ML-PMH Tracking in Three Dimensions Using Cluttered Measurements From Multiple Two-Dimensional Sensors

ZACHARIAH SUTTON
PETER WILLETT
TIM FAIR
YAAKOV BAR-SHALOM

The maximum-likelihood probabilistic multi-hypothesis tracker (ML-PMHT) is a tracking method whose flexibility and scalability derive from relinquishing the assumption that each target emits at most one "hit" per scan of the sensor. This is an ML method that essentially reduces to an optimization problem—recursively maximizing a likelihood function that is simple to evaluate given a batch of observations. Unlike maximum *a posteriori* or minimum mean squared error (MMSE) trackers, this likelihood maximization tracker requires neither prior knowledge about target motion nor measurement association, making it conceptually easy to work with. Here, this method is used to track targets in a three-dimensional "global" space with observations provided by multiple two-dimensional sensors placed throughout the global space. Since the observation model is non-linear, the likelihood maximization is done via hill climbing. For this purpose, we also address the issue of "hill finding." Due to the presence of clutter in the measurement model, the likelihood is a multi-modal function of the parameter space. That is, there are multiple hills in the likelihood function, and it is of great advantage to the tracker to initialize the hill climber close to the right hill—the one whose peak is the global maximum. In this work, we present a data-driven method of initializing the hill climber based on the received observations.

Manuscript received August 19, 2021; revised September 27, 2021; released for publication December 1, 2021.

Z. Sutton, P. Willett, and Y. Bar-Shalom are with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT, USA E-mail: zachariah.sutton@uconn.edu, peter.willett@uconn.edu, yaakov.bar-shalom@uconn.edu
T. Fair is from Toyon Research Corporation, E-mail: tfair@toyon.com

This work was partially supported by AFOSR under contract FA9500-18-1-0463.

1557-6418/21/$17.00 © 2021 JAIF

I. BACKGROUND

The maximum-likelihood probabilistic multihypothesis tracker (ML-PMHT) is an ML target tracking paradigm that is convenient in cases where data association—the measurement-to-target assignment processes prior to updating the estimate—involves significant numerical complexity, generally (but not always) due to heavy clutter. In some settings, it is possible that a single target will result in multiple measurements at a particular sensor and time (tracking of "extended objects," for instance). In such cases, filters that employ "hard" data association (the JPDA [20] and random finite set filters such as the multi-Bernoulli [6], [15]) will be sub-optimal since they make the fundamental assumption that each target being tracked produces at most one measurement per sensor per time step. In contrast, the likelihood function used in the ML-PMHT is formulated by considering each measurement individually, and applying a probability mass function over the possible measurement generating processes (targets and clutter). That is, instead of assuming that a particular measurement has come from a particular target and evaluating the measurement likelihood with that assignment, the ML-PMHT formulates the measurement likelihood with a "soft" assignment that accounts for uncertainty as to the process from which a particular measurement originated. This formulation naturally allows for the possibility that a target has originated multiple measurements in a single "scan" of a sensor. Along with being a better representation of reality in some settings, the soft assignment also avoids the computational bookkeeping cost of the hard assignment problem that the data association filters must solve for each scan with relatively expensive routines like Murty's k-Best Assignment Algorithm [12]. Thus, the ML-PMHT approach may also be desirable in some settings where computational cost is a consideration.

The ML-PMHT likelihood formulation is borrowed from the PMHT framework [7], [10]. The ML-PMHT differs, however, in that it treats the target state (joint target state in the case of multiple targets) as an unknown deterministic parameter, and obtains an ML estimate of the parameter based on batches of measurement scans. It has shown especially good performance in scenarios with high levels of clutter [14], [21].

This work will use the ML-PMHT to perform data-batch-based tracking of targets in a three-dimensional "global" space based on multiple passive sensors that return two-dimensional measurements. A generalized measurement model is presented that can be adapted to any type of sensor that returns measurements that can be transformed into lines-of-sight. Some common sensor types that could be used with this model are focal plane arrays (cameras) with measurements given in the two-dimensional image space, or passive radars that return azimuth and elevation angles (or azimuth and elevation angle sines).

It is assumed that all sensors report to a central processor that performs likelihood maximization based on all observed data. The central optimization reveals another benefit of the ML-PMHT in that data from multiple sensors are naturally included in the likelihood formulation in a simple linear sum manner. While sequential updating over sensors is a common practice in data association filters, it is theoretically sub-optimal [20]; and optimality with multiple sensors according to the rules of "hard" data association is computationally costly.

The ML-PMHT reduces to a conceptually straightforward optimization problem where likelihood maximization happens over a multi-dimensional parameter space. However, an important practical consideration in optimization problems is initialization: how to pick the "initial guess" for the parameter value. Initialization is particularly important in the tracking setting since we assume the presence of clutter measurements, which makes the likelihood multi-modal. That is, coincidental "patterns" in clutter measurements can lead to false maximums in the likelihood value that gradient-based maximizers will reach in error if not initialized carefully. Although the global maximum tends to occur at the true parameter value, a maximizer must be initialized sufficiently close to the global maximum in order to reach it. For this purpose, a "hill finding" method is presented where received measurements are used to identify statistically significant points in the global parameter space that can be used to initialize the maximization. This hill finding routine is conceptually separate from the ML-PMHT, and is perhaps the most novel contribution of this work. The method represents a means of identifying pairs of line-of-sight measurements from separate sensors that strongly correlate to a single point in three-dimensional space. Thus, it could theoretically be used in other settings where one would wish to identify points in three-dimensional space that are statistically supported by lower-dimensional measurements. (It could inform the "target birth" process in the multi-Bernoulli filter, for example.)

The benefit of central data processing and the resulting ability to perform the "hill finding" is demonstrated by comparing the method to a decentralized option where ML estimates are obtained individually by each sensor, then fused in the global space. It is shown that centralized optimization has a significant advantage in settings with low target visibility. A comparison is also made to the joint probabilistic data association filter (JPDAF).

The paper is structured as follows. The models used in the work are presented in Section II, including the target parameter model, the measurement model, and the model of the geometric arrangement of sensors. The ML-PMHT likelihood formula is given in Section III. The "hill finder" is presented in Section IV, and a step-by-step summary of the overall ML-PMHT method is given. Simulated results are shown in Section V.

II. MODELING ASSUMPTIONS

The model assumes a three-dimensional global space in which targets are to be tracked. Measurements are received from a group of N^{s} sensors distributed around the global space. This work uses a conventional Cartesian coordinate system in the global space, but in theory it could be replaced by a local north-east-down reference frame or any other space where the following conditions are fulfilled:

1) The motion of targets can be (approximately) parameterized in the space.
2) The *pose* of every sensor is known in the space.

The pose of a sensor parameterizes the transformation between the global coordinate system and the sensor's coordinate system. The sensors are assumed to have six degrees of freedom (DOFs)—three translational (location) and three rotational (pointing). So the pose consists of six known parameters for each sensor. If a sensor's pose changes over time, it is assumed to be known for each point in time that a measurement is received. The conventions used for the pose and the resulting transformations are discussed in more detail in Section II.C.

A. Target Motion

The user must choose a batch size parameter N^{b}, which is the number of scans from each sensor that will be used in the likelihood evaluation. It is assumed that, for all targets, the true target motion can be reasonably approximated by a constant velocity model over the duration of the batch. That is, for any discrete global time index k, the motion of target j over the past $N^{\mathrm{b}}-1$ sampling periods (N^{b} sampling points) is given by

$$\mathbf{X}_{j,\ell} \simeq (\mathbf{X}_{j,k} - \mathbf{X}_{j,k-N^{\mathrm{b}}+1})\frac{n}{N^{\mathrm{b}}-1} + \mathbf{X}_{j,k-N^{\mathrm{b}}+1} \qquad (1)$$

$$n = 0, \ldots, N^{\mathrm{b}}-1,$$

where $\mathbf{X}_{j,\ell}$ denotes a three-element column vector containing the target's position in the global Cartesian space at global time step ℓ and n is a local time index such that $\ell = k - N^{\mathrm{b}} + 1 + n$.

A constant sampling period is assumed here, but there is no loss of generality. With the batch size chosen, the motion in (1) is entirely parameterized by $\mathbf{X}_{j,k-N^{\mathrm{b}}+1}$ and $\mathbf{X}_{j,k}$—the positions of target j at the start and end times of the batch. Thus, the motion to be estimated via likelihood maximization can be described with six parameters for each target. For a scenario with N^{t} targets

present, form a parameter vector

$$\boldsymbol{\mathcal{X}}^k \triangleq \begin{bmatrix} \mathbf{X}_{1,k-N^{\mathrm{b}}+1} \\ \mathbf{X}_{1,k} \\ \vdots \\ \mathbf{X}_{N^{\mathrm{t}},k-N^{\mathrm{b}}+1} \\ \mathbf{X}_{N^{\mathrm{t}},k} \end{bmatrix}. \tag{2}$$

We ultimately will be maximizing the log likelihood function over this vector given $\boldsymbol{\mathcal{Z}}^k$—the batch of measurements up to and including time step k—to obtain the estimate

$$\boldsymbol{\mathcal{X}}^k_{\mathrm{ML}} = \arg\max_{\boldsymbol{\mathcal{X}}^k \in \mathbb{R}^{6N^{\mathrm{t}}}} \mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k). \tag{3}$$

The definition of $\boldsymbol{\mathcal{Z}}^k$ and the formulation of the likelihood function $\mathcal{L}(\cdot)$ are discussed later. The maximization is mentioned here to emphasize that the dimensionality of the space over which maximization is performed increases by 6 for each additional target. This is a computational consideration in a practical application.

In theory, this likelihood maximization requires only the constant velocity assumption over the course of any particular batch. However, it will be of use if the user has some more prior knowledge about target motion. Specifically, if there is knowledge available about the range of possible target speeds, it will prove useful in the initialization of the hill climber as discussed in Section IV.A.

Note that the particular six-parameter motion model used here is not the only model compatible with batch tracking. One could also choose—at the cost of computation time—to use a nine-parameter (initial position, initial velocity, and acceleration) model or any other method of parameterizing the target motion over the duration of the batch.

B. Measurement Model

The tracker developed in this work uses line-of-sight measurements. That is, it is assumed that each and every sensor returns some form of two-element measurements that can be used to parameterize a line-of-sight beginning at the origin of its own coordinate system and extending infinitely in the direction of sight. The line-of-sight measurement model is a fundamental feature of this work. However, there are multiple types of two-dimensional measurements that provide a line-of-sight, which allows the sensor type to remain ambiguous. Simple passive radar models directly provide line-of-sight measurements. Also, a point in the image space of a camera can be converted into a line-of-sight given the camera model. The simplicity of radars in this context renders them rather uninteresting. Thus, cameras are assumed in the peripheral theoretical modeling and verification in this work, without loss of generalization in the fundamental aspects of the work (likelihood maximization). This section discusses the relationship between image space and line-of-sight measurements within the coordinate system of a sensor's reference frame. The transformation between a sensor's coordinate system and the global coordinate system is discussed in Section II.C.

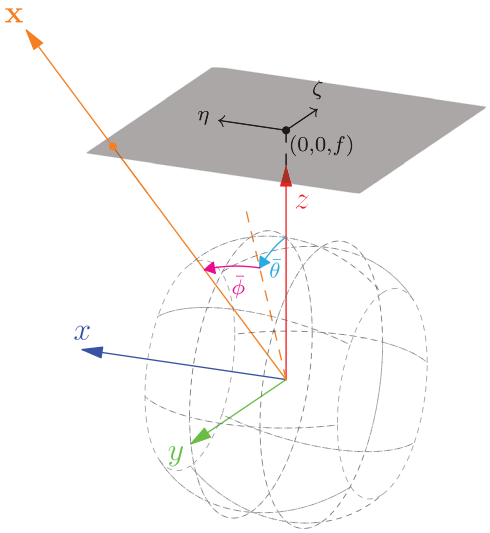


Fig. 1. Illustration of the sensor reference frame convention used in this work. Two different measurement types are shown. Given the coordinates of the vector $\mathbf{x}$, one could solve for $\bar{\theta}$ and $\bar{\phi}$, which is the azimuth–elevation measurement model assumed throughout this work. Given a point in the image plane (shaded) at a known distance f from the origin (where f has the same length units as η, ζ coordinate system), one could also calculate the corresponding azimuth and elevation angles.

When camera images are used with this algorithm, it is necessary to first perform measurement extraction. The assumption when using cameras as sensors is that targets have a contrasting appearance to the background. The extraction must find significantly bright or dark spots in an image, and condense each spot down to a point in the image space of the camera, which, in turn, can be converted into line-of-sight measurements via the camera model. Methods of extracting these measurements from images are discussed in [3] and [11]. The signal-to-noise ratio (SNR) of the scenario is an important factor in the measurement extraction step. Note that a significant portion of [3] is dedicated to defining the SNR. The extraction process will have a certain probability of extracting "false" measurements, which we will refer to as clutter.

Once measurement extraction has been performed on the images, the resulting measurements must be converted from the image space into lines-of-sight. Refer to Fig. 1 for a summary of this conversion. The camera model used here is the pinhole projection model [5], [23]. The image is treated as a plane parallel to the x–y plane of the sensor coordinate frame, set at some non-zero focal distance f along the sensor z axis. Let the coordinate system in the image plane be denoted by (η, ζ), which is centered on the sensor z axis, and has directional convention that agrees with the typical row–column format of images. Then a point $[\eta, \zeta]'$ in the image plane coordinate system has location $[\eta, -\zeta, f]'$ in the coordinate

system of the sensor's reference frame. Note that the image plane is treated as a continuous space (as opposed to a quantized space) since the measurement extraction process yields continuous values. Azimuth–elevation angles are used for parameterizing line-of-sight measurements. The azimuth angle θ is taken in the sensor y–z plane, with zero defined as the positive z axis and the positive direction defined as from the positive z axis to the positive y axis. The elevation angle ϕ is the angle between the sensor y–z plane and the positive x axis, defined as zero on the y–z plane and positive toward the positive x axis. Under this choice of convention, a given point in the image plane $[\eta, \zeta]'$ yields line-of-sight angles

$$\theta = \tan^{-1}\left(\frac{-\zeta}{f}\right), \tag{4}$$

$$\phi = \tan^{-1}\left(\frac{\eta}{\sqrt{\zeta^2 + f^2}}\right). \tag{5}$$

Note that this particular camera model is somewhat simplistic. It does not account for image distortion or other practical effects. If the user has a more accurate model of the cameras, it will be compatible with this algorithm so long as it provides a way to obtain lines-of-sight from points in the image. Since the algorithm ultimately works with line-of-sight measurements, this writing will occasionally use the general term "measurements" when referring to azimuth–elevation measurements.

