Full State Information Transfer Across Adjacent Cameras in a Network Using Gauss Helmert Filters

RONG YANG
YAAKOV BAR-SHALOM

This paper develops three-dimensional (3D) Cartesian tracking algorithms for a high-resolution wide field of view (FOV) camera surveillance system. This system consists of a network linking multiple narrow FOV cameras side-by-side looking at adjacent areas. In such a multi-camera system, a target usually appears in the FOV of one camera first, and then shifts to an adjacent one. The tracking algorithms estimate target 3D positions and velocities dynamically using the angular information (azimuth and elevation) provided by multiple cameras. The target state (consisting of position and velocity) is not fully observable when it is detected by the first camera only. Once it moves into the FOV of the next camera, the state can then be fully estimated. The main challenge is how to transfer the state information from the first camera to the next one when the target moves across cameras. In this paper, we develop an approach, designated as Cartesian state estimation with full maximum likelihood information transfer (fMLIT), to cope with this challenge. Since the fMLIT consists of an implicit state relationship, the conventional Kalman-like filters (for explicit constraints) are not suitable. We then develop three Gauss–Helmert filters, and test them with simulation data.

Manuscript received August 20, 2021; revised November 11, 2021; released for publication July 7, 2022

Refereeing of this contribution was handled by David Crouse.
Authors' addresses: R. Yang, DSO National Laboratories, 12 Science Park Drive, Singapore 118225 (E-mail: yrong@dso.org.sg); Y. Bar-Shalom, Department of ECE, University of Connecticut, Storrs, CT 06269, USA (E-mail: yaakov.bar-shalom@uconn.edu).

I. INTRODUCTION

As more and more cameras are used in surveillance systems, intensive research and development works have been conducted on target detection and tracking using cameras. Most of them deal with extended targets (such as people or vehicles) in near range. After illegal drone intrusions were frequently reported, the camera surveillance has been extended to the more challenging applications which track small air targets in a relatively far range. This paper will focus on developing appropriate algorithms to track small air targets in three-dimensional (3D) Cartesian space using a network of cameras.

In the earlier stage, the computer vision research focused on detection of targets from images, and association of the detections from the same targets over frames based on various features (color, shape, edge, etc.) [9], [16], [18]. Later, it was extended target tracking over video frames to help the track continuity when the detection was imperfect [3], [19]. Further extensions focused on estimation of the target location and trajectory in Cartesian 3D space instead of the two-dimensional (2D) image space. A direct conversion (from 2D image to 3D Cartesian space) can be applied when the camera projection matrix and the target range are available [8], [17]. Another approach is to estimate target range from the ratio of target size and image size (assuming both are known). This is actually an old technique called Stadiametric range finding. Recently, artificial intelligence techniques were introduced to this approach for better accuracy [5].

However, a small air target in a relatively far range (the focus of this paper) has only a few pixels in a video frame. The target image size and range information (with sky background) cannot be obtained accurately. It is hard to obtain the target locations or trajectories in 3D Cartesian space using the above-mentioned techniques. Triangulation from multiple cameras is then a suitable method. In [11], a drone location is estimated by the measured azimuths and elevations from multiple cameras, assuming the cameras are widely spaced and detect a target simultaneously.

In this paper, we consider a realistic camera deployment as shown in Fig. 1. It is a camera network with multiple high-resolution narrow field of view (NFOV) stationary cameras located side by side. Each of them covers a small area and overlaps with the neighbors a little. In this system, a target is detected by one camera only most of time, and the triangulation cannot be performed always. This paper will focus on developing appropriate approaches to track targets in 3D Cartesian space even for cameras with non-overlapping regions. In such a multi-camera system, a target usually appears in the field of view (FOV) of one camera first, and then shifts to an adjacent one. The proposed tracking algorithms will estimate target 3D positions and velocities dynamically using the angular information (azimuth and elevation) provided by multiple cameras. The target state

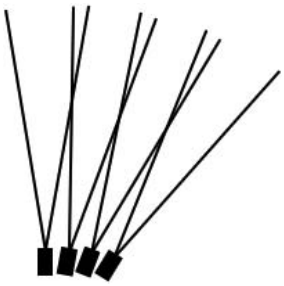


Fig. 1. A wide field of view surveillance system consisting of four NFOV cameras.

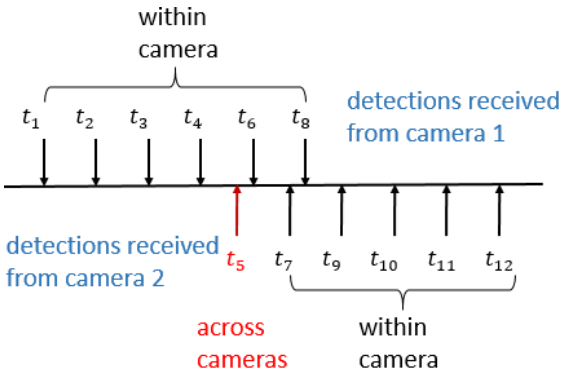


Fig. 2. A target flies from the FOV of camera 1 to FOV of camera 2. The across-camera event (crossover) occurs at time t_5 when camera 2 detects the target for the first time. The others are considered as within-camera events.

(consisting of position and velocity) is not fully observable when the target is detected by the first camera only. Once the target moves into the FOV of the next camera, the state can then be estimated completely using suitable algorithms. The main challenge is how to transfer the useful state information from the first camera to the next one when the target moves across cameras. We will formulate the problem with an unique Gauss–Helmert model (GHM), which provides an implicit constraint that is used to transfer full state information (including 3D position and velocity) across cameras. The approach is designated as Cartesian state estimation with full maximum likelihood information transfer (fMLIT).

A conventional dynamic estimation problem is formulated by two basic models, the state transition model and the measurement model. Usually these two models are in explicit forms as

$$\mathbf{x}(t_k) = \mathbf{f}[\mathbf{x}(t_{k-1})] + \mathbf{v}(t_k, t_{k-1}), \tag{1}$$

$$\mathbf{z}(t_k) = \mathbf{h}[\mathbf{x}(t_k)] + \mathbf{w}(t_k), \tag{2}$$

where $\mathbf{x}(t_k)$ is the state vector to be estimated at time t_k, $\mathbf{z}(t_k)$ is the measurement vector observed by a sensor, $\mathbf{f}[\cdot]$ and $\mathbf{h}[\cdot]$ are the state transition function and measurement function, respectively, and $\mathbf{v}(\cdot)$ and $\mathbf{w}(\cdot)$ are the process noise and measurement noise, respectively. Based on these models, the Kalman-like filters has the following two generic steps to perform estimation:

- Predict the state from time t_{k-1} to t_k using the state transition model (1). The predicted state is $\hat{\mathbf{x}}(t_k|t_{k-1})$.
- Update the predicted state $\hat{\mathbf{x}}(t_k|t_{k-1})$ by the measurement $\mathbf{z}(t_k)$ using the measurement model (2). The updated state is $\hat{\mathbf{x}}(t_k|t_k)$.

However, to perform fMLIT across cameras, an implicit constraint needs to be taken into consideration in addition to the measurement model given in (2) (the details will be given later in the Section II-B). The implicit relationship is between the predicted state $\hat{\mathbf{x}}(t_k|t_{k-1})$ from the previous camera and the current state $\mathbf{x}(t_k)$. Thus, the following GHM is used to replace the measurement model in (2) at the crossover time (details to be given later—see (20)).

$$\mathbf{g}[\mathbf{x}(t_k), \hat{\mathbf{x}}(t_k|t_{k-1}), \mathbf{z}(t_k)] = \mathbf{w}_{\mathrm{g}}, \tag{3}$$

where $\mathbf{w}_{\mathrm{g}}$ is a small zero-mean Gaussian model error. Thus the Kalman-like filters cannot be applied, and need a further development.

The GHM is commonly used for similarity estimation in geodetic science [10], [12], and is also applied to computer vision and curve fitting [4], [7], [13], [14], [15]. It was introduced in dynamic estimation for solving unknown propagation delay problem in the state transition model in [20], [21]. The Unscented GHF (UGHF) was developed to solve the problem. A number of works were conducted for various applications with implicit state transition models [22], [23], [24].

The above mentioned Gauss–Helmert filter (GHF) and its applications use the GHM in their state transition models. The problem in this paper requires an implicit measurement model. Consequently, we will discuss three GHFs and three other algorithms which totally or partially ignore the implicit constraints in the GHM. Their performance will be demonstrated using simulation data.

The rest of paper is structured as following. Section II formulates the problem. Section III develops the six estimation algorithms. Section IV presents the simulation results, and Section V draws the conclusions.

II. PROBLEM FORMULATION

To track a target using the camera suite shown in Fig. 1, we need to formulate the dynamic estimation problem with the state transition model and measurement model within and across cameras, respectively. Fig. 2 illustrates the within and across camera events. When a target flies from the FOV of camera 1 to the FOV of camera 2, it is detected by camera 1 first at times t_1, t_2, t_3, t_4, t_6, and t_8. Camera 2 detects the target at times $t_5, t_7, t_9, t_{10}, t_{11}$, and t_{12}. The across camera event occurs at time t_5 when the target is firstly detected by camera 2. All other occurrences are considered as within-camera events, even they fall in overlapping FOVs. The problem formulate is then based on these two events.

