be equal to 0.7. This corresponds to the peak pixel SNR for the targets varying from $\sim$1.6 to $\sim$2.1. The intensity marginalized likelihood ratio was computed at every quarter pixel increments in the image. Any target track with a probability of existence greater than 0.3 were handed-off to the MBF.

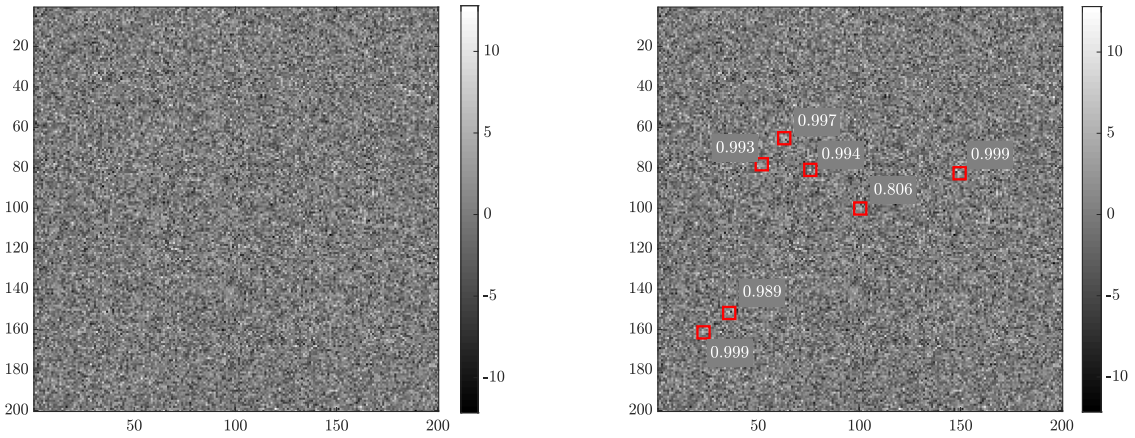
Fig. 8a shows the true tracks of the targets in pixel space. Fig. 8b shows only the targets along with their intensity for frame 18 of the dataset, whereas Fig. 9a shows the actual image with noise. Fig. 9b shows the detections from the MBF along with the probability of existence estimate for each target track after they have been handed-off from the target likelihood ratio recursion. Fig. 10 shows the cardinality, i.e., the total number of targets detected in each frame over time and is compared to the true number of targets. Since the targets have very low SNR, the likelihood ratio recursion requires a few frames to confirm the existence of the target. Finally, Fig. 11 compares the position state estimates from the filter to the truth.



(a) True Target Tracks.

(b) Without noise.

Fig. 8: True target tracks along with their intensities.



(a) Noisy data.

(b) Target detections overlayed.

Fig. 9: Image 18 of the dataset. The left image shows raw frame and the right image shows the target estimates from the filter along with the probability of existence.

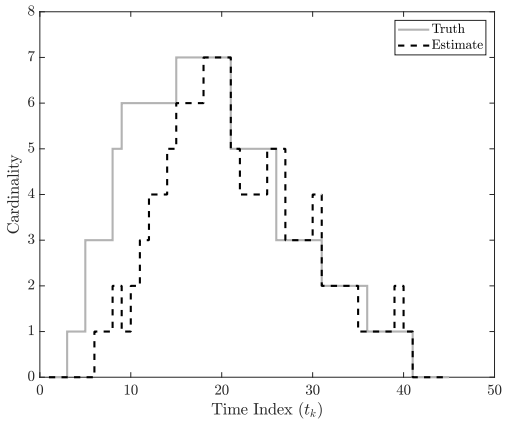


Fig. 10: Cardinality over time index, t_k.

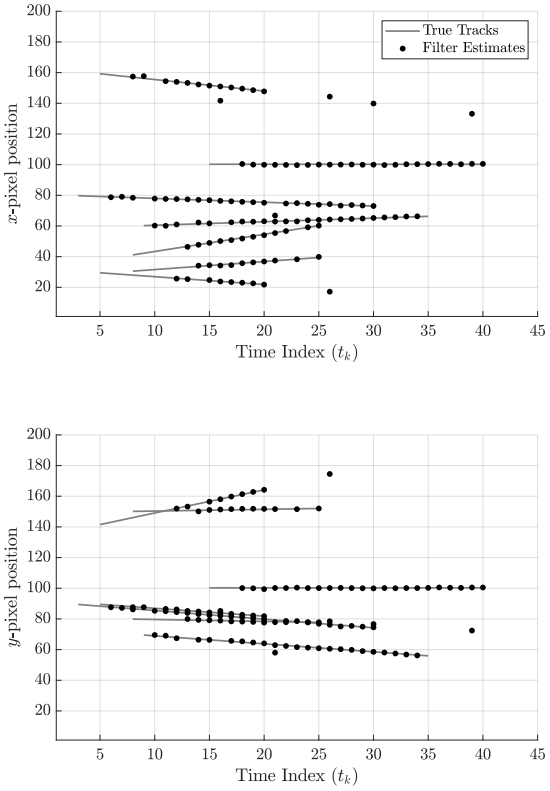


Fig. 11: Position state estimates compared to truth over time index, t_k.