Overall, it is assumed that at a particular time, each sensor i returns a set of line-of-sight measurements—potentially after conversion with (4) and (5)—which includes any target-originated measurements along with any measurements originating from the clutter process. That is, the set of measurements returned by sensor i at time ℓ can be denoted

$$\mathbf{Z}_{i,\ell} = \{\mathbf{z}_{i,\ell,m}\}_{m=1}^{N_{i,\ell}^{z}} = \left\{\begin{bmatrix}\theta_{i,\ell,m}\\ \phi_{i,\ell,m}\end{bmatrix}\right\}_{m=1}^{N_{i,\ell}^{z}}, \tag{6}$$

where each $\mathbf{z}_{i,\ell,m}$ is a two-element column vector and $N_{i,\ell}^{z}$ is the number of measurements at the current time/sensor.

It will be helpful later on, during formulation of the likelihood function, to have a simple expected value parameterization for the number of clutter and target-originating measurements. The expected numbers of received measurements are fundamentally tied to the measurement extraction process, which is left non-specific for most of this work. Let us assume some general extraction process such that $\lambda_{i,\ell}$ and $\varphi_{i,\ell}$ are the expected numbers of clutter measurements and target-originated measurements, respectively, in the scan from sensor i at time ℓ. For the sake of more generality, allow each individual target j to originate a potentially unique expected number of measurements $\varphi_{i,j,\ell}$ such that

$$\varphi_{i,\ell} = \sum_{j} \varphi_{i,j,\ell}. \tag{7}$$

For purposes of simulation later in this work, it will be assumed that the number of clutter measurements is Poisson with some expected value λ_i. It will be further assumed that, independently for each target present, sensor i either reports a single measurement with "detection probability" $p_{\mathrm{d},i}$ or "misses" the target. Thus, the expected number of target-originated measurements from each target is $p_{\mathrm{d},i}$, and $\varphi_{i,\ell} = p_{\mathrm{d},i}N^{\mathrm{t}}$ for all ℓ. These assumptions are made to fit with a typical model used in other trackers for the sake of comparison. However, one of the main benefits of the ML-PMHT likelihood formulation is that it is more flexible than trackers that consider one-to-one data associations. Whereas the JPDA and its derivative algorithms must make the fundamental assumption that each target produces at most one measurement per scan, the ML-PMHT formulation requires no such assumption. Some data association tracking methods do exist for extended targets (targets that produce more than one measurement in a single scan), and usually involve recursive estimation of properties (e.g., shape, size) of targets. The ML-PMHT offers a relatively cheap way around this extra estimation for the case when the extended target measurement assumption is true, but the shape/size properties of targets are not of particular interest. For instance, a simple model could assume that the number of measurements originating from target j is Poisson random number with expected value $\varphi_{i,j}$.

It is assumed that any target-originated measurements have a random additive measurement error. Since the ML-PMHT is an objective function optimization problem, the parameterization of the measurement error is somewhat flexible: Any objectively computable error probability density can be used. This work will use the typical Gaussian error assumption. That is, if a particular measurement $\mathbf{z}_{i,\ell,m}$ originates from target j, then it is a random vector given by

$$\mathbf{z}_{i,\ell,m} = \bar{\mathbf{z}}_{i,j,\ell} + \boldsymbol{\nu}_{i,j,\ell}, \tag{8}$$

where $\bar{\mathbf{z}}_{i,j,\ell}$ is the noiseless (zero-error) measurement by sensor i due to target j at time ℓ. Given the three-element vector $\mathbf{x}_{i,j,\ell}$ representing the Cartesian (x–y–z) position of target j *in the sensor i coordinate frame* at time ℓ, the noiseless measurement for the azimuth–elevation model is given by

$$\bar{\mathbf{z}}_{i,j,\ell} = \begin{bmatrix}\bar{\theta}_{i,j,\ell}\\ \bar{\phi}_{i,j,\ell}\end{bmatrix} = \begin{bmatrix}\tan^{-1}\left(\frac{y_{i,j,\ell}}{z_{i,j,\ell}}\right)\\ \tan^{-1}\left(\frac{x_{i,j,\ell}}{\sqrt{y_{i,j,\ell}^2 + z_{i,j,\ell}^2}}\right)\end{bmatrix}, \tag{9}$$

where $x_{i,j,\ell}$, $y_{i,j,\ell}$, $z_{i,j,\ell}$ are the individual components of $\mathbf{x}_{i,j,\ell}$. For a visual representation of this measurement model refer to Fig. 1 and treat the orange vector as the target position. The term $\boldsymbol{\nu}_{i,j,\ell}$ in (8) is a two-element multivariate random Gaussian vector with distribution

$$\boldsymbol{\nu}_{i,j,\ell} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{i,\ell}), \tag{10}$$

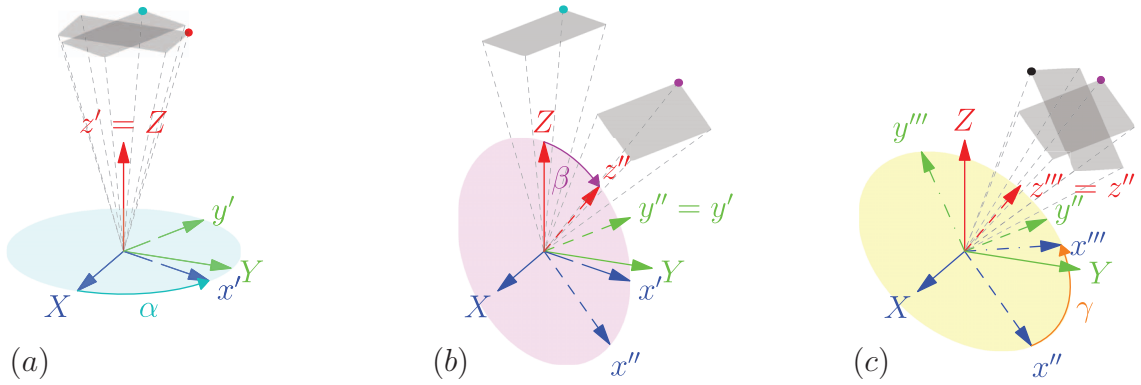


Fig. 2. Step-by-step illustration of a three-DOF proper Euler rotation. The total rotation is the result of applying three elemental rotations successively. The specific convention shown here and assumed in this work is a z–y–z intrinsic rotation. The focal plane shown in the figures matches the orientation of the one shown in Fig. 1. The origin of the (u, v) coordinate system is marked by a dot. The rotations bring the plane from the orientation marked by the red dot in (a) to the orientation marked by the black dot in (c). Orientations that are marked with a common color are the same. Colored circles attempt to show planes of rotation.

where, in general, the covariance matrix $\mathbf{R}_{i,\ell}$ is allowed to change from one sensor to the next and one time step to the next. Measurement errors are assumed to be independent between sensors and between time steps.

Clutter measurements are assumed to be uniformly distributed in the measurement space. It has been assumed in this work that each sensor has a limited field of view, modeled by setting limits on the azimuth and elevation angles symmetrically around zero. Then for a sensor i with a total azimuth field of view W_i^θ and total elevation field of view W_i^ϕ, the spatial distributions of the individual components of clutter measurements are

$$\theta_i^{\text{clutter}} \sim \mathcal{U}(-W_i^\theta/2,\ W_i^\theta/2), \tag{11}$$

$$\phi_i^{\text{clutter}} \sim \mathcal{U}\left(-W_i^\phi/2,\ W_i^\phi/2\right), \tag{12}$$

where, in general, different sensors are allowed to have differently sized fields of view—hence the indexing with i. For sensors with reasonably narrow fields of view (within the range of realistic cameras), the uniform distributions in azimuth and elevation result in image space measurements that are very close to uniform in the image plane. See Figs. 7 and 8 for a visual example of measurements in an image plane resulting from this model.

The total measurement batch $\mathcal{Z}^k$ in (3) can be expressed as

$$\mathcal{Z}^k = \{\{\mathbf{Z}_{i,\ell}\}_{i=1}^{i=N^{\text{s}}}\}_{\ell=k-N^{\text{b}}+1}^{\ell=k}. \tag{13}$$

Or, in words, $\mathcal{Z}^k$ is the set of all subsets of measurements (both target and clutter originated) from all sensors for the most recent N^{b} sample times (up to and including the current estimation time k).

C. Sensor-World Setup

Notice that in Fig. 1 and in the formulas in (8) and (9), it is assumed that $\mathbf{x}_{i,j,\ell}$—the three-dimensional Cartesian position of target j in the sensor i reference frame at time ℓ—is given. Since the likelihood in (3) is being maximized over target motion parameters given in the *global* reference frame, the relationship between the global reference frame and each sensor reference frame must be defined.

Let the axes of the global coordinate system be denoted $(X$–Y–$Z)$ and those of the sensor coordinate system $(x$–y–$z)$. The *pose* of the sensor is the position and orientation of its reference frame relative to the global reference frame, defined such that at a pose of zero, the two coordinate systems are one and the same.

Let the three rotational DOFs be described by the angles $\alpha_{i,\ell}$, $\beta_{i,\ell}$, and $\gamma_{i,\ell}$, where the time index ℓ is noted since the rotation of a sensor can change with time. There are multiple choices of convention for the actual meaning of these angles. To be exact, there are 12 unique ways to describe every possible orientation in terms of three angles.

The rotation convention used here is illustrated in Fig. 2, where, for convenience, a particular sensor at a particular time is considered and the indexing is momentarily dropped. The sensor coordinate system is initially aligned with the global coordinate system. The overall rotation is the combination of three intrinsic rotations performed sequentially.

(a) A rotation by α around the Z axis results in the new coordinate system $(x'$–y'–$z')$.

(b) Then a rotation by β around the y' axis to obtain $(x''$–y''–$z'')$.

(c) Finally, a rotation by γ around the z'' axis gives the fully rotated coordinate system $(x'''$–y'''–$z''')$.

Here, the positive direction for all rotations is given by the "right-hand rule." This describes what is commonly called an z–y–z intrinsic rotation. Here z–y–z refers to the sequence of rotation axes, and intrinsic refers to the fact that successive rotations are performed around the axes of the rotating coordinate system (sensor coordinate system) itself as opposed to rotating

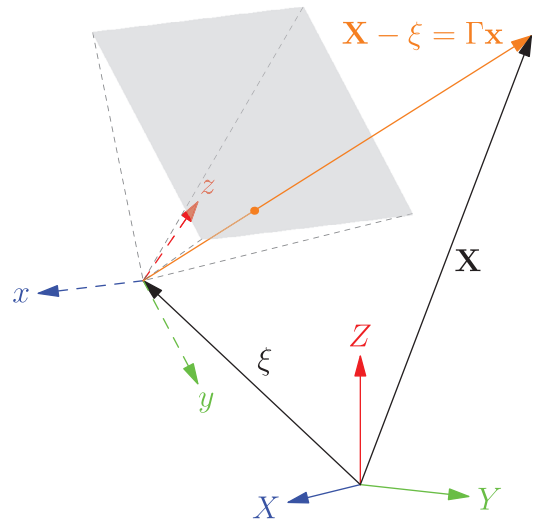


Fig. 3. Illustration of sensor-world setup for a particular sensor, target, and time step. Given target location $\mathbf{X}$ in the global (X–Y–Z) coordinates and sensor pose information $\boldsymbol{\Gamma}$, $\boldsymbol{\xi}$, the zero-error measurements $\bar{\theta}$, $\bar{\phi}$ in (9) are calculated by first evaluating $\mathbf{x}$—the target location in sensor (x–y–z) coordinates.

around the axes of the fixed global coordinate system. There is nothing particularly special about the choice of the z–y–z convention; in light of the camera model in Fig. 1, it is merely a way that one could conceivably go about orienting such a camera in a practical scenario. If the rotation angles are in some other convention, they will work just as well, and only the rotation matrix will change. For a comprehensive description of Euler rotations, see [22].

Referring now to Fig. 3, let the three-dimensional Cartesian position of sensor i at time step ℓ be a vector denoted by $\boldsymbol{\xi}_{i,\ell}$. After rotating the sensor coordinate system, the final sensor coordinate system is given by translating the origin of the fully rotated system into the point $\boldsymbol{\xi}_{i,\ell}$.

Now, given $\mathbf{X}_{j,\ell}$—a position vector for target j at time ℓ in the *global* reference frame—the resulting target position in the reference frame of sensor i is given by the inversing transformation

$$\mathbf{x}_{i,j,\ell} = \boldsymbol{\Gamma}_{i,\ell}^{-1}(\mathbf{X}_{j,\ell} - \boldsymbol{\xi}_{i,\ell}). \tag{14}$$

Here, $\boldsymbol{\Gamma}_{i,\ell}$ is the 3×3 rotation matrix that transforms Cartesian points from the fixed global frame to the rotated (but non-translated) reference frame. For the particular rotation convention used in this work, the rotation matrix is given by

$$\boldsymbol{\Gamma}_{i,\ell} = \begin{bmatrix} \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma & -\cos\gamma\sin\alpha - \cos\alpha\cos\beta\sin\gamma & \cos\alpha\sin\beta \\ \cos\alpha\sin\gamma + \cos\beta\cos\gamma\sin\alpha & \cos\alpha\cos\gamma - \cos\beta\sin\alpha\sin\gamma & \sin\alpha\sin\beta \\ -\cos\gamma\sin\beta & \sin\beta\sin\gamma & \cos\beta \end{bmatrix}, \tag{15}$$

where the (i, ℓ) indexing is removed from the angles for convenience. Note that $\boldsymbol{\Gamma}_{i,\ell}$ entirely describes the rotation of sensor i at time ℓ. That is, the poses of all sensors can be recorded as the set of data matrices

$$\mathbf{S} = \{\boldsymbol{\Gamma}_{i,\ell},\ \boldsymbol{\xi}_{i,\ell}\}_{i=1}^{N^s}, \quad \forall \ell. \tag{16}$$

III. LIKELIHOOD EVALUATION

A fundamental feature of the ML-PMHT is that no hard limit is assumed for the maximum number of measurements originating from any one target in any one sensor at any one sample time. This is a significant departure from the assumptions made in data association filters where various one-to-one measurement-to-target assignment events are enumerated and considered. This modeling relaxation allows the ML-PMHT formulation to consider any particular measurement (indexed m) independently from all other measurements, and assign a prior probability mass function over the set of possible measurement generating processes (clutter and all targets)

$$\Pi_{i,\ell} = (\pi_{i,j,\ell})_{j=0}^{N^t}, \quad \text{s.t.} \quad \sum_{j=0}^{N^t} \pi_{i,j,\ell} = 1, \tag{17}$$

where $\pi_{i,j,\ell}$ is the prior probability that any particular measurement in the scan from sensor i originated from process j, and $j = 0$ indicates the clutter process. For now, allow the value of the priors to be ambiguous. Methods for setting the priors are discussed in Section III.A.