A. Dynamic Estimation Models for the Within-Camera Event

The state vector to be estimated at time t_k is defined as

$$\mathbf{x}(t_k) = [x(t_k)\ y(t_k)\ z(t_k)\ \dot{x}(t_k)\ \dot{y}(t_k)\ \dot{z}(t_k)]'. \quad (4)$$

It consists of the position and velocity components of a target in 3D Cartesian coordinates. The measurement vector from the ith camera (with $i = 1, \ldots, n$, and n is the total number of cameras in the system) is

$$\mathbf{z}_{s_i}(t_k) = [a_{s_i}(t_k)\ e_{s_i}(t_k)]', \quad (5)$$

where $a_{s_i}(t_k)$ is the measured (noisy) azimuth from true North clockwise, and $e_{s_i}(k)$ is the measured elevation up from the horizontal, with the reference to the ith camera position. The state transition model uses the nearly constant velocity (NCV) model [2] as

$$\mathbf{x}(t_k) = \mathbf{F}\mathbf{x}(t_{k-1}) + \mathbf{v}(t_k, t_{k-1}), \quad (6)$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & T(t_k) & 0 & 0 \\ 0 & 1 & 0 & 0 & T(t_k) & 0 \\ 0 & 0 & 1 & 0 & 0 & T(t_k) \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

with time interval

$$T(t_k) = t_k - t_{k-1}, \quad (8)$$

and $\mathbf{v}(t_k, t_{k-1})$ is the white Gaussian process noise with covariance

$$\mathbf{Q}(t_k, t_{k-1}) = \begin{bmatrix} \frac{T(t_k)^3}{3} & 0 & 0 & \frac{T(t_k)^2}{2} & 0 & 0 \\ 0 & \frac{T(t_k)^3}{3} & 0 & 0 & \frac{T(t_k)^2}{2} & 0 \\ 0 & 0 & \frac{T(t_k)^3}{3} & 0 & 0 & \frac{T(t_k)^2}{2} \\ \frac{T(t_k)^2}{2} & 0 & 0 & T(t_k) & 0 & 0 \\ 0 & \frac{T(t_k)^2}{2} & 0 & 0 & T(t_k) & 0 \\ 0 & 0 & \frac{T(t_k)^2}{2} & 0 & 0 & T(t_k) \end{bmatrix} q, \quad (9)$$

and q is the Cartesian acceleration power spectral density (PSD). The measurement model is

$$\mathbf{z}_{s_i}(t_k) = \mathbf{h}[\mathbf{x}(t_k), \mathbf{x}_{s_i}] + \mathbf{w}(t_k) \quad (10)$$

with

$$\mathbf{h}[\mathbf{x}(t_k), \mathbf{x}_{s_i}] = \begin{bmatrix} \operatorname{atan}\left[\frac{x(t_k) - x_{s_i}(t_k)}{y(t_k) - y_{s_i}(t_k)}\right] \\ \operatorname{atan}\left[\frac{z(t_k) - z_{s_i}(t_k)}{\sqrt{[x(t_k) - x_{s_i}(t_k)]^2 + [y(t_k) - y_{s_i}(t_k)]^2}}\right] \end{bmatrix}, \quad (11)$$

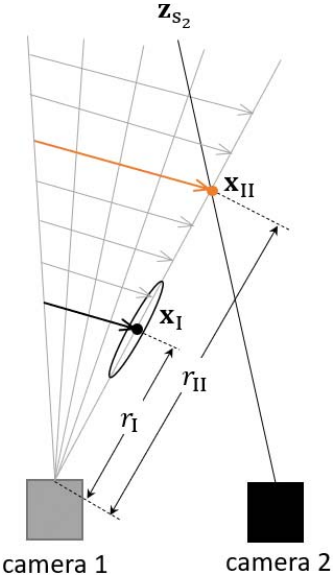


Fig. 3. The ground truth is at $\mathbf{x}_{II}$. The estimated state from camera 1 is $\mathbf{x}_I$. The large estimate bias of $\mathbf{x}_I$ is because the state is not observable from the measurements from one camera.

where $\mathbf{x}_{s_i} = [x_{s_i}\ y_{s_i}\ z_{s_i}]'$ is the position of the ith camera, which provides the measurement, and $\mathbf{w}(t_k)$ is the white Gaussian measurement noise with covariance

$$\mathbf{R} = \operatorname{diag}(\sigma_a^2\ \sigma_e^2), \quad (12)$$

with σ_a and σ_e, which are the measurement error standard deviations of azimuth and elevation, respectively.

To apply these models to our problem directly one has the following issue. The state is not fully observable when a target is detected by one camera only, as the target range cannot be obtained. Fig. 3 shows that the possible target trajectories are a set of parallel lines (assuming the target is in constant velocity motion, and does not head to or move away from the camera directly) [6]. All these parallel trajectories share the same heading, and the speeds on parallel lines are proportional to their ranges (the distances to the camera). The proof of the target heading observability and range unobservablity from a single stationary camera, shown in Fig. 3, is given in Appendix A. The state estimate, using this incompletely observable models, can fall on any one of possible trajectories depending on the initial condition. The initial state estimate is usually set at a particular point along the line of sight (LOS) of the first detection in 3D space and zero velocities with predefined large range and velocity errors. When the state estimate is updated by the angular measurements, it is adjusted to suitable range and speed to match the angular change. This process can lead to a biased estimation for this unobservable problem, namely, the ground truth can fall outside the state uncertainty region. Fig. 4 and its enlarged version Fig. 5 show an estimation example. There are two cameras located at $(0, 0, 50)$ m and $(1, 0, 50)$ m, respectively. The target is in the FOV of the first camera at beginning where the target trajectory is not observable, and moves to the

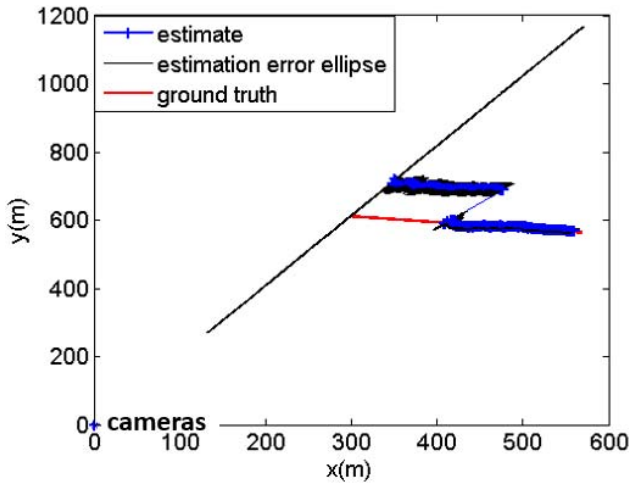


Fig. 4. Estimation from angle-only measurements using two adjacent cameras (here the error ellipses are very elongated).

FOV of the second camera. The track has an initial state with large error in range to cover the ground truth. After the second time cycle, the filter-calculated estimation error is much smaller, and the state converges to a trajectory (blue) parallel to the ground truth (red). Obviously, the estimation is seen to be biased in range. Using a very large uncertainty for the initial state does not solve the problem.

While we cannot make the state observable in this situation, we can transfer all useful information obtained from the first camera to the subsequent estimation. A Gauss–Helmert (GH)-based information transfer can make the state fully observable from the across-camera event (given in the next subsection).

After the across-camera event, the estimation is carried out continuously in the FOV of camera 2. Although the state is still not fully observable in theory (namely, the measurements are from camera 2 only), the bias is, in spite of the marginal observability, reduced significantly after the full information transfer (see Figs. 4 and 5).

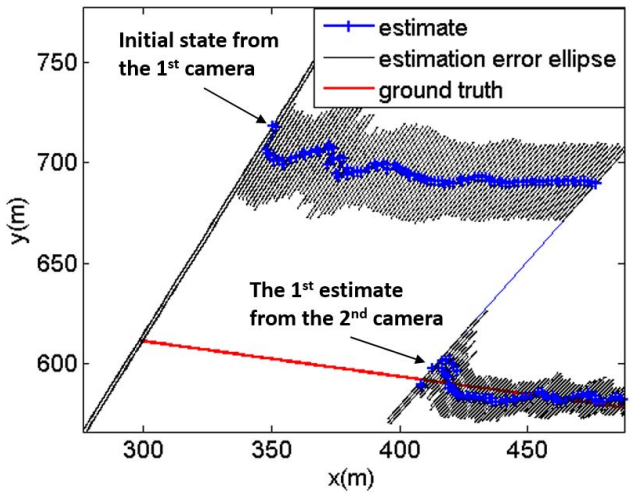


Fig. 5. An enlarged version of Fig. 4.