5. CONCLUSION

This paper uses the concept of separable likelihood function to motivate the use of target likelihood ratio recursion for detecting and tracking low SNR targets. The target likelihood ratio based filtering is reviewed and a new intensity marginalized likelihood ratio is introduced. This likelihood ratio is then used with a point-mass approximation of the likelihood recursion, which allows for the detection and tracking of multiple moving targets. The intensity marginalized likelihood ratio can be computed in real-time if implemented carefully. Assuming constant velocity for target motion in a short period of time allows for quick detection of possible tracks. Any targets tracks with a probability of existence greater than a chosen threshold are passed to an image-based Multi-Bernoulli filter for track maintenance and account to for process noise in target's motion. The intensities are computed using maximum likelihood estimation and a closed-form solution is also introduced. This methodology is compared to the well-known synthetic tracking/ shift-and-add algorithm. Additional details on the exact implementation along with results on real data from various sensors will be shown in a follow-on paper.

6. ACKNOWLEDGMENTS

This research was funded by the National Defense Science and Engineering Graduate (NDSEG) Fellowship awarded by the Department of Defense (DoD) and Air Force Office of Scientific Research (AFOSR). The authors would also like to acknowledge Dr. Nisar Ahmed, Assistant Professor at University of Colorado - Boulder, for passing on his thorough knowledge in estimation theory as well as providing thoughtful insight & motivation for this work.

7. REFERENCES

[1] Michael Shao, Bijan Nemati, Chengxing Zhai, Slava G Turyshev, Jagmit Sandhu, Gregg Hallinan, and Leon K Harding. Finding Very Small Near-Earth Asteroids using Synthetic Tracking. *The Astrophysical Journal*, 782(1):1, 2014.

[2] Ronald PS Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., 2007.

[3] Timothy S Murphy, Marcus J Holzinger, and Brien Flewelling. Space Object Detection in Images using Matched Filter Bank and Bayesian Update. *Journal of Guidance, Control, and Dynamics*, 40(3):497–509, 2017.

[4] D. J. Salmond and H. Birch. A Particle Filter for Track-before-Detect. *Proceedings of the American Control Conference*, 5:3755–3760, 2001.

[5] Y. Boers and J. N. Driessen. Particle Filter based Detection for Tracking. *Proceedings of the American Control Conference*, 6:4393–4397, 2001.

[6] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle filters for Tracking Applications*, volume 685. 2004.

[7] Mark G. Rutten, Neil J. Gordon, and Simon Maskell. Efficient Particle-based Track-before-Detect in Rayleigh Noise. *Proceedings of the Seventh International Conference on Information Fusion, FUSION 2004*, 2(1):693–700, 2004.

[8] Roy L. Streit, Marcus L. Graham, and Michael J. Walsh. Multitarget Tracking of Distributed Targets Using Histogram-PMHT. *Digital Signal Processing: A Review Journal*, 12(2-3):394–404, 2002.

[9] Samuel J. Davey, Mark G. Rutten, and Brian Cheung. Using Phase to Improve Track-before-Detect. *IEEE Transactions on Aerospace and Electronic Systems*, 48(1):832–849, 2012.

[10] Ba Ngu Vo, Ba Tuong Vo, Nam Trung Pham, and David Suter. Bayesian Multi-Object Estimation from Image Observations. *2009 12th International Conference on Information Fusion, FUSION 2009*, pages 890–898, 2009.

[11] Ba Ngu Vo, Ba Tuong Vo, Nam Trung Pham, and David Suter. Joint Detection and Estimation of Multiple Objects from Image Observations. *IEEE Transactions on Signal Processing*, 58(10):5129–5141, 2010.

[12] Shahzad Virani, Timothy Murphy, Marcus J Holzinger, and Brandon A Jones. Empirical Dynamic Data Driven Detection and Tracking Using Detectionless and Traditional FiSSt Methods. *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2017.
[13] L. Stone, T. L. Corwin, and C. A. Barlow. Bayesian Multiple Target Tracking. 1999.