The most conveniently scaled statistic to maximize is the log-likelihood ratio (LLR) of the target state $\boldsymbol{\mathcal{X}}^k$ based on the measurement batch $\boldsymbol{\mathcal{Z}}^k$. By definition, the LLR is given by

$$\mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k) = \ln\left(\frac{p(\boldsymbol{\mathcal{Z}}^k|\boldsymbol{\mathcal{X}}^k)}{p(\boldsymbol{\mathcal{Z}}^k|\boldsymbol{\emptyset})}\right), \tag{18}$$

where $p(\boldsymbol{\mathcal{Z}}^k|\boldsymbol{\emptyset})$ represents the probability density function (pdf) of the measurement batch given that no targets are present—the pdf of the entire batch given that everything is clutter-generated. Under the measurement independence assumptions and the product to sum logarithm property, (18) can be written as

$$\mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k) = \sum_{i=1}^{N^s} \sum_{\ell=k-T^b}^{k} \sum_{m=1}^{N_{i,\ell}^z} \ln\left(\frac{p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\mathcal{X}}^k)}{p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\emptyset})}\right), \tag{19}$$

where $N_{i,\ell}^z$ is the number of measurements in the scan of sensor i at time step ℓ. The term $p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\emptyset})$ is the pdf of a single measurement given that it originated from the clutter process. Under the simplifying assumption in (11) and (12), this pdf is uniform in azimuth–elevation space, given by

$$p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\emptyset}) = \frac{1}{V_i} = \frac{1}{W_i^\theta W_i^\phi}, \quad \forall\, \ell, m, \tag{20}$$

where V_i is the total volume of the measurement space of sensor i.

Under the ML-PMHT framework, the term in the numerator on the right-hand side of (19) is given by

$$p(\mathbf{z}_{i,\ell,m}|\boldsymbol{\mathcal{X}}^k) = \pi_{i,0,\ell}\, p(\mathbf{z}_{i,\ell,m}|\emptyset) + \sum_{j=1}^{N^{\mathrm{t}}} \pi_{i,j,\ell}\, p(\mathbf{z}_{i,\ell,m}|\mathbf{X}_{j,\ell}), \tag{21}$$

which is a convex combination of the likelihoods based on the different possible measurement generating processes with the coefficients being the prior probabilities in (17). The term $\mathbf{X}_{j,\ell}$ is the global position of target j at time step ℓ and is given in (1). The relation between $\mathbf{X}_{j,\ell}$ and the batch joint target motion vector $\boldsymbol{\mathcal{X}}^k$ is given by (1) and (2).

Under the Gaussian measurement error assumption, the term $p(\mathbf{z}_{i,\ell,m}|\mathbf{X}_{j,\ell})$ is given by the multivariate Gaussian density

$$p(\mathbf{z}_{i,\ell,m}|\mathbf{X}_{j,\ell}) = \frac{1}{\sqrt{|2\pi\mathbf{R}_{i,\ell}|}} \mathrm{e}^{-0.5(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})^{\mathrm{T}}\mathbf{R}_{i,\ell}^{-1}(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})}. \tag{22}$$

Under the azimuth–elevation measurement model, $\bar{\mathbf{z}}_{i,j,\ell}$—the predicted measurement originating from target j at time step ℓ from sensor i—is given by (9). The sensor reference frame target position $\mathbf{x}_{i,j,\ell}$ required in the measurement prediction is given in terms of the global target position $\mathbf{X}_{j,\ell}$ by (14).

Combining (19)–(22) gives a final expression for the LLR in (23).

$$\mathcal{L}(\boldsymbol{\mathcal{X}}^k; \boldsymbol{\mathcal{Z}}^k) = \sum_{i=1}^{N^{\mathrm{s}}} \sum_{\ell=k-T^{\mathrm{b}}}^{k} \sum_{m=1}^{N_{i,\ell}^{\mathrm{z}}} \ln\left(\pi_{i,0,\ell} + V_i C_{i,\ell} \sum_{j=1}^{N^{\mathrm{t}}} \pi_{i,j,\ell}\, \mathrm{e}^{-0.5(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})^{\mathrm{T}}\mathbf{R}_{i,\ell}^{-1}(\mathbf{z}_{i,\ell,m}-\bar{\mathbf{z}}_{i,j,\ell})} \right). \tag{23}$$

In (23), $C_{i,\ell}$ is the Gaussian constant given by

$$C_{i,\ell} = \frac{1}{\sqrt{|2\pi\mathbf{R}_{i,\ell}|}}. \tag{24}$$

Since the remainder of this work deals with maximizing the LLR over the target motion space given a batch of measurements at time step k, the function in (23) will be denoted as $\mathcal{L}(\boldsymbol{\mathcal{X}}^k)$ for simplicity.

A. Choice of Priors

The performance of the ML-PMHT has been shown in pioneering works to be rather robust to changes in the prior probabilities in (17), (21), and (23). Qualitatively speaking, if it is expected that most of the measurements in any one scan will be clutter, the prior for the clutter process should be significantly higher than the priors for targets.

The simplest option is to naively set the clutter prior based on expected values. If $\lambda_{i,\ell}$ is the expected number of clutter measurements from sensor i at time ℓ and $\varphi_{i,j,\ell}$ is the expected number of measurements originating from target j, the prior for the clutter process can reasonably be set as

$$\pi_{i,0,\ell} = \frac{\lambda_{i,\ell}}{\lambda_{i,\ell} + \sum_j \varphi_{i,j,\ell}}, \tag{25}$$

and the prior for target j as

$$\pi_{i,j,\ell} = \frac{\varphi_{i,j,\ell}}{\lambda_{i,\ell} + \sum_j \varphi_{i,j,\ell}}. \tag{26}$$

The priors also provide a convenient way of working with sensors with restricted fields of view. If, for instance, a target position $\mathbf{x}_{i,j,\ell}$ in the reference frame of sensor i is such that either of the corresponding line-of-sight angles is out of the sensor's angular range (introduced in Section II.A to model a restricted field of view), then the corresponding "predicted observation" is $\bar{\mathbf{z}}_{i,j,k} = \emptyset$. That is, target j is expected to be out of view of sensor i at time ℓ. A convenient way to deal with this is to simply set $\pi_{i,j,\ell} = 0$, and adjust the other priors so that $\sum_j \pi_{i,j,\ell} = 1$.

IV. LIKELIHOOD MAXIMIZATION

As stated earlier in (3), the ML batch estimate up to time k is the $\boldsymbol{\mathcal{X}}^k$ that maximizes $\mathcal{L}(\boldsymbol{\mathcal{X}}^k)$. Since $\mathcal{L}(\boldsymbol{\mathcal{X}}^k)$ is highly non-linear, an analytic solution is not obtainable. However, given a batch of measurements and some fixed $\boldsymbol{\mathcal{X}}_0^k$, it is easy enough to evaluate $\mathcal{L}(\boldsymbol{\mathcal{X}}_0^k)$ using (9), (14), and (23). Therefore, maximization can be done via a hill climbing algorithm combined with other techniques (discussed later) to get within the neighborhood of the global maximum.

Generally speaking, any hill climbing algorithm functions by stepping around a parameter space attempting to find the global maximum in some function of the parameters. Obviously, the hill climber must be started at some initial point in the parameter space. For some applications, it would be perfectly reasonable to sample the initial point from a uniform distribution on the parameter space. In other applications, the measurement space and the parameter space are one and the same. In such cases, one may simply treat some observed measurement as the initial step in the hill climber. There are two main challenges in the likelihood maximization in this work. First, the measurement spaces are not the same as the parameter space. Second, due to clutter, there is not a single hill in the LLR (see Fig. 4). Instead, there are multiple "false" hills (local maxima) along with a single true hill (global maximum). In general, the true hill will be taller than the false hills. However, from the point of view of the hill climber, there is no way to determine how tall a given hill is at the start of climbing. This means that the climber can get stuck climbing the wrong hill. Thus, it would be to our advantage to have a method of initializing the climber as close to the true hill peak as pos-

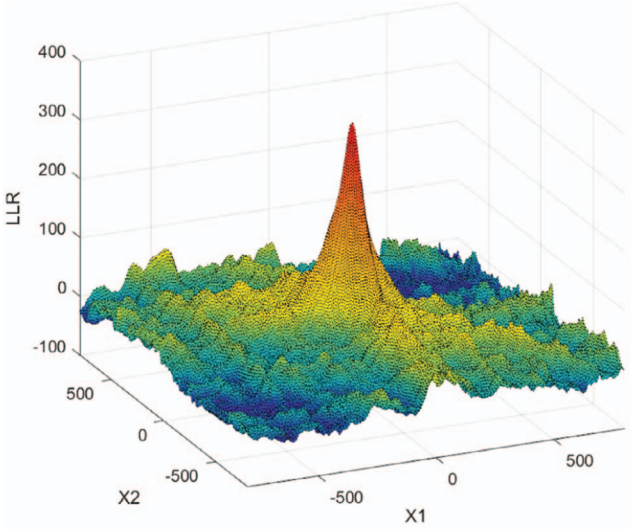


Fig. 4. The LLR surface centered on truth with two out of six dimensions varied (the global X coordinate of both the start and end point of the one target). The true hill (global max) is significantly taller than the rest (local maxima). But since hill climber termination is based on hill slope, the climber could potentially terminate at the top of a false hill. This demonstrates why it is important to initialize the climber as close to the true hill as possible.

sible. This routine will be referred to as "hill finding" in this work. However, "hill finding" is somewhat of a misnomer since the LLR surface has many hills. Precisely speaking, the routine is an attempt to get close (in the parameter space) to the one hill whose peak is the global maximum; the term "hill finding" is used for concision.

A. Hill Finding

The parameter space in this application is described by (2). For a scenario with N^{t} targets, the parameter exists in a $6N^{\mathrm{t}}$ dimensional space. Each six-dimensional sub-space parameterizes a line segment in the three-dimensional global tracking space—three dimensions for the start point and three for the end point. The line segments parameterized in this space will represent the estimate of the corresponding target's trajectory over the course of a batch. With this formulation, a "point" in the parameter space represents a group of N^{t} line segments in the tracking space. Maximizing the likelihood over this parameter space amounts to finding the group of line segments that best represents the target trajectories based the batch of measurements.

"Good" initialization of the hill climber involves setting the initial parameter vector to represent a "good guess" as to the target trajectory segment(s) during the batch. Given some prior knowledge of the targets' behavior, it is possible to predict a parameter to use for initializing the climber. However, for the sake of robustness and to deal with scenarios where no prior information about the targets is available, we have developed a data-driven method of initializing the hill finder. This method, detailed in the rest of this section, obtains a parameter initialization based only on the currently observed batch of measurements. Thus, the tracker can be started "blind"—with no prior information about the targets—and the data-driven hill finding should cause the tracker to converge on the target track(s) within a few batches. Based on the formulation of the parameter space, the ultimate goal of the hill finder—presented in the following, somewhat verbose discussion—is to obtain estimates of target locations in the three-dimensional tracking space at the start and end times of the batch of measurements.

Given the pose of sensor i at time step ℓ, $\{\mathbf{\Gamma}_{i,\ell},\ \boldsymbol{\xi}_{i,\ell}\}$, and a single two-dimensional measurement $\mathbf{z}_{i,\ell,m}$ from that sensor and time, there is not enough information present to solve for a potential target location in three-dimensional global space. There is, however, enough information to define a ray in three-dimensional global space that starts at the origin of the sensor coordinate system and extends infinitely in the line-of-sight direction indicated by the measurement. The azimuth–elevation measurement model is particularly convenient here since it directly gives the azimuth and elevation of this ray in the sensor reference frame.

Now since the algorithm is working with a group of at least two sensors, consider a pair of sensors $\{a, b\}$ at time ℓ and a pair of measurements $\{\mathbf{z}_{a,\ell},\ \mathbf{z}_{b,\ell}\}$—one from each sensor. Then consider the two corresponding line-of-sight rays, one starting at the origin of sensor a extending in the direction indicated by $\mathbf{z}_{a,\ell}$ and the other starting at the origin of sensor b extending in the direction indicated by $\mathbf{z}_{b,\ell}$. In the absence of clutter and measurement error, and given that the two measurements originated from the same target, this pair of rays would provide the precise location of the target as discussed in (for example) [1] and [8] by finding the point where the rays intersect. With additive measurement errors, however, these rays are unlikely to intersect. Furthermore, if one or both of the measurements are clutter originated, or if they do not originate from the same target, then the measurements do not have any meaning when considered as a pair. However, if the measurements in the pair both happen to originate from a target at a point $\mathbf{p}$ in the global space, then the rays indicated by the two measurements should "closely agree" on a point near $\mathbf{p}$, though they will not have an exact intersect due to the measurement errors.

Recall that at sample time ℓ, a single sensor i returns a "scan"—a set of measurements $\mathbf{Z}_{i,\ell}$ that includes all clutter-originated and target-originated measurements. So for each possible pair of sensors, the hill finder routine should check for "hit points" in three-dimensional space that both sensors "closely agree" upon. Obviously not every pair of measurements will "closely agree" on a point since many measurements are clutter-generated and it is not guaranteed that the two sensors are even "looking at" any of the same points (the sensors could, for instance, be placed back-to-back and faced in opposite directions). Thus, in a process of elimination, measurement pairs are subjected to a series of increasingly

strict tests. First of all, the tests must eliminate from consideration pairs of measurements that could not possibly correspond to a single target location in three-dimensional space. Then, pairs that are likely to be unrelated (one or both are clutter originated or they originate from different targets) must be eliminated. And, finally, any measurement pairs that remain must be condensed into composite point measurements in the three-dimensional tracking space—a point $\hat{\mathbf{p}}$ upon which they agree according to some criterion—and the "strength" of agreement should be quantified so that the composite measurements can be objectively ranked in quality.