B. Dynamic Estimation Models for the Across-Camera Event

At the across-camera event, the state vector and the state transition model are the same as those of the within-camera event given in (4) and (6). Assuming the across-camera event happens at time t_k, we predict the state estimate $\hat{\mathbf{x}}(t_{k-1}|t_{k-1})$ and its error covariance $\mathbf{P}(t_{k-1}|t_{k-1})$ to time t_k by

$$\hat{\mathbf{x}}(t_k|t_{k-1}) = \mathbf{F}\hat{\mathbf{x}}(t_{k-1}|t_{k-1}), \tag{13}$$

$$\mathbf{P}(t_k|t_{k-1}) = \mathbf{F}\mathbf{P}(t_{k-1}|t_{k-1})\mathbf{F}' + \mathbf{Q}(t_k, t_{k-1}). \tag{14}$$

To simplify the expression, we denote the predicted state and its error covariance as

$$\mathbf{x}_{\mathrm{I}} = \hat{\mathbf{x}}(t_k|t_{k-1}), \tag{15}$$

$$\mathbf{P}_{\mathrm{I}} = \mathbf{P}(t_k|t_{k-1}), \tag{16}$$

and the state at t_k (see Fig. 3) as

$$\mathbf{x}_{\mathrm{II}} = \mathbf{x}(t_k). \tag{17}$$

Intuitively, the predicted state $\mathbf{x}_{\mathrm{I}}$ can be updated to $\mathbf{x}_{\mathrm{II}}$ by the second camera measurement $\mathbf{z}_{s_2}(t_k)$ based on the measurement model giving in (10). However, it faces the following issues. First, the predicted state $\mathbf{x}_{\mathrm{I}}$ can be significantly biased compared to the ground truth due to the unobservability in the range. Fig. 3 shows an example that the estimated range from the first camera is r_{I}, but the actual range is r_{II}. The error covariance $\mathbf{P}_{\mathrm{I}}$ does not include camera 2's measurement $\mathbf{z}_{s_2}(t_k)$ in its uncertainty region. The Kalman-like filters cannot bring the state $\mathbf{x}_{\mathrm{I}}$ close to $\mathbf{x}_{\mathrm{II}}$ enough with $\mathbf{z}_{s_2}(t_k)$, as the filters are designed to reduce the random error, and are not effective for a large bias error. Second, the state information estimated by the first camera is not fully utilized. The headings in $\mathbf{x}_{\mathrm{I}}$ and $\mathbf{x}_{\mathrm{II}}$ should be consistent, and the speeds should be proportional to their ranges. Thus, we reformulate the problem to estimate $\mathbf{x}_{\mathrm{II}}$ from a known augmented parameter $\mathbf{y}$ with error covariance $\mathbf{P}_{\mathrm{y}}$

$$\mathbf{y} = [\mathbf{x}_{\mathrm{I}}' \; \mathbf{z}_{s_2}(t_k)']', \tag{18}$$

$$\mathbf{P}_{\mathrm{y}} = \begin{bmatrix} \mathbf{P}_{\mathrm{I}} & \mathbf{0}_{6\times 2} \\ \mathbf{0}_{2\times 6} & \mathbf{R} \end{bmatrix}. \tag{19}$$

Since $\mathbf{x}_{\mathrm{II}}$ and $\mathbf{y}$ have an implicit relationship, as shown below in (20)–(31), a GH measurement model is defined as

$$\mathbf{g}(\mathbf{x}_{\mathrm{II}}, \mathbf{y}) = \mathbf{w}_{\mathrm{g}}, \tag{20}$$

which is the same as (3), where

$$\mathbf{g}(\cdot) = [g_1(\cdot)\; g_2(\cdot)\; g_3(\cdot)\; g_4(\cdot)\; g_5(\cdot)\; g_6(\cdot)\; g_7(\cdot)\; g_8(\cdot)]', \tag{21}$$

with

$$g_1(\cdot) = r_{\mathrm{I}}(x_{\mathrm{II}} - x_{s_1}) - r_{\mathrm{II}}(x_{\mathrm{I}} - x_{s_1}), \tag{22}$$

$$g_2(\cdot) = r_{\rm I}(y_{\rm II} - x_{s_1}) - r_{\rm II}(y_{\rm I} - x_{s_1}), \tag{23}$$

$$g_3(\cdot) = r_{\rm I}(z_{\rm II} - x_{s_1}) - r_{\rm II}(z_{\rm I} - x_{s_1}), \tag{24}$$

$$g_4(\cdot) = r_{\rm I}\dot{x}_{\rm II} - r_{\rm II}\dot{x}_{\rm I}, \tag{25}$$

$$g_5(\cdot) = r_{\rm I}\dot{y}_{\rm II} - r_{\rm II}\dot{y}_{\rm I}, \tag{26}$$

$$g_6(\cdot) = r_{\rm I}\dot{z}_{\rm II} - r_{\rm II}\dot{z}_{\rm I}, \tag{27}$$

$$g_7(\cdot) = \arctan\left(\frac{x_{\rm II} - x_{s_2}}{y_{\rm II} - y_{s_2}}\right) - a_{s_2}(t_k), \tag{28}$$

$$g_8(\cdot) = \arctan\left(\frac{z_{\rm II} - z_{s_2}}{\sqrt{(x_{\rm II} - x_{s_2})^2 + (y_{\rm II} - y_{s_2})^2}}\right) - e_{s_2}(t_k), \tag{29}$$

and

$$r_{\rm I} = \sqrt{(x_{\rm I} - x_{s_1})^2 + (y_{\rm I} - y_{s_1})^2 + (z_{\rm I} - z_{s_1})^2}, \tag{30}$$

$$r_{\rm II} = \sqrt{(x_{\rm II} - x_{s_1})^2 + (y_{\rm II} - y_{s_1})^2 + (z_{\rm II} - z_{s_1})^2}. \tag{31}$$

This model considers three constraints, first, the position $(x_{\rm II}, y_{\rm II}, z_{\rm II})$ in $\mathbf{x}_{\rm II}$ and the position $(x_{\rm I}, y_{\rm I}, z_{\rm I})$ in $\mathbf{x}_{\rm I}$ are on the same LOS with reference to camera 1. This is described in (22)–(24). Second, the velocities $(\dot{x}_{\rm II}, \dot{y}_{\rm II}, \dot{z}_{\rm II})$ and $(\dot{x}_{\rm I}, \dot{y}_{\rm I}, \dot{z}_{\rm I})$ are proportional to their ranges with reference to camera 1, which is given in (25)–(27). Third, the position $(x_{\rm II}, y_{\rm II}, z_{\rm II})$ is on the LOS of camera 2's measurement, as given in (28)–(29). The third constraint is the within-camera measurement model given in (10). The GH measurement model adds the constraints in (22)–(27). This ensures that the useful state information obtained from camera 1 is fully transferred to the subsequent estimation in a proper manner.

III. DYNAMIC ESTIMATION ALGORITHMS

This section presents six dynamic estimation algorithms for the problem addressed in this paper:

- EKF: extended Kalman filter without implicit constraint;
- pMLIT: partial maximum likelihood information transfer;
- fMLIT: full maximum likelihood information transfer without considering cross-correlation;
- fMLIT-EGHF: full maximum likelihood information transfer using extended Gauss–Helmert filter;
- fMLIT-UGHF: full maximum likelihood information transfer using unscented Gauss-Helmert filter; and
- fMLIT-UGHFapp: full maximum likelihood information transfer using unscented Gauss-Helmert filter with approximation.

Like all dynamic estimation algorithms, they consist of two steps, prediction and update. The six algorithms share the same prediction step given in (13) and (14). The update step is also the same for within-camera estimation, which follows the extended Kalman filter (EKF)[1] as

$$\hat{\mathbf{x}}(t_k|t_k) = \hat{\mathbf{x}}(t_k|t_{k-1}) + \mathbf{W}(t_k)\nu(t_k), \tag{32}$$

$$\mathbf{P}(t_k|t_k) = \mathbf{P}(t_k|t_{k-1}) - \mathbf{W}(t_k)\mathbf{S}(t_k)\mathbf{W}(t_k)', \tag{33}$$

where

$$\nu(t_k) = \mathbf{z}_{s_i}(t_k) - \mathbf{h}[\hat{\mathbf{x}}(t_k|t_{k-1}), \mathbf{x}_{s_i}], \tag{34}$$

$$\mathbf{W}(t_k) = \mathbf{P}(t_k|t_{k-1})\mathbf{H}(t_k)\mathbf{S}(t_k)^{-1}, \tag{35}$$

$$\mathbf{S}(t_k) = \mathbf{R} + \mathbf{H}(t_k)\mathbf{P}(t_k|t_{k-1})\mathbf{H}(t_k)', \tag{36}$$

and

$$\mathbf{H}(t_k) = \left.\frac{\partial \mathbf{h}(\mathbf{x}, \mathbf{x}_{s_i})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(t_k|t_{k-1})}, \tag{37}$$

with $\mathbf{h}(\cdot)$ given in (11). The differences between the six algorithms are at the across-camera update step. This is the main focus of this section, and will be given next.

A. Extended Kalman Filter without Implicit Constraint

This algorithm updates the target across camera state using the measurement model given in (10). The implicit constraints given in (22)–(27) are totally ignored. Its update step is according to the EKF given in (32)–(33).

As we discussed in Section II, this algorithm suffers from large errors due to unobservability. We include it here as baseline for comparison purpose. This algorithm may be in use in some real applications due to its easy implementability. It is worth to evaluate its performance to understand its limitations.