1) Formation of Composite Point Measurements: This series of tests is presented next for a single measurement pair. The first level of tests are based on computing the closest approach between the pair of line-of-sight rays. This is heuristic but useful for computationally cheap elimination of measurement pairs that are most likely unrelated, which is especially useful in scenarios with large amounts of clutter in each scan. The final, more strict test is based on the iterative least-squares (ILS) estimator.

To formalize, consider a particular pair of sensors $\{a, b\}$, $a \neq b$, at a particular time ℓ. Take the pair of scans $\{\mathbf{Z}_{a,\ell},\ \mathbf{Z}_{b,\ell}\}$, and let

$$N^{z}_{i,\ell} = |\mathbf{Z}_{i,\ell}|, \qquad i = a, b, \tag{27}$$

be the number of individual measurements in the scans.

Now consider some pair of measurements (one from each scan)

$$\{\mathbf{z}_{a,\ell,m},\ \mathbf{z}_{b,\ell,n}\}, \quad m \in \{1, \ldots, N^{z}_{a,\ell}\},\ n \in \{1, \ldots, N^{z}_{b,\ell}\}. \tag{28}$$

Let us momentarily drop the time (ℓ) and measurement pair (m, n) indexing and simply consider a particular measurement from sensor a, call it $\mathbf{z}_a$, a particular measurement from sensor b, call it $\mathbf{z}_b$, with both measurements taken at the same time. Each measurement can be taken to represent a ray (half line). So the pair of measurements yields a pair of rays parameterized by a pair of origin points and a pair of unit vectors that indicate the rays' pointing directions *in the global coordinate system*.

The origin of the first ray is $\boldsymbol{\xi}_a$—the location of sensor a, which is assumed to be known. Similarly, the second ray has origin point $\boldsymbol{\xi}_b$. The direction vectors are found by first obtaining the unit vectors in their respective sensor reference frames and transforming them both into the global reference frame with the known sensor orientations. In the case of azimuth–elevation measurements, the unit direction vectors are given by

$$\mathbf{v}_i = \boldsymbol{\Gamma}_i \begin{bmatrix} \sin\phi_i \\ \sin\theta_i \cos\phi_i \\ \cos\theta_i \cos\phi_i \end{bmatrix} \qquad i = a, b, \tag{29}$$

where θ_i, ϕ_i are the individual components of $\mathbf{z}_i$ and $\boldsymbol{\Gamma}_i$ is the rotation matrix of the ith sensor pose.

For any non-parallel pair of rays in three-dimensional space, there is a single line segment somewhere that connects the rays and is perpendicular to both rays. The end points of this line segment are the points of closest approach of the rays, and the length is the minimum distance between the rays [19]. For this case, solve for the end points of the minimum distance segment with

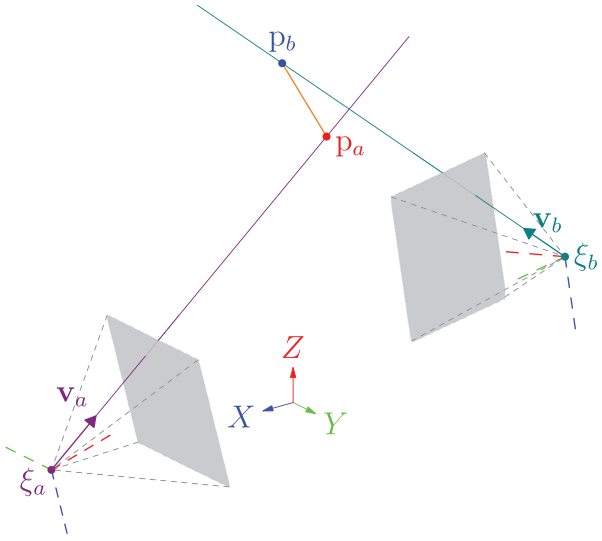


Fig. 5. Notional illustration of the points of closest approach (red and blue points) between two line-of-sight measurements.

$$\mathbf{p}^{\mathrm{md}}_i = \rho_i \mathbf{v}_i + \boldsymbol{\xi}_i, \quad i = a, b, \tag{30}$$

where $\mathbf{p}^{\mathrm{md}}_i$ is the location of the closest approach that lies on ray i. The scalar values ρ_a, ρ_b determine the distance along each ray at which the closest approach occurs. Let

$$\mathbf{c} \triangleq \boldsymbol{\xi}_b - \boldsymbol{\xi}_a, \tag{31}$$

then these scalar values are given by

$$\rho_a = \frac{-(\mathbf{v}_a \cdot \mathbf{v}_b)(\mathbf{v}_b \cdot \mathbf{c}) + (\mathbf{v}_a \cdot \mathbf{c})}{1 - (\mathbf{v}_a \cdot \mathbf{v}_b)^2}, \tag{32}$$

$$\rho_b = \frac{(\mathbf{v}_a \cdot \mathbf{v}_b)(\mathbf{v}_a \cdot \mathbf{c}) - (\mathbf{v}_b \cdot \mathbf{c})}{1 - (\mathbf{v}_a \cdot \mathbf{v}_b)^2}, \tag{33}$$

where the dot indicates a vector dot product. Both values are defined as long as the two measurement rays are not perfectly parallel, which happens with probability zero. Refer to Fig. 5 for an illustration of the closest approach between an example pair of measurements.

First, a test can be performed by considering just the signs of the scalar values ρ_a, ρ_b. These are the Cartesian distances along the rays where the closest approach points occur. Thus, if either ρ_a or ρ_b is negative, it means the corresponding closest approach point occurs "behind the sensor." (It is assumed that the negative z half of the sensor coordinate frame is never observable.) So, if either value is negative, reject the corresponding pair of measurements as being indicative of a "hit" on a target.

Furthermore, in some settings, it would be reasonable to set maximum limits for the values ρ_a, ρ_b such that if either value exceeds its maximum, the pair of

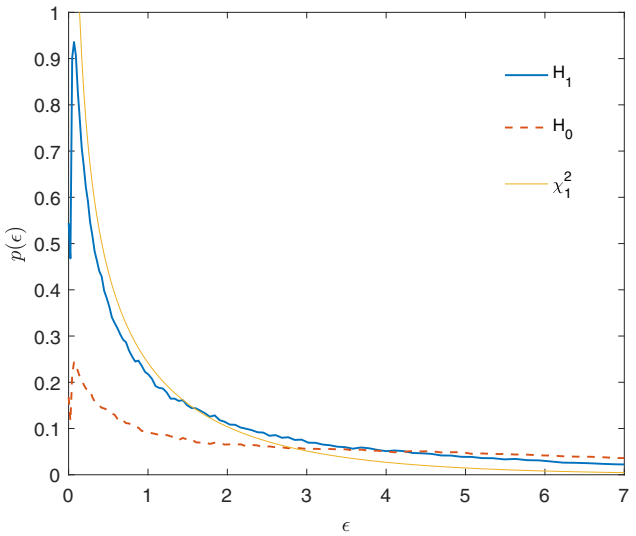


Fig. 6. The empirical distributions of the normalized measurement error squared under the binary hypotheses: H_0: the measurements used in the ILS estimate are unrelated and H_1: the measurements used in the ILS estimate originated from the same target. The one-DOF Chi-squared distribution is also plotted for reference. The discrepancy between the theoretical Chi-squared distribution and empirical distribution under H_1 is due to the thresholded termination of the ILS. The distributions would match if the ILS was run to exact termination with perfect numerical precision.

measurements is eliminated. For instance, if visibility is such that the user knows that no sensor can see farther than 10 000 m, and two measurement rays have a closest approach point that is 15 000 m away from one of the sensors, then that pair of measurements could also reasonably be rejected as originating from a common target. This would also be of use if it is known that the target tracking space down range of one or both sensors has a hard limit—e.g., the sensors are orbiting the earth at some known altitude and are pointed toward the earth's surface.

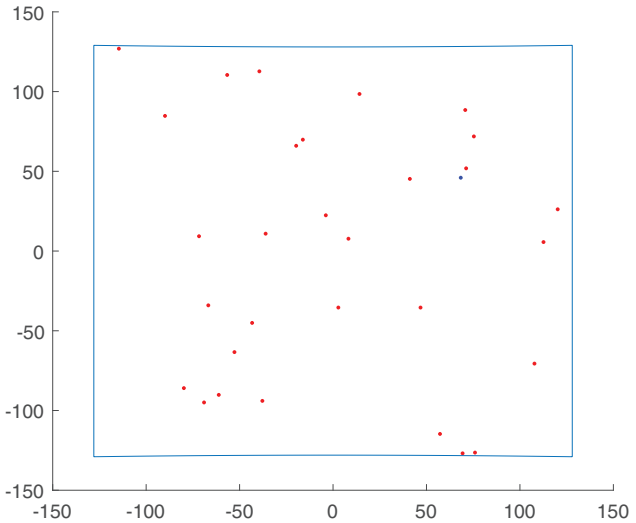


Fig. 7. Example camera view of hits for a single scan for a clutter level of $\lambda = 30$.

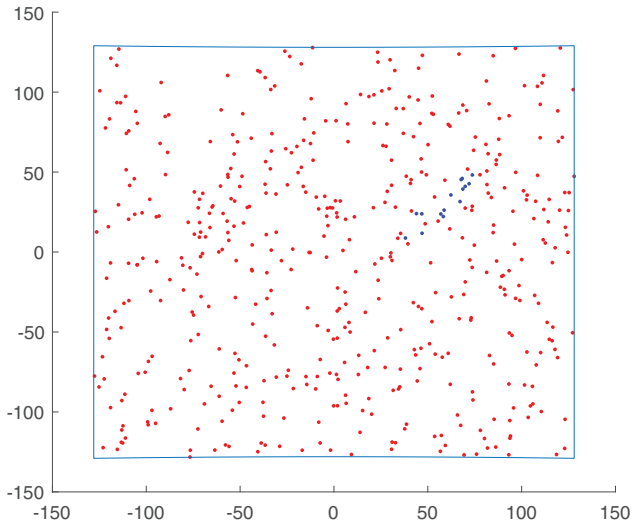


Fig. 8. Example camera view of superimposed scans for an entire batch for a clutter level of $\lambda = 30$. Blue dots indicate true target hits and red dots are clutter. Note that without the color coding, it is not clearly evident to the human eye where the target track is located. The blue outline shows the image edge that results from setting limits on the azimuth and elevation angles.

A final heuristic test involves setting a threshold τ_{md} on the length of the minimum distance segment such that if

$$||\mathbf{p}_a^{\text{md}} - \mathbf{p}_b^{\text{md}}||_2 \geq \tau_{\text{md}}, \tag{34}$$

then the measurement pair is eliminated from consideration. The ideal value for τ_{md} depends on the particular scenario and the desired level of restraint in eliminating measurement pairs. Qualitatively speaking, the farther down range the targets are expected to appear, the greater the τ_{md} should be. And, if it is found that the algorithm is considering more measurement pairs than the user finds reasonable, then τ_{md} can be decreased. This is perhaps the most heuristic of the series of tests. However, in simulated scenarios with large amounts of clutter, and sensors, which are known to be observing a common space in which the targets are known to exist, it has been found to be the workhorse test that eliminates all but the most likely-to-be-related measurement pairs.

The tests presented thus far serve to eliminate line-of-sight measurement pairs that either do not point toward a common space, point at a space that is physically too far away for the sensors to observe, point at a space where targets are not likely to exist, or are likely unrelated based on not approaching each other within a reasonably constrained space. Now, any measurement pairs that remain must be either eliminated as well or consolidated into a single composite measurement point in three-dimensional space. Suppose that the measurement pair under consideration—$\{\mathbf{z}_a, \mathbf{z}_b\}$—has passed the simple tests involving the closest approach between the lines-of-sight. If those tests were reasonably well-tuned to the operating scenario, the fact that the measurement pair has passed increases the likelihood that both

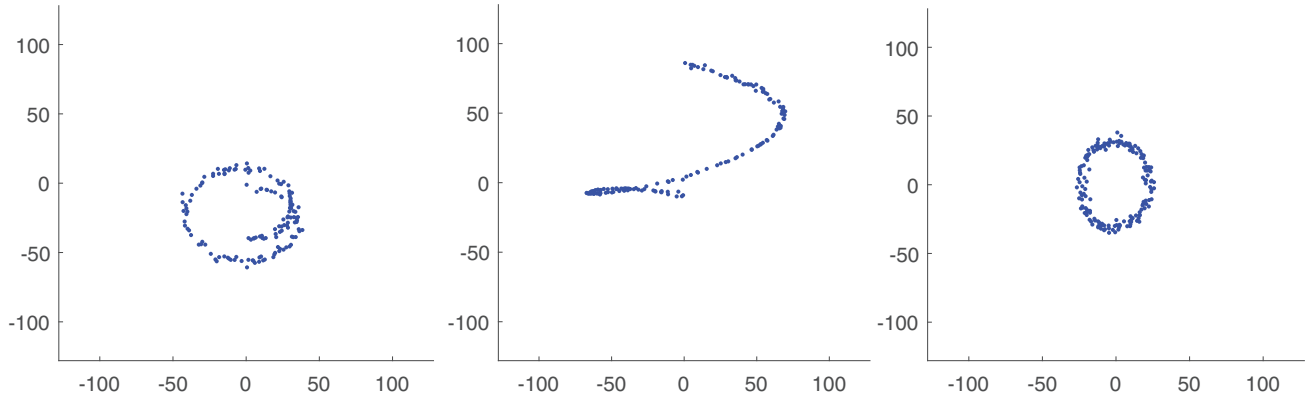


Fig. 9. The true hits received by each camera over the course of the entire scenario. Missed detections are accounted for (not visible).

measurements in the pair originated from a common target. Thus, a composite point measurement in the three-dimensional tracking space based on the pair of measurements is likely to be "meaningful"—an estimate of a true target position.