B. Partial Maximum Likelihood Information Transfer (pMLIT)

This algorithm updates the across-camera state using the partial information in the predicted state $\mathbf{x}_{\rm I}$ (position only). It converts the predicted position to the LOS information (azimuth and elevation) with reference to the first camera, and then combines with the LOS measurement from the second camera to estimate the position in $\mathbf{x}_{\rm II}$ using the maximum likelihood (ML) estimation. The velocity in $\mathbf{x}_{\rm I}$ is not utilized, and the velocity in $\mathbf{x}_{\rm II}$ is not estimated. The details are given in sequel.

[1] The EKF is selected for within-camera estimation. This is because the camera measurement error is small. The other nonlinear filters like unscented Kalmen filter and particle filter cannot improve estimation accuracy when the measurement errors are small. Furthermore, the EKF is more efficient, and is good for the high sampling/measurement rate of a camera.

First, the LOS and its error covariance from the first camera are computed by

$$\mathbf{z}_{\mathrm{IA}} = [a_{\mathrm{I}}\ e_{\mathrm{I}}]' = \mathbf{h}(\mathbf{x}_{\mathrm{I}}, \mathbf{x}_{\mathrm{s}_1}), \tag{38}$$

$$\mathbf{R}_{\mathrm{IA}} = \mathbf{H}(t_k)\mathbf{P}_{\mathrm{I}}\mathbf{H}(t_k)', \tag{39}$$

where $\mathbf{h}(\cdot)$ and $\mathbf{H}(\cdot)$ are given in (10) and (37), respectively.

Second. the position in $\mathbf{x}_{\mathrm{II}}$ is estimated. The LOS of the second camera is the measurement

$$\mathbf{z}_{\mathrm{s}_2}(t_k) = [a_{\mathrm{s}_2}(t_k)\ e_{\mathrm{s}_2}(t_k)]'. \tag{40}$$

The iterated least squares (ILS) [1] is used to obtain the ML estimate of the target position by the model

$$\mathbf{z}_{\mathrm{p}} = \mathbf{h}_{\mathrm{p}}(\mathbf{x}_{\mathrm{p}}, \mathbf{x}_{\mathrm{s}_1}, \mathbf{x}_{\mathrm{s}_2}) + \mathbf{w}_{\mathrm{p}}, \tag{41}$$

where

$$\mathbf{z}_{\mathrm{p}} = [\mathbf{z}'_{\mathrm{IA}}\ \mathbf{z}_{\mathrm{s}_2}(t_k)']', \tag{42}$$

$$\mathbf{x}_{\mathrm{p}} = \mathbf{x}_{\mathrm{II}}(1:3) = [x_{\mathrm{II}}\ y_{\mathrm{II}}\ z_{\mathrm{II}}]', \tag{43}$$

$$\mathbf{h}_{\mathrm{p}}[\mathbf{x}_{\mathrm{p}}, \mathbf{x}_{\mathrm{s}_1}, \mathbf{x}_{\mathrm{s}_2}] = \begin{bmatrix} \arctan\left(\frac{x_{\mathrm{II}}-x_{\mathrm{s}_1}}{y_{\mathrm{II}}-y_{\mathrm{s}_1}}\right) \\ \arctan\left(\frac{z_{\mathrm{II}}-z_{\mathrm{s}_1}}{\sqrt{(x_{\mathrm{II}}-x_{\mathrm{s}_1})^2+(y_{\mathrm{II}}-y_{\mathrm{s}_1})^2}}\right) \\ \arctan\left(\frac{x_{\mathrm{II}}-x_{\mathrm{s}_2}}{y_{\mathrm{II}}-y_{\mathrm{s}_2}}\right) \\ \arctan\left(\frac{z_{\mathrm{II}}-z_{\mathrm{s}_2}}{\sqrt{(x_{\mathrm{II}}-x_{\mathrm{s}_2})^2+(y_{\mathrm{II}}-y_{\mathrm{s}_2})^2}}\right) \end{bmatrix}, \tag{44}$$

and $\mathbf{w}_{\mathrm{p}}$ is the error of $\mathbf{z}_{\mathrm{p}}$ with the covariance

$$\mathbf{R}_{\mathrm{p}} = \begin{bmatrix} \mathbf{R}_{\mathrm{IA}} & \mathbf{0}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{R} \end{bmatrix}. \tag{45}$$

The position ML estimate is obtained by the ILS algorithm as

$$\hat{\mathbf{x}}_{\mathrm{p}}^{j+1} = \hat{\mathbf{x}}_{\mathrm{p}}^{j} + \mathbf{P}_{\mathrm{p}}^{j+1}(\mathbf{J}^j)'\mathbf{R}_{\mathrm{p}}^{-1}[\mathbf{z}_{\mathrm{p}} - \mathbf{h}_{\mathrm{p}}(\hat{\mathbf{x}}_{\mathrm{p}}^{j}, \mathbf{x}_{\mathrm{s}_1}, \mathbf{x}_{\mathrm{s}_2})], \tag{46}$$

$$\mathbf{P}_{\mathrm{p}}^{j+1} = [(\mathbf{J}^j)'\mathbf{R}_{\mathrm{p}}^{-1}\mathbf{J}^j]^{-1}, \tag{47}$$

where j is the iteration index, and

$$\mathbf{J}^j = \left.\frac{\partial \mathbf{h}_{\mathrm{p}}(\mathbf{x}_{\mathrm{p}}, \mathbf{x}_{\mathrm{s}_1}, \mathbf{x}_{\mathrm{s}_2})}{\partial \mathbf{x}_{\mathrm{p}}}\right|_{\mathbf{x}_{\mathrm{p}}=\hat{\mathbf{x}}_{\mathrm{p}}^{j}}. \tag{48}$$

The state $\mathbf{x}_{\mathrm{II}}$ estimate and its error covariance are then given by

$$\hat{\mathbf{x}}_{\mathrm{II}} = [\hat{\mathbf{x}}'_{\mathrm{p}}\ 0\ 0\ 0]', \tag{49}$$

$$\mathbf{P}_{\mathrm{II}} = \begin{bmatrix} \mathbf{P}_{\mathrm{p}} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathrm{diag}(\sigma_{\dot{x}}^2\ \sigma_{\dot{y}}^2\ \sigma_{\dot{z}}^2) \end{bmatrix}, \tag{50}$$

where $\sigma_{\dot{x}}$, $\sigma_{\dot{y}}$, and $\sigma_{\dot{z}}$ are predefined velocity error standard deviations in x, y, and z coordinates, respectively.

This algorithm can overcome the range bias problem in the EKF described in the Section III-A. It carries the estimated azimuth and elevation information from the first camera to the subsequent tracking process. However, the velocity estimates are lost, and the velocities in $\mathbf{x}_{\mathrm{II}}$ in (49) are set to 0 without the contribution from the previous state estimate. Thus the information from the previous state is only partially transferred to the subsequent tracking process.

C. Full Maximum Likelihood Information Transfer (fMLIT) without Considering Cross-Correlation

This algorithm estimates $\mathbf{x}_{\mathrm{II}}$ and its error covariance $\mathbf{P}_{\mathrm{II}}$ using the fMLIT. However, it ignores the cross-covariance between the position and velocity error. The details are given below.

First, the algorithm estimates the position in $\mathbf{x}_{\mathrm{II}}$ using the same method as the pMLIT described in the Section III-B.

Second, $\mathbf{x}_{\mathrm{II}}$ (including velocity estimation) and its error covariance $\mathbf{P}_{\mathrm{II}}$ are estimated by

$$\hat{\mathbf{x}}_{\mathrm{II}} = [\hat{\mathbf{x}}'_{\mathrm{p}}\ \lambda\mathbf{x}'_{\mathrm{I},(4:6)}]', \tag{51}$$

$$\mathbf{P}_{\mathrm{II}} = \begin{bmatrix} \mathbf{P}_{\mathrm{p}} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \lambda^2\mathbf{P}_{\mathrm{I},(4:6,4:6)} \end{bmatrix}, \tag{52}$$

where

$$\lambda = \frac{\hat{r}_{\mathrm{II}}}{r_{\mathrm{I}}}, \tag{53}$$

where r_{I} and $\hat{r}_{\mathrm{II}}$ are computed using (30) and (31), respectively (see Fig. 3).

This is an approximate solution to compute the velocity in $\mathbf{x}_{\mathrm{II}}$. The velocity error covariance is also approximately proportional with factor λ^2. The cross-covariance between the errors of the position and velocity is not taken into consideration, setting it to $\mathbf{0}_{3\times 3}$ in $\mathbf{P}_{\mathrm{II}}$.

D. Full Maximum Likelihood Information Transfer Using Extended Guass–Helmert Filter (fMLIT-EGHF)

This algorithm updates the across-camera state based on the GHM described in Section II-B. Since the GHM is nonlinear, we approximate the nonlinear transformation by the first-order Taylor expansion. Following the name of the EKF, using the same approximation approach, we designate the algorithm as extended GHF (EGHF).