The ILS estimator provides a mathematically rigorous method to form a composite point measurement based on the pair of line-of-sight measurements. It functions by iteratively solving for the point $\hat{\mathbf{p}}$ in the three-dimensional tracking space, which minimizes the normalized measurement error squared implied by the measurement pair due to a target at point $\hat{\mathbf{p}}$, which is a scalar value given by

$$\epsilon = \sum_{i=a,b} (\mathbf{z}_i - \bar{\mathbf{z}}_i(\hat{\mathbf{p}}))' \mathbf{R}_i^{-1} (\mathbf{z}_i - \bar{\mathbf{z}}_i(\hat{\mathbf{p}})). \tag{35}$$

where $(\cdot)'$ denotes the matrix transpose. The term $\bar{\mathbf{z}}_i(\hat{\mathbf{p}})$ is the zero-error measurement returned by sensor i due to a target at $\hat{\mathbf{p}}$. Notice the implicit assumption that both of the measurements in the pair have in fact originated from the same target—something that cannot be known for certain in this case. However, due to the previous series of tests based on the closest approach points, the ILS estimate will be calculated only for likely-to-be-related measurement pairs. Furthermore, the final minimized value of ϵ will be used as a final test statistic to eliminate all but the strongest composite measurements.

The ILS estimator using line-of-sight measurements in azimuth–elevation form is given in [13], and used for related work in [9]. It is summarized here. First, form a 4×1 vector by stacking the pair of measurement column vectors

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_a \\ \mathbf{z}_b \end{bmatrix} \tag{36}$$

and form the corresponding 4×4 covariance matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_b \end{bmatrix}, \tag{37}$$

where $\mathbf{0}$ is a 2×2 matrix of zeros. Then $\hat{\mathbf{p}}_q$—the ILS estimate at the qth iteration—is updated with an additive term as

$$\hat{\mathbf{p}}_{q+1} = \hat{\mathbf{p}}_q + \boldsymbol{\Delta}_q, \tag{38}$$

where the additive update term is calculated as

$$\boldsymbol{\Delta}_q = \left[\mathbf{H}_q' \mathbf{R}^{-1} \mathbf{H}_q \right]^{-1} \mathbf{H}_q' \mathbf{R}^{-1} \left[\mathbf{z} - \bar{\mathbf{z}}(\hat{\mathbf{p}}_q) \right], \tag{39}$$

where the 4×1 zero-error measurement vector is given by stacking the individual zero-error measurement vectors as

$$\bar{\mathbf{z}}(\hat{\mathbf{p}}_q) = \begin{bmatrix} \bar{\mathbf{z}}_a(\hat{\mathbf{p}}_q) \\ \bar{\mathbf{z}}_b(\hat{\mathbf{p}}_q) \end{bmatrix}, \tag{40}$$

in which the individual vectors $\bar{\mathbf{z}}_i(\hat{\mathbf{p}}_q)$ are obtained by first setting $\mathbf{X}_{j,\ell} = \hat{\mathbf{p}}_q$ in (14), and then substituting the result into (9). The matrix $\mathbf{H}_q$, defined as

$$\mathbf{H}_q = \left. \frac{\partial \bar{\mathbf{z}}(\mathbf{X})}{\partial \mathbf{X}} \right|_{\mathbf{X}=\hat{\mathbf{p}}_q}, \tag{41}$$

is the 4×3 Jacobian matrix of the stacked zero-error measurement vector with respect to Cartesian position in the *global reference frame*, evaluated at the current ILS estimate.

The formulas for the individual elements of the Jacobian matrix and the initialization of the ILS estimator are given in the Appendix.

The ILS estimator is terminated by setting a threshold τ_Δ and iterating until

$$||\boldsymbol{\Delta}_q||_2 \leq \tau_\Delta \tag{42}$$

and recording the final estimate as

$$\hat{\mathbf{p}} = \hat{\mathbf{p}}_{q+1}. \tag{43}$$

Once the final ILS estimate is obtained, the minimum normalized measurement error squared given by (35) is theoretically Chi-squared distributed with one DOF *given that the pair of measurements used in the estimate originated from a common target* [4]. If the pair of measurements used in the estimate is unrelated despite having passed the previous tests, the distribution of the minimum normalized measurement error has no known closed form. However, the distribution is obtained empirically through simulation and plotted in Fig. 6. The

shape of this distribution is much like the Chi-squared one DOF distribution, but with a much heavier tail. This allows for a final thresholded test using ϵ—calculated with the formula in (35)—as the test statistic. That is, if the value of ϵ resulting from the ILS estimate based on the measurement pair under consideration is above some threshold τ_ϵ, then the pair and the resulting composite measurement can be eliminated from consideration. Otherwise, the composite measurement has passed all elimination tests and is taken to correspond to an actual target. A composite measurement that passes this final test can be assigned a quantitative score given by

$$\sigma = 1 - \chi_1^2(\epsilon) \tag{44}$$

—the complement of the one DOF Chi-squared cumulative distribution function (cdf) evaluated at the minimized normalized measurement error squared. This will provide a score in the range $[0, 1]$, with better estimates receiving higher scores.

The elimination tests above were presented in terms of a single pair of measurements from a particular pair of sensors (a, b) at a single time step, and the indexes of the measurement pair (m, n) and time step (ℓ) were omitted throughout. Now, suppose that the algorithm performs the entire series of elimination tests for every element of the set

$$\{\{\mathbf{z}_{a,\ell,m},\ \mathbf{z}_{b,\ell,n}\}\ :\ 1 \le m \le N^{\mathrm{z}}_{a,\ell} \quad 1 \le n \le N^{\mathrm{z}}_{b,\ell}\}$$

—every possible pair of measurements from the scans of sensors a and b at time ℓ. Record to memory all the resulting composite measurement points that have passed all the elimination tests along with their corresponding Chi-squared scores given by (44) as the set of parameter pairs

$$\tilde{\mathbf{P}}_{(a,b),\ell} = \left\{(\hat{\mathbf{p}}, \sigma)_p\right\}_{p=1}^{N^{\mathrm{p}}_{(a,b),\ell}}, \tag{45}$$

where $N^{\mathrm{p}}_{(a,b),\ell}$ is the number of composite measurements from sensor pair (a, b) at time step ℓ that have not been eliminated.

Notice that along with the obvious dependence on time index ℓ, $\tilde{\mathbf{P}}_{(a,b),\ell}$ is also dependent on the *pair* (a, b). That is, the list of points obtained at a particular time depends on which pair of sensors is being used to look for points. With a group of $N^{\mathrm{t}} > 2$ sensors, find the list of hit points from each possible pair of sensors and form the total list

$$\mathcal{P}_\ell = \bigcup_{(a,b)\in\mathbf{r}} \tilde{\mathbf{P}}_{(a,b),\ell}, \tag{46}$$

where $\mathbf{r}$ is the set of all unordered pairs of the integers $[1, N^{\mathrm{s}}]$.

2) Formation of Initial Parameter Estimates: To form initial parameter vectors to pass to the climber at estimation time step k, suppose that $\mathcal{P}_\ell$ has been obtained for every time step $\ell \le k$. Potential target track segments over the current batch are formed by pairing composite measurements from the current batch start time in $\mathcal{P}_{k-N^{\mathrm{b}}+1}$ with composite measurements from the current batch "leading edge" time in $\mathcal{P}_k$, which parameterizes line segments in global three-dimensional space. If the user has knowledge of a minimum and/or maximum possible target velocity (in units length/sampling period), then it can be used here to select only the reasonable potential track segments. The segments are also assigned a score equal to the product of the Chi-squared score of the composite measurements that make up its end points. This is useful in quantitatively ranking which segments are the best if the number of segments sent to the hill climber must be limited.

Formally, let $s = k - N^{\mathrm{b}} + 1$ be the batch start time step, and form the set of parameter pairs given in (47),

$$\begin{aligned}\mathcal{Q}^k = \Bigg\{ \Bigg(\mathbf{Q} = \begin{bmatrix} \hat{\mathbf{p}}_{s,p} \\ \hat{\mathbf{p}}_{k,q} \end{bmatrix},\ \delta = \sigma_{s,p}\sigma_{k,q} \Bigg) \\ :\ v_{\min} T^{\mathrm{b}} \le ||\hat{\mathbf{p}}_{s,p} - \hat{\mathbf{p}}_{k,q}||_2 \le v_{\max} T^{\mathrm{b}}, \\ 1 \le p \le |\mathcal{P}_s|, \quad 1 \le q \le |\mathcal{P}_k| \Bigg\},\end{aligned} \tag{47}$$

where $\hat{\mathbf{p}}_{s,p}$ is the pth composite measurement in the set of composite measurements from the batch start time step s, and $\sigma_{s,p}$ is the measurement's Chi-squared score. Similarly, $\hat{\mathbf{p}}_{k,q}$, $\sigma_{k,q}$ are the individual members of the qth element in the set of composite measurements from the batch leading edge time step k. Let $[v_{\min},\ v_{\max}]$ represent the range of possible target velocities and $T^{\mathrm{b}} = N^{\mathrm{b}} - 1$ be the *duration* of the batch in sampling periods. Forgive the reuse of the index q—it was used in Section IV.A.1 for an unrelated purpose.

To be verbose, each element of the set $\mathcal{Q}^k$ contains a 6×1 column vector parameterizing a line segment in three-dimensional space that satisfies a length restriction, paired with a scoring value that represents the "target indication strength" of the segment based on the Chi-squared scores of the two composite point measurements that parameterize the segment. Notice that only composite measurements from the start and end time steps of the batch are used to populate $\mathcal{Q}^k$. If the user finds that this does not provide enough target track segment estimates, then pairs of composite measurements from intermediate batch times can be used to form segments that are projected to the start and end batch times. That is, select pairs of composite measurements, one from time step t and one from time step ℓ, such that $s \le t < \ell \le k$ and—based on the assumption of a constant sampling period—compute the individual 3×1 components of the stacked vector $\mathbf{Q}$ in (47) as

$$\hat{\mathbf{p}}_s = \hat{\mathbf{p}}_\ell + \left(\frac{\ell - s}{\ell - t}\right)(\hat{\mathbf{p}}_t - \hat{\mathbf{p}}_\ell), \tag{48}$$

$$\hat{\mathbf{p}}_k = \hat{\mathbf{p}}_t + \left(\frac{k - t}{\ell - t}\right)(\hat{\mathbf{p}}_\ell - \hat{\mathbf{p}}_t), \tag{49}$$

and assign the projected segment a score equal to the product of the Chi-squared scores of the two composite measurements used in the calculation. This segment projection is particularly useful in scenarios where all sensors have a low probability of target detection in any one scan since, for there to exist a meaningful composite measurement at any one time step, at least two sensors will need to have detected the same target at that sampling instant.

The likelihood maximization will assume the number of targets based on the size of the initial parameter vector it is passed. In order to choose a good initial vector that parameterizes N^{t} targets, sample N^{t} segments from $\mathcal{Q}^k$ *without replacement*, and stack the individual segment vectors into a $6N^{\mathrm{t}} \times 1$ parameter vector of the form given in (2). The sampling should give preference to segments with high δ scores. A decent way to accomplish this is to use a high-level function like MATLAB's datasample() and sample element r of $\mathcal{Q}^k$ with the weight of δ_r relative to all other δ values. If the number of segments in $\mathcal{Q}^k$ is small enough that it will not be computationally prohibitive to simply run a new instance of the hill climber for every possible initial parameter vector, then this sampling can be avoided altogether.

Thus far, this section has discussed a method of using observed data to initialize the hill climber. This has been found to perform well enough as the sole method of hill climber initialization at the beginning of each batch. However, performance can be improved further if the target motion for the current batch is predicted from the best previous batch estimate (under the constant velocity assumption). The predicted segments along with segments sampled from $\mathcal{Q}^k$ are then each used to initialize individual instances of the hill climber. In simulation, it was found that this prediction aids the tracker in "sticking" to the track once it has a good estimate. On the other hand, it cannot be used as the only method of initializing the hill climber since the data-driven hill finder is needed to converge on a good estimate in the first place and to recover if the track is ever lost.

B. Hill Climbing

There are many types and variations of hill climbers. The maximization in this work uses the conjugate gradient method implemented in Python's SciPy "optimize" library [24]. It should be pointed out that an elegant expectation-maximization (EM) approach could also be used (see [2]).

In some cases, the user might be interested in deciding whether there is even a single target present or not (target detection). In this case, one could pass the hill climber an initial vector $\mathcal{X}_0^k$ that implies a single target (six elements), and allow the climber to run till it reaches a peak. Then, compare the value of the LLR at this peak to some threshold to decide if there is a target present. That is, if the peak of hill is below some height, decide that the hill is just due to a randomly occurring pattern in clutter instead of an actual target. The challenge with this is picking a good threshold for this test. The peak height of a given LLR hill depends heavily on the number of sensors being used and the geometric arrangement of the sensors relative to each other and the targets. The simulated detection performance is discussed in Section V.B.

C. Tracker Summary

Tracking is performed in a "sliding batch" fashion where after an estimate is obtained, the leading edge of the batch slides forward by some number of sampling periods less than the length of the batch. This means that consecutive batch estimates share some observations and are thus correlated. The following will give a step-by-step synopsis of the algorithm:

1) At time step k, take the N^{b} most recent scans from each of the N^{s} sensors to form the current measurement batch. That is, k indicates the leading edge of the batch.
2) For each time step in the batch, for each possible *pair* of sensors, obtain via the process of elimination in Section IV.A.1 pairs of measurements from the pair of sensors that "strongly agree" on some point in the global space. Record the resulting composite measurements.
3) With composite measurements obtained in the previous step, form line segments representing target tracks over the course of the batch. These segments are parameterized by their start and end points. They can be formed either by pairing hit points from the start and end times of the batch or by considering pairs of hit points from intermediate time steps and projecting out to the end points of the batch. If information is available about maximum and/or minimum target speed, this can be used to eliminate segments that are either too long or too short.
4) If working with a multi-target scenario with N^{t} targets, then form potential target parameters by combining line segments from the previous step into groups of size N^{t}. In a single target scenario, any one of the line segments can be taken as a potential target parameter.
5) (Optional) Form a predicted target parameter by using a constant velocity assumption to predict the target trajectories over the current batch based on the ML estimate from the previous batch.
6) For each of the parameter vectors obtained in 4) and 5), evaluate the LLR. Keep the N best parameters according the LLR value.
7) Initialize N hill climbers with each of the N best parameters from the previous step. Allow climbers to run until terminal condition or until some maximum number of steps has been exceeded. Take the param-

eter indicated by the climber that reached the highest peak in LLR to be the ML estimate for the current batch.