The update step is performed by the following iteration

$$\mathbf{P}_{\mathrm{II}}^{j+1} = \left[(\mathbf{A}^j)'[\mathbf{B}^j\mathbf{P}_{\mathrm{y}}(\mathbf{B}^j)']^{-1}\mathbf{A}^j\right]^{-1,} \tag{54}$$

$$\hat{\mathbf{x}}_{\mathrm{II}}^{j+1} = \hat{\mathbf{x}}_{\mathrm{II}}^{j} - \mathbf{P}_{\mathrm{II}}^{j+1}(\mathbf{A}^j)'[\mathbf{B}^j\mathbf{P}_{\mathrm{y}}(\mathbf{B}^j)']^{-1}\mathbf{g}(\hat{\mathbf{x}}_{\mathrm{II}}^{j}, \mathbf{y}), \tag{55}$$

where j is the iteration index,

$$\mathbf{A}^j = \left.\frac{\partial \mathbf{g}(\mathbf{x}_{\rm II}, \mathbf{y})}{\partial \mathbf{x}_{\rm II}}\right|_{\mathbf{x}_{\rm II}=\hat{\mathbf{x}}_{\rm II}^j}, \tag{56}$$

$$\mathbf{B}^j = \left.\frac{\partial \mathbf{g}(\mathbf{x}_{\rm II}, \mathbf{y})}{\partial \mathbf{y}}\right|_{\mathbf{x}_{\rm II}=\hat{\mathbf{x}}_{\rm II}^j}. \tag{57}$$

The initial $\hat{\mathbf{x}}_{\rm II}^0$ is set as in (51). The Jaconbians $\mathbf{A}$ and $\mathbf{B}$ are given in Appendix B.

This algorithm is a solution for the implicit GHM given in the Section II-B, using the first-order Taylor expansion to approximate the nonlinear function (20). However, in addition to the well-known disadvantages of the first-order Taylor approximation, such as not being suitable for highly nonlinear models with large errors, the EGHF involves matrix inversion operations. This may cause numerical issues when the matrices are ill-conditioned. This will be discussed later in the simulation tests.

E. Full Maximum Likelihood Information Transfer using Unscented Gauss–Helmert Filter (fMLIT-UGHF)

This algorithm is also a GHF to provide a solution to the GH measurement model. To better approximate the nonlinear transformation, the unscented transform is used to replace the first-order Taylor expansion in the fMLIT-EGHF. The algorithm is designated as fMLIT-UGHF.

First, $(2n_{\rm y}+1)$ weighted sigma points of $\mathbf{y}$ are generated as

$$\mathbf{y}_1 = \mathbf{y}, \tag{58}$$

$$\mathbf{y}_i = \mathbf{y} + \left[\sqrt{(n_{\rm y}+\kappa)\mathbf{P}_{\rm y}}\right]_{i-1} \quad i = 2, \ldots, n_{\rm y}+1, \tag{59}$$

$$\mathbf{y}_i = \mathbf{y} - \left[\sqrt{(n_{\rm y}+\kappa)\mathbf{P}_{\rm y}}\right]_{i-n_{\rm y}-1} \quad i = n_{\rm y}+2, \ldots, 2n_{\rm y}+1. \tag{60}$$

Their corresponding weights are

$$w_1 = \frac{\kappa}{n_{\rm y}+\kappa}, \tag{61}$$

$$w_i = \frac{1}{2(n_{\rm y}+\kappa)} \quad i = 2, \ldots, 2n_{\rm y}+1, \tag{62}$$

where $n_{\rm y} = 8$ is the dimension of $\mathbf{y}$, and κ is a scalar to determine the spread of sigma points.

Second, the following iteration step is performed for each sigma point, so that the sigma points of $\mathbf{x}_{\rm II}$ are obtained.

$$\hat{\mathbf{x}}_{{\rm II},i}^{j+1} = \hat{\mathbf{x}}_{{\rm II},i}^{j} - [(\mathbf{A}^j)'\mathbf{A}^j]^{-1}(\mathbf{A}^j)'\mathbf{g}(\hat{\mathbf{x}}_{{\rm II},i}^{j}, \mathbf{y}_i). \tag{63}$$

The initial $\hat{\mathbf{x}}_{{\rm II},i}^0$ is set by (51) based on $\mathbf{y}_i$.

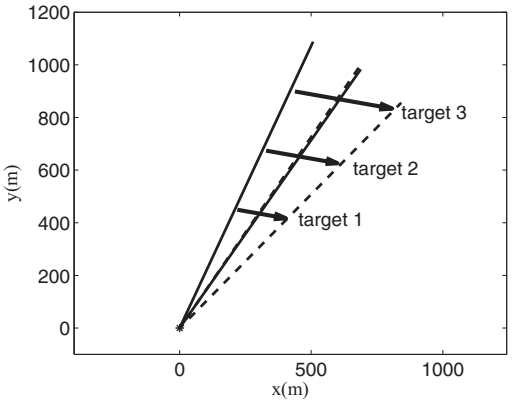


Fig. 6. Test scenario with two adjacent cameras and three targets (the two cameras, at 1 m apart, appear overlapped due to the necessary scale of the figure).

Third, the updated state and its error covariance are computed by the sigma points $\mathbf{x}_{{\rm II},i}$ with $i = 1, \ldots, 2n_{\rm y}+1$

$$\hat{\mathbf{x}}_{\rm II} = \sum_{i=1}^{2n_{\rm y}+1} w_i \hat{\mathbf{x}}_{{\rm II},i}, \tag{64}$$

$$\mathbf{P}_{\rm II} = \sum_{i=1}^{2n_{\rm y}+1} w_i (\hat{\mathbf{x}}_{{\rm II},i} - \hat{\mathbf{x}}_{\rm II})(\hat{\mathbf{x}}_{{\rm II},i} - \hat{\mathbf{x}}_{\rm II})'. \tag{65}$$

The fMLIT-UGHF still has inverse matrix operation, but less occurrence than the fMLIT-EGHF.

F. Full Maximum Likelihood Information Transfer using Unscented Guass–Helmert Filter with approximation (fMLIT-UGHFapp)

This algorithm is an approximation version of the fMLIT-UGHF. It removes the iteration process of the fMLIT-UGHF in (63). The sigma points of $\hat{\mathbf{x}}_{\rm II}$ are the initial ones $\hat{\mathbf{x}}_{{\rm II},i}^0$. If the initial sigma points are accurate enough, this approximation will not affect the final accuracy much. Furthermore, this approximation version has no matrix inversion operations, so the numerical issues due to the inversion of possibly ill-conditioned matrices are avoided. It also more efficient as it has no iteration process.

IV. SIMULATION RESULTS

This section evaluates the performance of the six algorithms described in Section III. The test scenario is shown in Fig. 6. Two adjacent cameras 1 and 2 are located at (0, 0, 50) m and (1, 0, 50) m, respectively, 1 m apart. Each camera has a field of view of 10° and 5.6° horizontal and vertical, respectively. The pointing angles of cameras 1 and 2 are 30° and 38° (clockwise from true North), respectively. Both of them are looking 2° up in elevation (from the horizontal). Their measurement error standard deviations of azimuth and elevation are $\sigma_b = \sigma_e = 0.046$ mrad, which is equivalent to 1 pixel

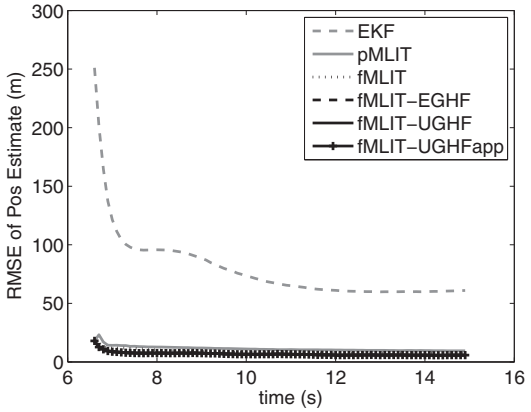


Fig. 7. The RMSE of position estimates versus time from 100 runs for the six algorithms after camera 2 starts to detect **target 1** (with starting range 500 m).

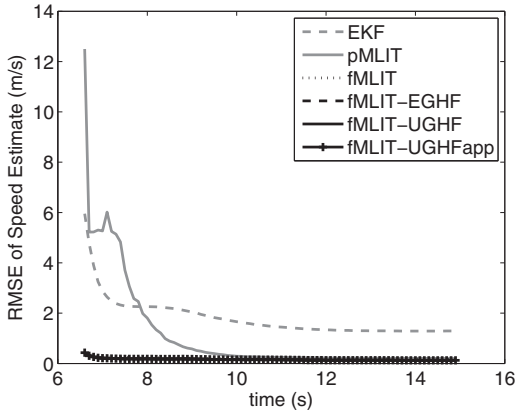


Fig. 8. The RMSE of speed estimates versus time from 100 runs for the six algorithms after camera 2 starts to detect **target 1** (with starting range 500 m).

(the cameras are assumed to have 8 megapixels). The two cameras provide their detections every 0.1 s. We simulate the targets 1, 2, and 3 with starting ranges at 500 m, 750 m, and 1,000 m, respectively. They move across the two cameras horizontally with the same heading 100° (clockwise from true North) and the same constant speed of 12.5 m/s. The observation durations for the targets 1, 2, and 3 are 15 s, 22 s, and 30 s, respectively. A near-range target has better observability than a far-range target, as the two LOSs from the two cameras have larger angle between them when the target crosses the cameras. So target 1 should have better estimation accuracy than target 2, and target 3 is the worst. The PSD in (9) is set as (see [2] for the units)

$$q_{\rm p} = 0.01^2\,{\rm m}^2/{\rm s}^3. \tag{66}$$

The velocity error standard deviations in (50) are set as

$$\sigma_{\dot{x}} = \sigma_{\dot{y}} = \sigma_{\dot{z}} = 10\,{\rm m/s}. \tag{67}$$

The scalar κ in (59)–(62) is set to 1. We will present the estimation accuracy using the root mean square errors (RMSE) of the position and speed (magnitude of the velocity vector), the statistical consistency analysis using the normalized estimation error squared (NEES) [1], and the algorithms' performances when the probability of detection (PD) is less than unity in sequel.

A. Estimation Accuracy

The position and velocity estimates from 100 Monte Carlo runs for each targets will be presented in this subsection.

First, we present the estimation accuracy of the six algorithms for target 1 with starting range 500 m. The position and speed estimate RMSEs versus time from 100 Monte Carlo runs are shown in Figs. 7 and 8, respectively. The time starts from the moment camera 2 starts detecting the target (since the state is not fully observable before that). It can be seen that the EKF and pMLIT are obviously worse than the other four fMLIT algorithms, as they do not take all constrains into consideration when the target crosses the cameras' FOVs. We remove these two algorithms from the performance figures, so that the other algorithms' performance can be enlarged and seen clearly. Since the EKF and pMLIT have poor performance clearly, their performance will not be presented in the sequel. The position and speed RMSEs versus time for the four fMLIT algorithms are shown in Figs. 9 and 10, respectively. We can see the fMLIT is worse than the other three algorithms. This is because the fMLIT is an approximate solution to compute the position and velocity separately, the cross-correlation of the estimation error between the position and velocity is totally ignored, whereas, the other three are solutions to the GH model. We do not observe much differences in accuracy among the fMLIT-EGHF, fMLIT-UGHF, and fMLIT-UGHFapp. It means the nonlinearity and errors in the problem are small enough, so that the first-order Taylor expansion (in the fMLIT-EGHF) and unscented transform (in the fMLIT-UGHF and fMLIT-UGHFapp) have very similar accuracy. Also, since target 1 is the nearest with the best observability, it does not cause numerical issues with ill-conditioned ma-

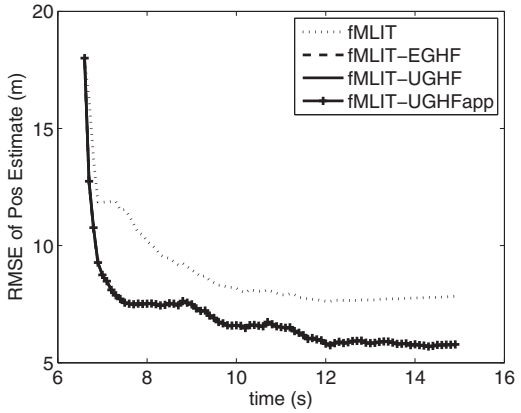


Fig. 9. The RMSE of position estimates versus time from 100 runs for the four fMLIT algorithms after camera 2 starts to detect **target 1** (with starting range 500 m).

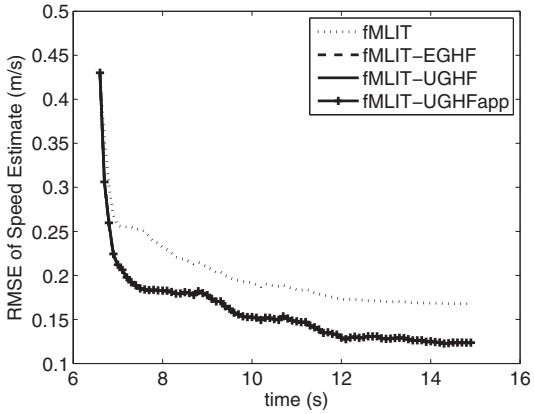


Fig. 10. The RMSE of speed estimates versus time from 100 runs for the four fMLIT algorithms after camera 2 starts to detect **target 1** (with starting range 500 m).

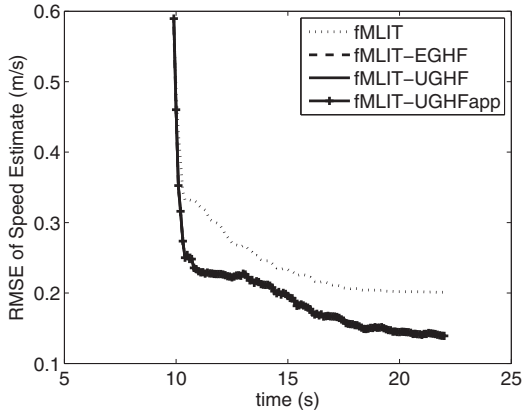


Fig. 12. The RMSE of speed estimates versus time from 100 runs for the four fMLIT algorithms after camera 2 starts to detect **target 2** (with starting range 750 m).

trices in the filtering process, so that the three GHFs all perform well. The similar results of the fMLIT-UGHF and fMLIT-UGHFapp also show the initial sigma points $\hat{\mathbf{x}}^0_{\mathrm{II,i}}$ in (63) are accurate enough, as similar accuracies are obtained with or without the iteration given in (63).

Second, we show the estimation accuracy of target 2 with starting range 750 m. The position and speed RMSEs versus time for the four fMLIT algorithms are shown in Figs. 11 and 12, respectively. Similarly to target 1, the fMLIT is worse than the other three GHFs, and the three GHFs have very similar results.

Next, we present the estimation performance for target 3 with starting range 1,000 m. Fig. 13 shows the position RMSEs versus time for the four fMLIT algorithms. It can be seen that the fMLIT-EGHF diverges. Divergence happened after the across-camera event in one run only. This is due to the state estimate error coveriance $\mathbf{P}_{\mathrm{II}}$ in (54), which is not positive-definite in the fMLIT-EGHF estimation. We investigated the condition number of $[\mathbf{B}\mathbf{P}_{\mathrm{y}}\mathbf{B}']$ in (54). It is 1×10^{18}. The inversion operation on such an ill-conditioned matrix encountered numerical issues, and caused $\mathbf{P}_{\mathrm{II}}$ to lose its positive definiteness. This resulted in estimation divergence subsequently. Target 3 is at the farthest range among the three targets, and its observability is marginal. The two LOSs of the two cameras are nearly parallel when the target crosses between the cameras. This leads to ill-conditioned matrices during the fMLIT-EGHF iteration process. We remove the fMLIT-EGHF, and show the position and speed estimate RMSEs for the remaining three algorithms in Figs. 14 and 15, respectively. It can be seen that the fMLIT-UGHF and fMLIT-UGHFapp have similar accuracy, and fMLIT is worse. We also investigate $\mathbf{P}_{\mathrm{II}}$ in fMLIT-UGHF. It is positive-definite. This is because $\mathbf{P}_{\mathrm{II}}$ is computed with sigma points using (65), and its positive-definiteness is maintained. However, the sigma points computation also has an inversion operation in (63). The condition number of the matrix $[\mathbf{A}'\mathbf{A}]$ to be inverted in (63) reached 1×10^{18}. Thus, the fMLIT-UGHF has a potential risk of numerical issues, although not as much as the fMLIT-EGHF.

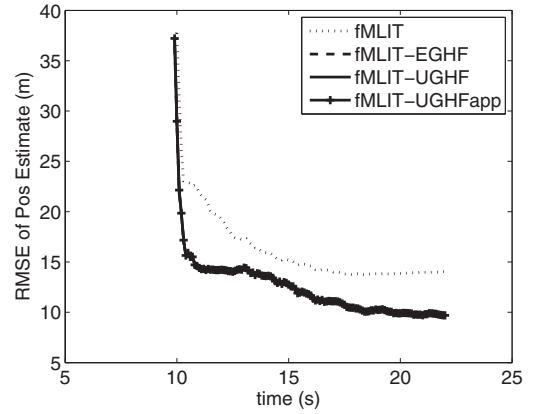


Fig. 11. The RMSE of position estimates versus time from 100 runs for the four fMLIT algorithms after camera 2 starts to detect **target 2** (with starting range 750 m).

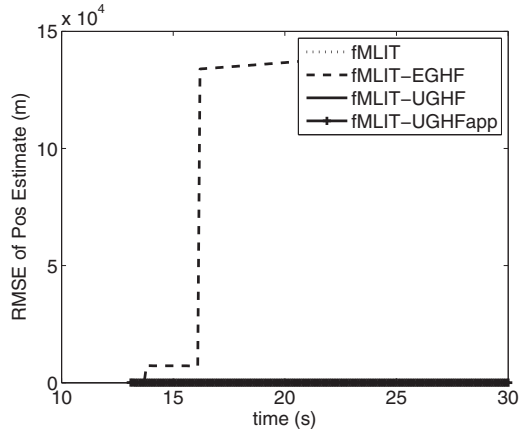


Fig. 13. The RMSE of position estimates versus time from 100 runs for the four fMLIT algorithms after camera 2 starts to detect **target 3** (with starting range 1,000 m).