8) Increment k by the desired batch slide amount. Return to 1).

It is of interest to summarize the computational complexity of the method for a single batch estimate. As presented, the expected computational complexity of the hill finding is $\mathcal{O}\left((\lambda + p_{\mathrm{d}}N^{\mathrm{t}})^2 \frac{N^{\mathrm{s}}(N^{\mathrm{s}}-1)}{2}\right)$, where $\lambda + p_{\mathrm{d}}N^{\mathrm{t}}$ is the expected total number of measurements per scan (clutter plus targets) and $\frac{N^{\mathrm{s}}(N^{\mathrm{s}}-1)}{2}$ is the number of unique sensor pairs. That is, the point finding (optimization-initialization, or hill finding) cost is quadratic in both the number of measurements per scan and in the number of sensors. While the expected number of measurements is usually dictated by "nature," settings with many sensors may require care when choosing which sensor pairs are used to find points.

The evaluation of the likelihood in (23) has expected complexity $\mathcal{O}\left((\lambda + p_{\mathrm{d}}N^{\mathrm{t}})N^{\mathrm{s}}N^{\mathrm{t}}\right)$. The likelihood maximization requires approximation of the gradient of the objective likelihood function via two-point differencing, which requires $6N^{\mathrm{t}}+1$ evaluations of the objective function (one for each element of the parameter vector plus a reference evaluation). Thus, the overall likelihood maximization has complexity $\mathcal{O}\left(\left[\lambda(N^{\mathrm{t}})^2 + p_{\mathrm{d}}(N^{\mathrm{t}})^3\right]N^{\mathrm{s}}\right)$. The number of targets assumed by the optimization is the most significant factor in computational cost. While $(N^{\mathrm{t}})^3$ is the asymptotically dominant term, the entire expression $\lambda(N^{\mathrm{t}})^2 + p_{\mathrm{d}}(N^{\mathrm{t}})^3$ is noted since, in most practical settings, $\lambda \gg p_{\mathrm{d}}N^{\mathrm{t}}$ (there is usually much more clutter than target-originated measurements), so the clutter level can dominate practical computation. Both the hill finder and maximizer also have simple linear complexity in the batch length N^{b}.

V. RESULTS

The following sections present simulated results for the sake of testing the presented method. The detection performance (performance of a test to decide whether or not a target is present) is given in Section V.B. The tracking performance for a single target under various states of nature is studied in Section V.C. In Section V.D, a comparison is made to a decentralized method in which each sensor obtains an ML estimate of the track segment in its own measurement space, and then batch estimates from pairs of sensors are fused to obtain track segment estimates in the global space. This is in contrast to the method presented in the main body of this work could be considered a fuse-before-track method since the likelihood involves measurements from all sensors. Finally, a scenario with 2 targets is simulated, and a comparison is made between the tracking performance of the presented method and the JPDA method presented in [20].

A. Single Target Scenario Setup

The scenario has a single target and three sensors. The target travels along an upward spiraling path at a constant speed, see Fig. 9. Notice that, technically, the curvature of the path violates the constant velocity assumption in the target motion model. However, due to the constant speed of the target and the small amount of curvature that occurs over the course of any one batch, constant velocity is a sufficient approximation.

Each simulated sensor has a restricted, conical field of view with an angular range of 20°. The sensors are placed so that, for the majority of the target's trajectory, it is in view of all sensors. The sensor locations are all out of view in Fig. 11, but referring to the coordinate system in the figure:

- Sensor 1 is stationary at position (−18000,0,0), and aimed in the ^{+}X direction (toward the origin) and slightly up. This mimics a camera viewing the sky from the surface of the earth.
- Sensor 2 is stationary at position (0,0,60000), and aimed toward the origin. This gives an overhead view from a very high altitude.
- Sensor 3 is in motion. It orbits the Z axis at a height of 12 000 above the X–Y plane and with a radius of 10 000. Its orientation changes so that it is consistently aimed down and in toward the origin. Its orbital speed is such that it makes only half an orbit throughout the scenario.

In a real scenario, the expected number of clutter λ and the probability of target detection p_{d} are functions of SNR and the threshold used in the measurement extraction process. Qualitatively speaking, at a fixed SNR, lower extraction threshold values will result in more clutter but higher probability of target detection.

For simulation purposes, it is assumed that the measurement extraction process is Gaussian as in [11]. That is, assume some extraction threshold τ_{me}, which, when applied to a test for a measurement "hit" in some particular section of the sensor space where a target is not present, results in a single clutter measurement with a probability of "false alarm" given by

$$p_{\mathrm{f}} = \mathrm{Q}\left(\frac{\tau_{\mathrm{me}}}{\sigma}\right), \tag{50}$$

where $\mathrm{Q}(\cdot)$ is the Gaussian Q function and τ_{me} has been normalized by the measurement intensity standard deviation σ. Then, for a fixed SNR, the probability of target detection is

$$p_{\mathrm{d}} = \mathrm{Q}(\mathrm{Q}^{-1}(p_{\mathrm{f}}) - \mathrm{SNR}). \tag{51}$$

For the sake of simulation, it is assumed that there are a Poisson random number of clutter measurements in each scan with expected value λ, and the expected number is approximately related to the clutter level by

$$p_{\mathrm{f}} = \frac{\lambda}{N_{\mathrm{c}}}, \tag{52}$$

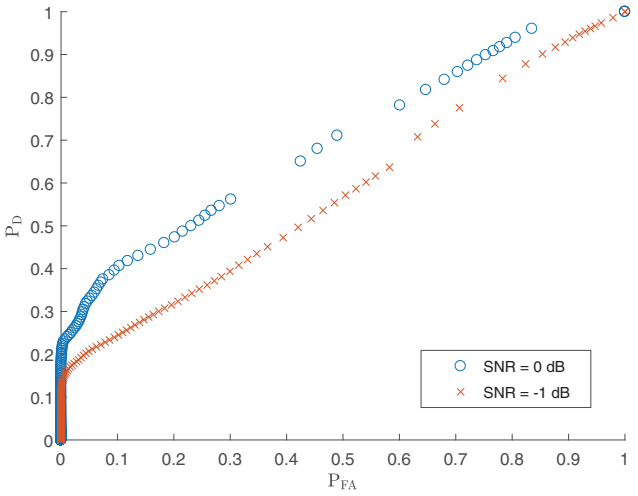


Fig. 10. The ROC curves for target detection based on the ML-PMHT likelihood at different SNR levels. The probability of target detection at individual sensors is fixed at $p_d = 0.45$. This implies a clutter level of $\lambda = \sim 54$ at an SNR of 0 dB and $\lambda = \sim 74$ at an SNR of -1 dB

where N_c represents some total number of resolved "cells" in the sensor space that are tested by the measurement extraction. It is assumed here that all sensors have $N_c = 400$.

The simulation uses azimuth–elevation measurements. Target-originated measurements have Gaussian additive error where the individual components are assumed independent and each has standard deviation $\sigma_\theta = \sigma_\phi = 0.25^\circ$. The small measurement error variance is used to mimic the behavior of cameras, which generally exhibit good measurement accuracy. The other "error-inducing" processes (clutter and missed detections) are considered more interesting in this context.

Unless otherwise noted, results use a fixed batch size $N^b = 18$, and the batch slides five sampling periods from one estimate to the next.

B. Detection Performance

It is of interest how well the algorithm does at detecting the presence of a target. Detection would be performed by choosing some threshold value and declaring a detection if the LLR for the current batch exceeds the threshold. To test the detection performance, the algorithm is run for 100 Monte Carlo runs under the null hypothesis—no target present. These data are combined with the data from the simulation with a target present to form the receiver operating characteristic (ROC) curves shown in Fig. 10. When the target is present, each sensor measures it with fixed probability of detection $p_d = 0.45$.

At first glance, the detection performance appears poor. However, it is important to note that detection is performed on a batch-to-batch basis. That is, if the detection was operating at a point on the ROC with $P_D = 0.3$ and $P_{FA} = 0.03$, then the user could expect to get a detection within four batches of the target appearing,

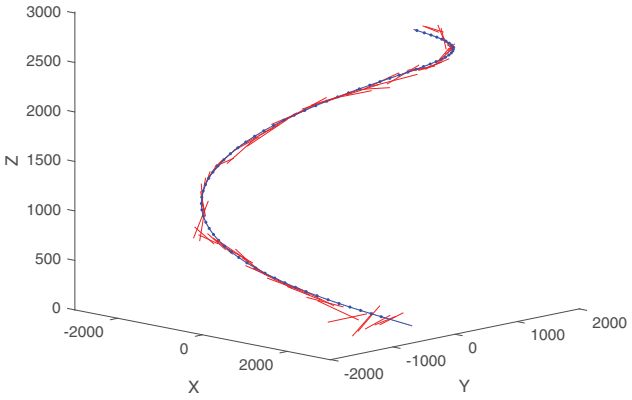


Fig. 11. The true target track (blue) plotted along with the batch estimates (red) for a single Monte Carlo run with $\lambda = 54$ and $p_d = 0.45$.

while expecting a false detection only once in every ~33 batches when there is no target present. When framed in this manner, the detection performance is acceptable. It is also important to emphasize that the detection and false alarm probabilities on the axes of Fig. 10 are not the same as the detection and false alarm probabilities in (50) and (51), which are properties of the underlying measurement extraction process.

C. Tracking Performance

When tracking with a single target present, the effect of the hill finder can be seen in the first few ML batch estimates in Fig. 11. While the first few estimates are not on track, the hill finder enables the tracker to converge on the track within a handful of batches. If instead the hill climbing was initialized randomly throughout the entire tracking space, it would have a tendency to settle on false hills instead of converging to the true track. There is a point about one third of the way into the scenario where the algorithm briefly does a poor job tracking the target. This is due to the geometry of the target relative to the sensors being less than ideal at that time—two sensors have almost anti-parallel lines-of-sight on the target. The resulting deviation in the estimate can be seen in Fig. 11 at the spot where the batch estimates drift away from the true track and in Fig. 12 by the spike in estimation error around time step 120. The spike in error is short-lived, however, since the hill finder compensates as soon as the sensors have good visibility on the target again.

We wish to also study the performance of the algorithm in terms of how often it is on track. To do so, we must first quantify what it means to be "on track." One way to define "on track" is to find the root mean square error (RMSE) over the course of each batch. Then, if the RMSE over the course of a particular batch is below some threshold, declare the algorithm to be on track for that batch. It is of interest to study the tracking performance based on the operating characteristic of the measurement extraction process. The SNR is fixed and sim-

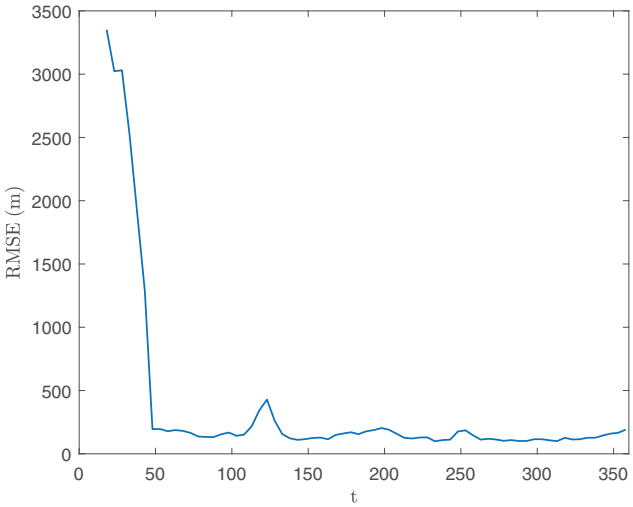


Fig. 12. The RMS distance error between the estimate and true target position versus time step. Averaged over 100 MC runs with $\lambda = 54$ and $p_{\mathrm{d}} = 0.45$.

ulations are performed over a range of normalized measurement extraction threshold values, which is equivalent to a range of clutter levels via (50) and (52). The empirical probability of the algorithm being on track (denoted P_{T}) is plotted for three different SNR values in Fig. 13. The RMSE threshold used to declare whether the algorithm is on track or not is set at 200 using the results in Fig. 12—slightly higher than the RMSE to which the algorithm empirically converges.

For each plot, a peak in tracking performance appears, above which the decreasing probability of target detection has a negative effect on the hill finder, and below which the increasing level of clutter results in decreasing estimation quality. The trend in the horizontal location of the peaks suggests that the lower the SNR of

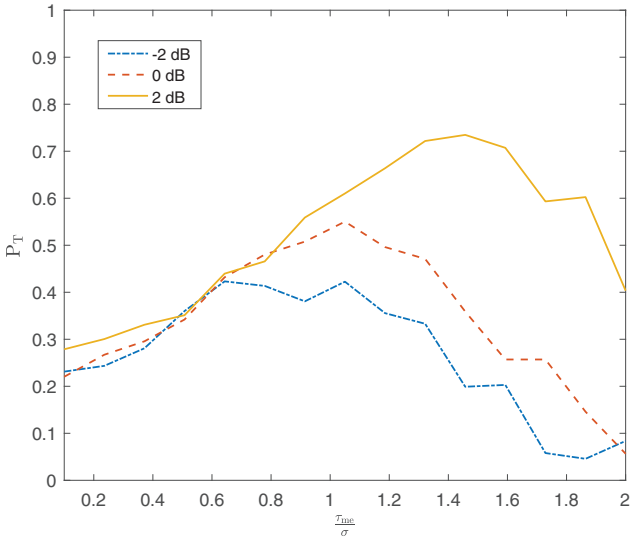


Fig. 13. The probability of being on track versus the normalized measurement extraction threshold. Different plots are for various measurement extraction SNR values. Results are averaged over 20 Monte Carlo runs of the scenario—or 1400 batches—for each data point.