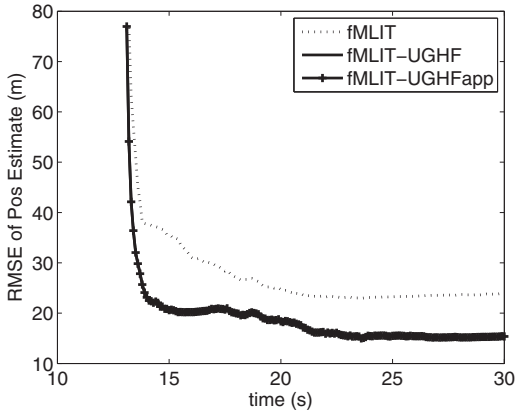


Fig. 14. The RMSE of position estimates versus time from 100 runs for the fMLIT, fMLIT-UGHF, and fMLIT-UGHFapp after camera 2 starts to detect **target 3** (with starting range 1,000 m).

B. Statistical Consistency Analysis

The statistical consistency analysis is conducted using the NEES [1] at the across-camera event and is computed by

$$\epsilon^i(t_k) = \tilde{\mathbf{x}}^i(t_k)' \mathbf{P}^i(t_k)^{-1} \tilde{\mathbf{x}}^i(t_k), \tag{68}$$

where $i = 1 \ldots 100$ is the run index,

$$\tilde{\mathbf{x}}^i(t_k) = \mathbf{x}(t_k) - \hat{\mathbf{x}}^i(t_k), \tag{69}$$

and $\hat{\mathbf{x}}^i(t_k)$ and $\mathbf{x}(t_k)$ are the ith run state estimate and ground truth at time t_k, respectively, assuming the target crosses between the cameras at time t_k. The NEESs of 100 runs are recorded. The NEES of the state (with dimension 6) is a 6 degrees of freedom chi-square random variable. Its two-sided 95% probability region is [1.24, 14.45]. The estimation is statistically consistent, if 95% of NEESs are within this interval.

Figs. 16–18 show the NEESs versus the run index at across-cameras for targets 1–3, respectively. The numbers of NEESs that are outside the 95% region for the four fMLIT algorithms are counted and listed in Table I. It can be seen, the fMLIT is not statically consistent for

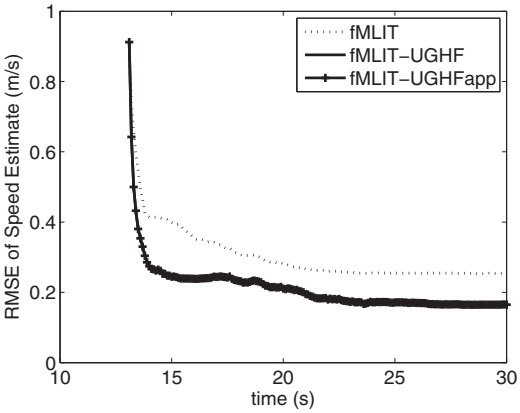


Fig. 15. The RMSE of speed estimates versus time from 100 runs for the fMLIT, fMLIT-UGHF, and fMLIT-UGHFapp after camera 2 starts to detect **target 3** (with starting range 1,000 m).

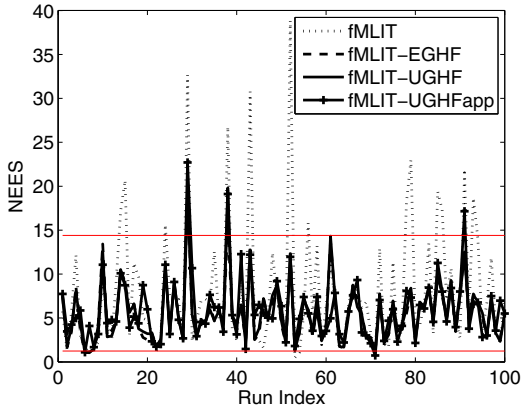


Fig. 16. The across-cameras NEES of **target 1** (with starting range 500 m) in 100 runs for the four fMLIT algorithms.

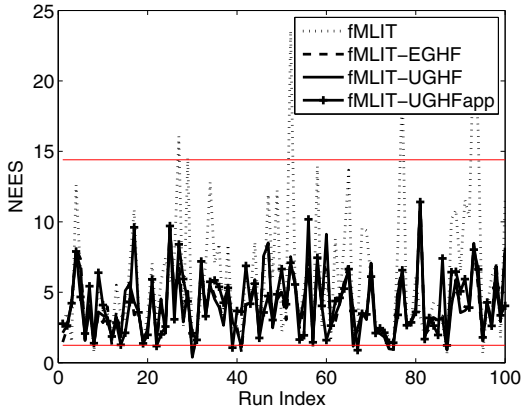


Fig. 17. The across-cameras NEES of **target 2** (with starting range 750 m) in 100 runs for the four fMLIT algorithms.

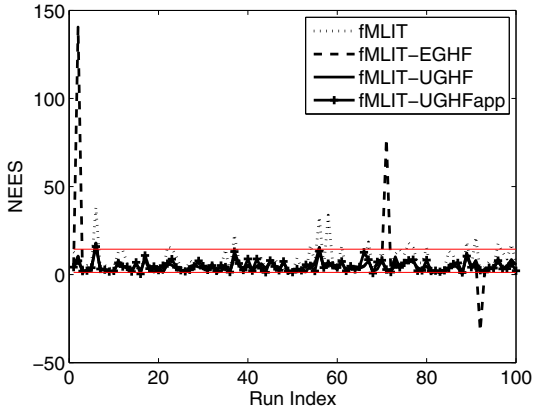


Fig. 18. The across-cameras NEES of **target 3** (with starting range 1,000 m) in 100 runs for the four fMLIT algorithms.

Table I
Number of NEES out of 95% boundary (100 runs)

Target	1	2	3
fMLIT	18	11	24
fMLIT-EGHF	5	5	11
fMLIT-UGHF	5	6	9
fMLIT-UGHFapp	5	4	4

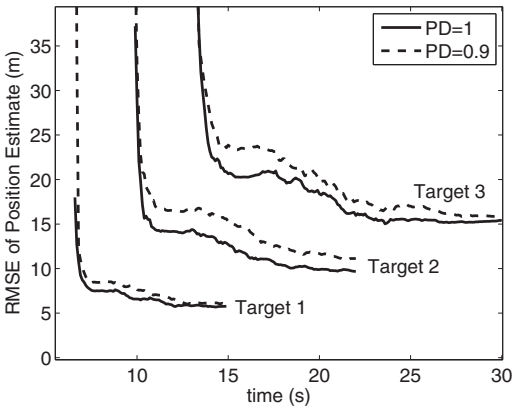


Fig. 19. RMSE of position estimate in 100 runs for the fMLIT-UGHF.

all three targets. The number of NEES outside the 95% region is 11 to 24, much larger than 5. For target 1, the three GHFs are statistically consistent, as they all have five NEESs outside the 95% region, i.e., they meet the criterion. For target 2, the three GHFs are also consistent. For target 3, only the fMLIT-UGHFapp is statistically consistent. The fMLIT-EGHF and fMLIT-UGHF do not meet the consistency criterion. The worse performance of the fMLIT-EGHF and fMLIT-UGHF is due to their numerical issues.

C. Performance under Imperfect Detection

The PD is usually imperfect in real camera applications. We investigate the estimation accuracy of the three GHFs (namely, fMLIT-EGHF, fMLTL-UGHF, and fMLTL-UGHFapp) when PD = 0.9 for the three targets shown in Fig. 6. The RMSE of position estimates in 100 Monte Carlo runs are studied. The fMLIT-EGHF diverges for all three targets when PD = 0.9. The results of the fMLIT-UGHF and fMLIT-UGHFapp are shown in Figs. 19 and 20, respectively. The corresponding RMSEs when PD = 1 are also shown in the figures for comparison. The position RMSEs increased 1–2 m when PD = 0.9 in both fMLIT-UGHF and fMLIT-UGHFapp. The fMLIT-UGHF and fMLIT-UGHFapp

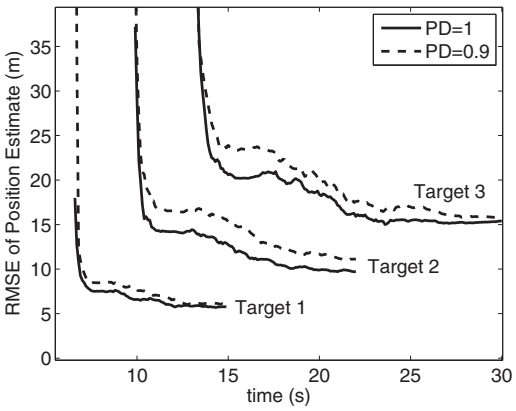


Fig. 20. RMSE of position estimate in 100 runs for the fMLIT-UGHFapp.

generate very similar results. Figs. 19 and 20 are shown separately because the results of the fMLIT-UGHF and fMLIT-UGHFapp would be indistinguishable on a combined figure at the same scale—they are, as in Figs. 9–15, practically the same.

In summary, it can be seen from the simulation results that the fMLIT-UGHFapp is the most robust algorithm. Its accuracy is similar to the fMLIT-UGHF and without numerical issues. Its estimation is consistent statistically. It also has lower computational cost than the fMLIT-UGHF as no iteration process is needed at each sigma point. The fMLIT-UGHF is good in theory, but is limited by numerical issues in practice. It is better than the fMLIT-EGHF which suffers from serious numerical problems. In general, the unscented transform (in the fMLIT-UGHF) should be better than the first-order Taylor approximation (in the fMLIT-EGHF) for a highly nonlinear model with large error. In this application, the nonlinearity and the degree of error do not affect their performance. However, we observed another disadvantage of the first-order Taylor approximation—it is easier to encounter numerical issues in the information transfer when the observablity is marginal. The other three algorithms (EKF, pMLIT, and fMLIT) are worse than those using GHFs.