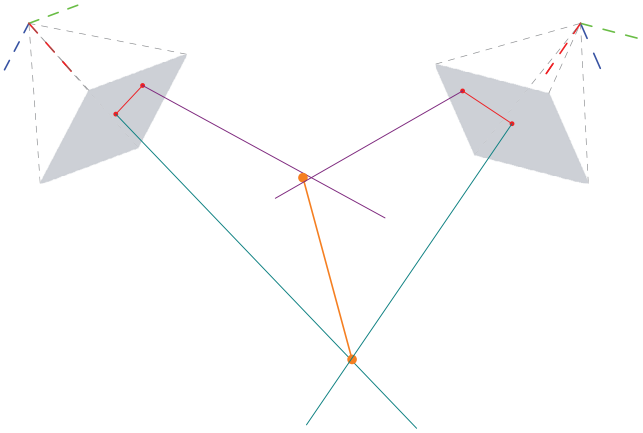


Fig. 14. Notional representation of fusing two batch estimates from two separate cameras (red segments) into a single estimate in global space (orange segment).

the tracking scenario, the lower the ideal measurement extraction threshold. That is, in scenarios with lower target visibility, it is better to operate in a "high clutter/high detection probability" regime. And when targets are more visible, it is ideal to compromise on the detection probability with the pay-off of having less clutter. The vertical location of the peaks are perhaps less informative since the probability of detection is defined in terms of an RMSE threshold (a peak would be higher if a higher error threshold was used for declaring the algorithm on track). However, when the same threshold is used for each data series in Fig. 13, the predictable trend of increasing peak performance with increasing SNR can be observed.

These results assume a Gaussian intensity detection structure for the measurement extraction, which represents a "worst case." Of course, if there is some other extraction process with which a higher detection probability and/or a lower level of clutter can be obtained (if, for instance, the preprocessing of images involved software that used features like shape or size to further discriminate between targets and clutter), then the performance of the tracker will be better.

D. Track-Before-Fuse Comparison

The tracker presented in this paper maximizes the likelihood of the target state directly in the global three-dimensional space. An alternative approach is to make a batch estimate of the track in the measurement space of each individual sensor and then fuse them into a batch estimate in the global three-dimensional space. For a notional representation of this process, refer to Fig. 14. If the sensors are cameras, one could obtain batch estimates of the target track in the image space of each camera (red segments in Fig. 14) using measurements from only the camera under consideration. Then, given the poses of a pair of the cameras, one could triangulate to a corresponding batch estimate in the global space (orange segments in Fig. 14). This approach may be con-

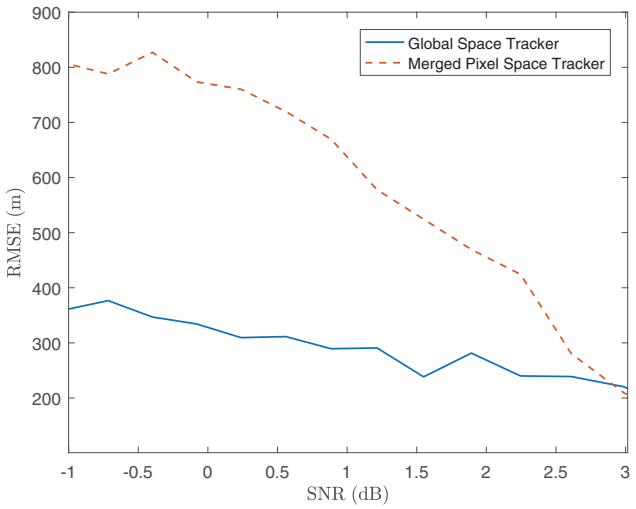


Fig. 15. The RMS position error for the global space tracker (solid line) and the RMSE for the image space tracker (dotted line) plotted for implied measurement extraction SNR values. Errors are averaged over 100 Monte Carlo runs and averaged over time.

sidered desirable in situations where communication of data is restricted: In most realistic settings, the communication cost for sensors to send batches of measurement scans to a central maximizer is higher than if each sensor sends an ML estimate of a track segment. However, this method has the obvious drawback of not having access to the discriminatory power of the "hill finder" algorithm in Section IV.A since the premise is that measurement scans have not been sent to a central location.

We use this decentralized method as a benchmark comparison at various SNR values. The expectation is that the method presented in this paper should be more robust—in terms of mean squared estimation error—as the SNR decreases. This intuitive expectation is due to the fact that both the hill finder and the centralized likelihood evaluation use the relative geometric arrangement of the sensors, which creates a triangulation effect. In terms of the log likelihood, the effect is that the hills climbed by the central tracker are much steeper and taller than the hills in the individual sensor space trackers, which results in a better ML estimate from the central tracker.

Monte Carlo simulations of the scenario described in the previous section are performed as the SNR varies from -1 to 3 dB. As the scenario progresses, both types of tracking are performed simultaneously—tracking according to the algorithm presented in this paper and ML-PMHT tracking in the image spaces of the individual sensors and then fusing into three-dimensional space. The RMSEs obtained with the two different methods are plotted versus the SNR in Fig. 15.

E. JPDA Comparison

A scenario with two targets is simulated in order to test the multi-target capabilities of the ML method presented in this work. For the sake of comparison, the scenario is also subjected to the JPDA tracker, which is a typical recursive maximum *a posteriori* method that has been adapted in [20] to work with multiple sensors.

The sensor arrangement remains the same as the simulation in Section V.A. Along with the original target from the scenario in Section V.A, an additional target is simulated that starts from rest at the point (1000.0, 0.0, 0.0) in the global space and accelerates linearly to arrive at the point $(-1000.0, 0.0, 2000.0)$ at the end of the scenario. The linear path of the second target remains well-resolved from the first target in the three-dimensional global space, although the targets may become unresolved in the two-dimensional measurement space of any one sensor. Although both targets violate the constant velocity assumption of the ML-PMHT, it remains a close approximation over the batch duration, which is shortened to $N^{\mathrm{b}} = 12$ for this simulation. The batch slide is also reduced to 1. That is, a batch estimate is obtained at every time step based on the current scans along with the scans from the past 11 time steps. This ensures that the ML-PMHT is informed by every point obtained by the hill finder routine.

The measurement error is the same as in Section V.A ($\sigma_\theta = \sigma_\phi = 0.25^\circ$). The measurement extraction is assumed to operate such that, for each target, the probability of receiving a hit at each sensor is $p_{\mathrm{d}} = 0.5$, and the expected number of clutter measurements in each scan of each sensor is $\lambda = 50$.

For the JPDA, a single linear white noise acceleration model is assumed for the targets with the process noise parameterization being informed by the maximum acceleration exhibited by the true targets.

Both the JPDA and the ML-PMHT are initialized with a random joint state estimate distributed (with large variance) around the truth, although the ML-PMHT is also allowed immediate access to the hill finder as well, so it has a chance of finding a better initialization for its first batch estimate.

Fig. 16 shows the root-mean-squared position estimate error versus the time step of the scenario averaged over both targets. The benefit of the point finder employed by the ML-PMHT is evident at the beginning of the scenario—while both trackers are given the same random initialization, the ML-PMHT immediately makes use of the point finding, which, on average, results in faster convergence to the "steady-state" performance. A test for statistical significance of the Monte Carlo comparison, as given in [4], is as follows. For a particular Monte Carlo run mc, the RMSE from each of the tracking methods is averaged over time, and the difference between the averages is noted as Δ_{mc}. After all runs are completed, the sample mean and sample standard deviation of the "deltas" are computed, and the significance of the comparison is taken to be the sample mean divided by the sample standard deviation. A value $\gg 2$ is taken to indicate that the performance difference is present in a significant number of runs. The comparison in this simulation was found to have a significance of 4.7.

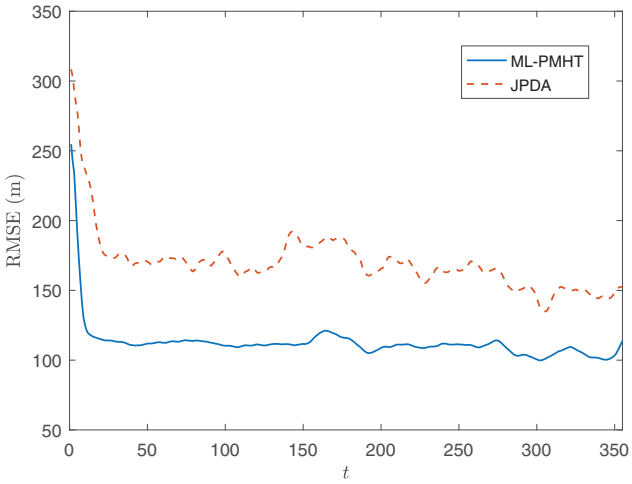


Fig. 16. The RMS position error for the maximum-likelihood tracker (solid line) and the RMSE for the JPDA (dotted line) plotted versus time step. Errors are averaged over 100 Monte Carlo runs and averaged over both targets.

Note that this scenario simulates a relatively low SNR, and previous work has suggested that the ML-PMHT likelihood formulation is more suitable than the JPDA in such settings [14], [21].

VI. CONCLUSION

The algorithm presented in this work is found to outperform a similar track-before-fusing algorithm at reasonably low SNR levels. This result is intuitively predictable given that the global space tracker is taking advantage of knowing the geometric layout of the sensors relative to each other when evaluating the observation likelihood. It is also shown that there is a performance "sweet spot" for the underlying measurement extraction (the process by which "point hits" are declared) where the target detection probability is high enough for the likelihood maximization to be effective, but the level of clutter does not overwhelm the batch estimation.

The "point finding" method presented in Section IV.A, culminating in the list of weighted points in (46), is not fundamentally tied to the ML-PMHT. Instead, it represents a standalone method for obtaining a list of points in three-dimensional Cartesian space by fusing "likely-to-be-related" pairs of two-dimensional line-of-sight measurements from different sensors. The extracted points are used in this work to initialize the likelihood maximization, and are found to help the ML-PMHT stay on track. However, the point extraction could also be used in other fundamentally different settings. For instance, the "target birth" process and two-point-differencing, which is essential to track initialization in the multi-Bernoulli formulation, could be informed by the presented point finding method in a tracking scenario where targets exist in $\mathbb{R}^3$ and multiple sensors report two-dimensional measurements.

The formulation of the batch ML-PMHT presented in this work allows for tracking multiple targets, and is compared to the JPDA for a two-target scenario with high clutter level and low probability of target detection. The ML-PMHT is found to outperform the JPDA in this case, which agrees with results in other works. In terms of average computational cost, the ML-PMHT avoids the expensive data-association step required by the JPDA, but encounters the curse of dimensionality in the parameter space when dealing with multiple targets. Overall, the results suggest that the ML-PMHT is a desirable option in settings where the user wishes to refine track estimates for relatively few targets in the presence of relatively high levels of clutter and low detection probability. Future work will insert the results of the ML-PMHT to a generalized likelihood ratio test for target existence using the threshold-setting techniques in [18], and will compare to an automatically track-managed approach such as the multi-Bernoulli filter [15].

APPENDIX A JACOBIAN COMPUTATION FOR ILS

Each iteration of the ILS estimator requires computation of (41): the Jacobian matrix of the stacked measurement function of sensors a and b with respect to Cartesian position of a target in the global space. In keeping with the notation used in Sections II.B and IV.A.1, this matrix is given by

$$\mathbf{H}_q = \begin{bmatrix} \frac{\partial \bar{\theta}_a}{\partial X} & \frac{\partial \bar{\theta}_a}{\partial Y} & \frac{\partial \bar{\theta}_a}{\partial Z} \\ \frac{\partial \bar{\phi}_a}{\partial X} & \frac{\partial \bar{\phi}_a}{\partial Y} & \frac{\partial \bar{\phi}_a}{\partial Z} \\ \frac{\partial \bar{\theta}_b}{\partial X} & \frac{\partial \bar{\theta}_b}{\partial Y} & \frac{\partial \bar{\theta}_b}{\partial Z} \\ \frac{\partial \bar{\phi}_b}{\partial X} & \frac{\partial \bar{\phi}_b}{\partial Y} & \frac{\partial \bar{\phi}_b}{\partial Z} \end{bmatrix}_{\mathbf{X}=\hat{\mathbf{p}}_q}, \tag{53}$$

where X, Y, Z are the individual Cartesian coordinates of the position vector $\mathbf{X}$ in the global reference frame. Using the formula in (14), which transforms a point in global coordinates into a point in the coordinate system of sensor i, and, specifically, the rotation matrix defined by (15), it can be shown that

$$\mathbf{H}_q = \begin{bmatrix} \mathbf{H}_{q,a}\ \boldsymbol{\Gamma}'_a \\ \mathbf{H}_{q,b}\ \boldsymbol{\Gamma}'_b \end{bmatrix}, \tag{54}$$

where $\boldsymbol{\Gamma}'_i$ is the transpose of the rotation matrix of sensor i, and we have used the property of a rotation matrix that its inverse is equal to its transpose. The other sub-matrix terms are given by

$$\mathbf{H}_{q,i} = \begin{bmatrix} \frac{\partial \bar{\theta}_i}{\partial x_i} & \frac{\partial \bar{\theta}_i}{\partial y_i} & \frac{\partial \bar{\theta}_i}{\partial z_i} \\ \frac{\partial \bar{\phi}_i}{\partial x_i} & \frac{\partial \bar{\phi}_i}{\partial y_i} & \frac{\partial \bar{\phi}_i}{\partial z_i} \end{bmatrix}_{\mathbf{x}_i=\boldsymbol{\Gamma}'_i(\hat{\mathbf{p}}_q-\boldsymbol{\xi}_i)}, \tag{55}$$

which is the Jacobian of the measurement vector of sensor i with respect to a point in the coordinate system of sensor i, evaluated at the current ILS estimate transformed to the coordinate system of sensor i. Taking the corresponding partial derivatives of (9) yields the

individual terms

$$\frac{\partial \bar{\theta}_i}{\partial x_i} = 0, \tag{56}$$

$$\frac{\partial \bar{\theta}_i}{\partial y_i} = \frac{z_i}{y_i^2 + z_i^2}, \tag{57}$$

$$\frac{\partial \bar{\theta}_i}{\partial z_i} = \frac{-y_i}{y_i^2 + z_i^2}, \tag{58}$$

$$\frac{\partial \bar{\phi}_i}{\partial x_i} = \frac{\sqrt{y_i^2 + z_i^2}}{x_i^2 + y_i^2 + z_i^2}, \tag{59}$$

$$\frac{\partial \bar{\phi}_i}{\partial y_i} = \frac{-x_i y_i}{(x_i^2 + y_i^2 + z_i^2)\sqrt{y_i^2 + z_i^2}}, \tag{60}$$

$$\frac{\partial \bar{\phi}_i}{\partial z_i} = \frac{-x_i z_i}{(x_i^2 + y_i^2 + z_i^2)\sqrt{y_i^2 + z_i^2}}, \tag{61}$$

where x_i, y_i, z_i are the individual Cartesian coordinates in the sensor i reference frame. As indicated in (55), for iteration q of the ILS, these terms are evaluated at the coordinates of $\mathbf{x}_i$ given by transforming the current ILS estimate into the reference frame of sensor i with

$$\mathbf{x}_i = \mathbf{\Gamma}_i'(\hat{\mathbf{p}}_q - \boldsymbol{\xi}_i), \tag{62}$$

which, when evaluated for both sensors a and b, will yield $\mathbf{H}_q$ via (54). Note that the notation used here is as in Section IV.A.1 where, since the ILS is being run using a single pair of measurements taken at a single point in time, the time indexing has been omitted. However, in general, the sensor rotation matrices $\mathbf{\Gamma}_i$ and positions $\boldsymbol{\xi}_i$ will be time-dependent.

APPENDIX B ILS INITIALIZATION AND A COMPUTATION-SAVING TRICK

The initial estimate $\hat{\mathbf{p}}_0$ given to the ILS estimator can be evaluated in various ways, and different works differ in the initialization method. In this work, we found that an initialization based on the closest approach between the two line-of-sight measurements worked very well. As discussed in Section IV.A.1, the closest approach will have been previously evaluated for any measurement pair being used in the ILS estimate. Thus, it costs no extra computation to compute the initial estimate as

$$\hat{\mathbf{p}}_0 = \frac{\rho_a}{\rho_a + \rho_b}(\mathbf{p}_b^{\text{md}} - \mathbf{p}_a^{\text{md}}) + \mathbf{p}_a^{\text{md}}, \tag{63}$$

where the positions $\mathbf{p}_i^{\text{md}}$ for $i = a, b$ are the end points of the minimum distance segment connecting the lines-of-sight indicated by measurements a and b evaluated via (30). And, ρ_a, ρ_b are scalars given by (32) and (33), respectively, which provide the Cartesian distance along the lines-of-sight at which the minimum distance occurs. Thus, (63) is a point on the minimum distance segment that lies proportionally closer to the line-of-sight from the sensor that is physically closer to the closest approach. The reasoning is that the farther the sensor is from the target, the farther the corresponding line-of-sight measurement will deviate from the target (in Cartesian distance) due to some fixed measurement error. Thus, when considering a pair of sensors, the line-of-sight from the sensor closer to the target tends to be—in an expected value sense—more "trustworthy" in terms of its Cartesian deviation from the target. This reasoning assumes that the measurement errors in each sensor are identically distributed; if not, then the reasoning becomes less logical. The ILS estimate takes into account non-identical measurement error distributions, however, and (63) still serves as a good initialization in such cases.

If the measurement errors from both sensors *are* identically distributed, the position given by (63) has been found to be a relatively good estimate of the target position, without even running ILS. Of course, ILS will always provide further refinement of the estimate. However, under certain conditions, the composite measurements obtained in Section IV.A.1 via ILS can be replaced by simple evaluation of (63) for any measurement pairs that pass the elimination tests. This would only be done in the name of saving the computation power involved in running ILS for each composite measurement, and if one has computation power to spare, it is not a recommended compromise. However, in simulations with a large number of sensors ($N^{\text{s}} \geq 3$) and in which all pairs of sensors have good cross-range confirmation (see geometric dilution of precision in [4]), composite measurements via (63) were found to serve the purposes of this algorithm rather well. This is due to the fact that the composite measurements, regardless of the level of refinement, are, in turn, used only for the initialization of the batch estimate, which is further refined via the hill climber. Thus, any lack of refinement in the composite measurements from pairs of sensors is quickly made up for once the batch ML estimation is started using all sensors. Qualitatively speaking, this is a good compromise for scenarios with many sensors and small measurement errors, becoming an increasingly poor practice with increasing measurement error and/or decreasing number of sensors.

If the user chooses to make the compromise laid out above, the final thresholding test after ILS is run in Section IV.A.1 can still be performed by replacing $\hat{\mathbf{p}}$ in (35) with $\hat{\mathbf{p}}_0$ from (63). However, no quantitative statements can be made concerning the distribution of the resulting normalized measurement error squared ϵ_0, regardless of whether or not the measurement pair originated from a common target. That is, under the hypothesis that the measurement pair did originate from a common target, it can no longer be claimed that the normalized measurement error squared is Chi-squared distributed. However, referring to Fig. 17, it can be seen that the shapes of the empirically obtained distributions still lend themselves to a simple test by setting a threshold such that, if the normalized estimation error squared

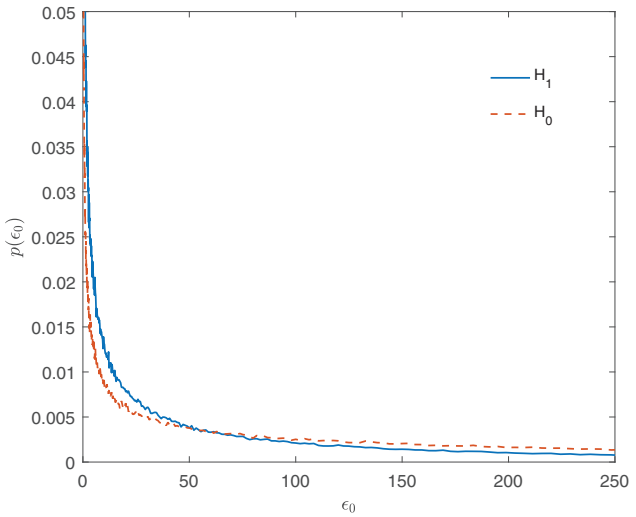


Fig. 17. The empirical distributions of the normalized measurement error squared resulting from the unrefined composite measurement under the binary hypotheses: H_0: the line-of-sight measurements used are unrelated and H_1: the line-of-sight measurements used originated from the same target.

exceeds the threshold, the composite measurement is rejected as being related to a target. However, comparing to the distributions in Fig. 6, it can be seen that the threshold used should be significantly larger than the one used if the composite measurements are refined via ILS. Furthermore, since the distribution under H_1 is no longer Chi-squared, the convenient score based on the Chi-squared cdf in (44) is no longer valid for finalized composite measurements. They must be ranked objectively according to their normalized measurement error squared directly, with smaller errors being considered better.

REFERENCES

[1] S. Avidam and A. Shashua
"Trajectory triangulation: 3D reconstruction of moving points from a monocular image,"
IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 4, pp. 348–357, Apr. 2000.

[2] D. Avitzour
"A maximum likelihood approach to data association,"
IEEE Trans Aerosp. Electron. Syst., vol. AES-28, no. 2, pp. 560–565, Apr. 1992.

[3] B. Balasingam, Y. Bar-Shalom, P. Willett, and K. Pattipati
"Maximum likelihood detection on images," in *Proc. 20th Int. Conf. Inf. Fusion*, Jul. 2017, pp. 1–8.

[4] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software. Hoboken, NJ, USA: Wiley, 2001.

[5] H. Coxeter
Projective Geometry. New York, NY, USA: Springer, 2003.

[6] A. Garcia-Fernandez, L. Svensson, J. Williams, Y. Xia, and K. Granstrom
"Trajectory Poisson multi-Bernoulli filters,"
IEEE Trans. Signal Process., vol. 68, pp. 4933–4945, Aug. 2020.

[7] E. Giannopoulos, R. Streit, and P. Swaszek
"Probabilistic multi-hypothesis tracking in a multi-sensor, multi-target environment,"
in *Proc. 1st Australian Data Fusion Symp.*, Nov. 1996, pp. 184–189.

[8] R. Hartley and A. Zisserman
Multiple View Geometry in Computer Vision. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[9] M. Kowalski, P. Willett, T. Fair, and Y. Bar-Shalom
"CRLB for estimating time-varying rotational biases in passive sensors,"
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 1, pp. 343–355, Feb. 2020.

[10] M. Krieg and D. Gray
"Multi-sensor, probabilistic multi-hypothesis tracking,"
in *Proc. 1st Australian Data Fusion Symp.*, Nov. 1996, pp. 153–158.

[11] Q. Lu, Y. Bar-Shalom, P. Willett, and B. Balasingam
"Measurement extraction for a point target from an optical sensor,"
IEEE Trans Aerospace Electron. Syst., vol. 54, no. 6, pp. 2735–2745, Apr. 2018.

[12] K. G. Murty
"An algorithm for ranking all the assignments in order of increasing cost,"
Operations Res., vol. 16, no. 3, pp. 682–687, May 1968.

[13] R. Osborne and Y. Bar-Shalom
"Statistical efficiency of composite position measurements from passive sensors,"
IEEE Trans. Aerosp. Electron. Syst., vol. 49, no. 4, Oct. 2013, pp. 2799–2806.

[14] C. Rago, P. Willett, and R. Streit
"A comparison of the JPDAF and PMHT tracking algorithms,"
in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, May 1995, pp. 3571–3574.

[15] S. Reuter, B. T. Vo, B. N. Vo, and K. Dietmayer
"The labeled multi-Bernoulli filter,"
IEEE Trans. Signal Process., vol. 62, no. 12, pp. 3246–3260, Jun. 2014.

[16] R. Streit and T. Luginbuhl
"A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning," in *Proc. 6th Joint Service Data Fusion Symp.*, Jun. 1993.

[17] R. Streit and T. Luginbuhl
"Probabilistic multi-hypothesis tracking," Naval Undersea Warfare Center, Newport, RI, USA, Tech. Rep. 10,428, Feb. 1995.

[18] S. Schoenecker, P. Willett, and Y. Bar-Shalom
"Extreme-value analysis for ML-PMHT, Part 1: Threshold determination for false track probability,"
IEEE Trans. Aerosp. Electron. Syst., vol. 50, no. 4, pp. 2500–2514, Oct. 2014.

[19] D. Sunday
Practical Geometry Algorithms With C++ Code.
Independently published, 2021.

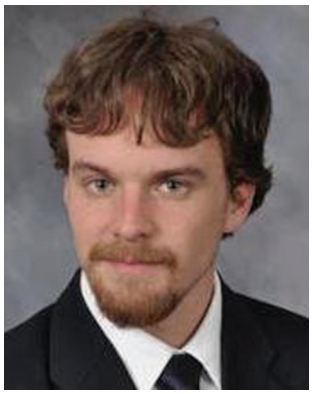
[20] J. K. Tugnait
"Tracking of multiple maneuvering targets in clutter using multiple sensors, IMM, and JPDA coupled filtering,"
IEEE Trans. Aerosp. Electron. Syst., vol. 40, no. 1, Jan. 2004, pp. 320–330.

[21] P. Willett and S. Coraluppi
"MLPDA and ML-PMHT applied to some MSTWG data,"
in *Proc. 9th Int. Conf. Inf. Fusion*, Feb. 2007, pp. 1–8.

[22] Euler angles. (2016). Accessed: Jan. 10, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Euler_angles).

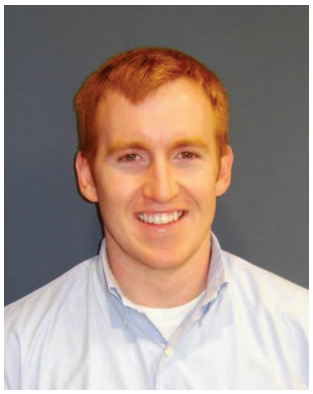
[23] Pinhole camera model. (2018). Accessed: Feb. 10, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Pinhole_camera_model.

[24] scipy.optimize.minimize. (2021). Accessed: Sep. 1, 2021. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html



Zachariah Sutton received the B.Sc. degree in electrical engineering from the University of Connecticut, Storrs, CT, USA, in 2016. He is currently working toward Ph.D. degree in the Electrical and Computer Engineering Department, University of Connecticut. His research interests include signal processing, statistical modeling, and target tracking.

Peter Willett (F'03) has been a faculty member in the Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT, USA, since 1986. Since 1998, he has been a Professor. His primary areas of research have been statistical signal processing, detection, machine learning, communications, data fusion, and tracking. He was the Editor-in-Chief for the IEEE Signal Processing Letters from 2014 to 2016. He was the Editor-in-Chief for the IEEE Transactions on Aerospace and Electronic Systems from 2006 to 2011 and AESS Vice President for Publications from 2012 to 2014. From 1998 to 2005, he was an Associate Editor for three active journals—IEEE Transactions on Aerospace and Electronic Systems (for data fusion and target tracking) and IEEE Transactions on Systems, Man, and Cybernetics, and parts A and B. He is remains an Associate Editor for the IEEE AES Magazine. He is a member of the IEEE AESS Board of Governors and of the IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee (and is now Chair).



Tim Fair, Toyon Senior Staff Analyst, Deputy Director Signal Processing, Principle Investigator, received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the University of California, San Diego, CA, USA, with focus on signal and image processing, in 2008 and 2009, respectively. He has worked with SAIC and Johns Hopkins Applied Physics Lab, where he developed and analyzed detection, tracking, association, and discrimination algorithms for next-generation ISR platforms and Missile Defense Agency satellite systems and the Navy Aegis Ballistic Missile Defense program. Since joining Toyon in 2012, he has focused on algorithm development for image and video processing. As part of this work at Toyon, Mr. Fair has developed solutions in the fields of target detection and tracking, trajectory estimation, target pose estimation, and more recently on methods for applying deep learning and artificial intelligence in the military sector.

Yaakov Bar-Shalom (F'84) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion, Haifa, Israel, in 1963 and 1967, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 1970. He is currently a Board of Trustees Distinguished Professor with the ECE Department and Marianne E. Klewin Professor with the University of Connecticut. His current research interests are in estimation theory, target tracking, and data fusion. He has published more than 650 papers and book chapters. He coauthored/edited eight books, including *Tracking and Data Fusion* (YBS Publishing, 2011). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He served as an Associate Editor for the IEEE Transactions on Automatic Control and Automatica. He was General Chairman of the 1985 ACC, General Chairman of FUSION 2000, President of ISIF in 2000 and 2002, and Vice President for Publications from 2004 to 2013. Since 1995, he has been a Distinguished Lecturer of the IEEE AESS. He is a corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAESystems in 1995 and 2000. In 2002, he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008, he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012, the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in aerospace engineering based on the citations of his work. He is the recipient of the 2015 ISIF Award for a Lifetime of Excellence in Information Fusion. This award has been renamed in 2016 as the Yaakov Bar-Shalom Award for a Lifetime of Excellence in Information Fusion. He has the following Wikipedia page: https://en.wikipedia.org/wiki/Yaakov Bar-Shalom.