V. CONCLUSIONS

In this paper, we formulated an implicit GHM to transfer the full state information when a target crosses between cameras. This model is based on the principle that the target heading estimated from the first camera is observable (assuming the target is in a NCV motion), so that the heading should be consistent when the target moves across cameras, and the speed is therefore proportional to the range from the cameras. These implicit constraints were added to the original measurement model at the crossover time as the GHM. We developed three GHFs, namely, fMLIT-EGHF, fMLIT-UGHF, and fMLIT-UGHFapp, as solutions for the GHM (with different approximations of the nonlinear constraint). The three GHFs outperform the other three algorithms (EKF, pMLIT, and fMLIT), which are not using the GHM, when numerical issues are not encountered. The three GHFs are statistically consistent when the target is near ($\leq$750 m) and PD = 1. However, when the target is at 1,000 m range, only fMLIT-UGHFapp is still statistically consistent, and the fMLIT-EGHF and fMLIT-UGHF cannot meet the consistency criterion due to numerical problems in the matrix inversion. The fMLIT-UGHFapp is an approximate implementation of the fMLIT-UGHF to avoid the matrix inversion. It has similar accuracy to the fMLIT-EGHF (when the latter does not diverge) and fMLIT-UGHF, and its estimates are statistically consistent. Therefore, the fMLIT-UGHFapp is the most robust algorithm in this problem.

To apply the proposed algorithm to real applications, other issues need to be addressed. First of all, cameras

should be calibrated accurately. Second, a suitable track to measurement association algorithm should be applied when multiple targets and false alarms exist. Third, maneuvering targets need to be handled. The GHM in this paper is based on the assumption of constant velocity motion. This assumption is only necessary in the neighborhood of the transfer between cameras. Based on our study, the velocity estimate from a single camera can converge within 1 s observation interval. If the target is not maneuvering for 1 s before camera crossing, the GHM is valid. Otherwise, the pMLIT without speed constraint should be applied. All these issues are separate topics in themselves and quite extensive. Further study on them will be conducted in the future.

APPENDIX A

OBSERVABILITY OF HEADING FROM A SEQUENCE OF ANGULAR MEASUREMENTS

This appendix proves that the target heading is observable for a constant velocity (CV) target from a sequence of angular measurements (azimuths and elevations) of a stationary sensor. The 3D problem can be simplified to a 2D problem in the plane formed by the target path (a straight line) and the sensor location point. The azimuth and elevation measurements can then be converted to bearing lines in this plane. If we can prove the target heading in this plane is observable, the heading is also observable in 3D Cartesian space.

Fig. 21 shows this 2D plane with the sensor at point O and three bearing lines OA, OB, and OC in time sequence with a fixed time interval T. Assuming the target path is on the straight line ABC, we have $|AB| = |BC|$ as the target is moving in CV. We will prove that the target heading is observable by the following two steps:

- At an arbitrary point A on the first bearing line, there is an unique straight path ABC with $|AB| = |BC|$ (namely, with two equal cuts by the three bearing lines).
- All the (CV target paths with two equal cuts (by the three bearing lines) are parallel.

These will prove that all possible target trajectories share the same unique heading.

First, we prove that if A is fixed, then ABC is the unique straight path with $|AB| = |BC|$ for the three LOS as above. We draw a line AD parallel to OC and it intersects OB at D. We prove $|AD| = |OC|$. Since the triangles ABD and OBC are similar and $|AB| = |BC|$, we have

$$\frac{|AB|}{|BC|} = \frac{|AD|}{|OC|} = 1, \tag{A1}$$

$$|AD| = |OC|. \tag{A2}$$

We then draw two arbitrary straight lines AB_1C_1 and AB_2C_2 below and above ABC, respectively. We will

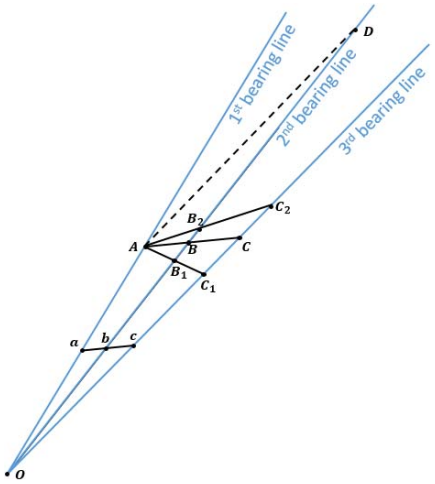


Fig. 21. Proof of the unique target heading obtained from three bearing lines for a constant velocity target.

prove these two arbitrary lines cannot meet the equalities (implied by CV assumption) $|AB_1| = |B_1C_1|$ or $|AB_2| = |B_2C_2|$. For AB_1C_1, we prove $|AB_1| > |B_1C_1|$ by

$$\frac{|AB_1|}{|B_1C_1|} = \frac{|AD|}{|OC_1|} > \frac{|AD|}{|OC|} = 1, \tag{A3}$$

$$\frac{|AB_1|}{|B_1C_1|} > 1, \tag{A4}$$

$$|AB_1| > |B_1C_1|. \tag{A5}$$

For AB_2C_2, we prove $|AB_2| < |B_2C_2|$ by

$$\frac{|AB_2|}{|B_2C_2|} = \frac{|AD|}{|OC_2|} < \frac{|AD|}{|OC|} = 1, \tag{A6}$$

$$\frac{|AB_2|}{|B_2C_2|} < 1, \tag{A7}$$

$$|AB_2| < |B_2C_2|. \tag{A8}$$

Since any arbitrary line below or above ABC does not have two equal cuts by the three bearing lines, it follows that ABC is unique for a CV target starting at A.

Second, we will prove all the possible target trajectories with equal cuts are parallel. Let's draw another arbitrary line abc parallel to ABC, and a is an arbitrary point besides A on the first bearing line. We will prove $|ab| = |bc|$ by

$$\frac{|ab|}{|AB|} = \frac{|Ob|}{|OB|}, \tag{A9}$$

$$|ab| = \frac{|Ob||AB|}{|OB|}, \tag{A10}$$

$$\frac{|bc|}{|BC|} = \frac{|Ob|}{|OB|}, \tag{A11}$$

$$|bc| = \frac{|Ob||BC|}{|OB|} = \frac{|Ob||AB|}{|OB|} = |ab|. \tag{A12}$$

Based on our previous proof, the straight line with two equal cuts is unique if a is fixed. Thus abc is the unique line starting at a with two equal cuts. Since abc is parallel to ABC, they share the same heading. Furthermore, point a is an arbitrary point on the first bearing line, and this proves that all possible CV trajectories (with two equal cuts) are parallel and share the same heading. This proves that the heading is observable.

Note the assumption of the above proof that the three bearing lines are different. If a target approaching to the sensor (or moving away from the sensor) directly, the three bearings coincide. The heading angle is known, but cannot differentiate if the target is approaching or moving away from O. That needs additional information, such as target size increasing/decreasing in a sequence of images (or intensity for point detection), to tell the direction.

APPENDIX B

JACOBIANS IN THE fMLIT-EGHF AND fMLIT-UGHF

The Jacobians $\mathbf{A}$ and $\mathbf{B}$ used in the fMLIT-EGHF and fMLIT-UGHF are derived below.

$$\mathbf{A} = \frac{\partial \mathbf{g}(\mathbf{x}_{\mathrm{II}}, \mathbf{y})}{\partial \mathbf{x}_{\mathrm{II}}} \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & 0 & 0 \\ A_{31} & A_{32} & A_{33} & 0 & 0 & 0 \\ A_{41} & A_{42} & A_{43} & r_{\mathrm{I}} & 0 & 0 \\ A_{51} & A_{52} & A_{53} & 0 & r_{\mathrm{I}} & 0 \\ A_{61} & A_{62} & A_{63} & 0 & 0 & r_{\mathrm{I}} \\ A_{71} & A_{72} & 0 & 0 & 0 & 0 \\ A_{81} & A_{82} & A_{83} & 0 & 0 & 0 \end{bmatrix}, \tag{B1}$$

$$\mathbf{B} = \frac{\partial \mathbf{g}(\mathbf{x}_{\mathrm{II}}, \mathbf{y})}{\partial \mathbf{y}} \begin{bmatrix} B_{11} & B_{12} & B_{13} & 0 & 0 & 0 & 0 & 0 \\ B_{21} & B_{22} & B_{23} & 0 & 0 & 0 & 0 & 0 \\ B_{31} & B_{32} & B_{33} & 0 & 0 & 0 & 0 & 0 \\ B_{41} & B_{42} & B_{43} & -r_{\mathrm{II}} & 0 & 0 & 0 & 0 \\ B_{51} & B_{52} & B_{53} & 0 & -r_{\mathrm{II}} & 0 & 0 & 0 \\ B_{61} & B_{62} & B_{63} & 0 & 0 & -r_{\mathrm{II}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \tag{B2}$$

with

$$A_{11} = r_{\mathrm{I}} - (x_{\mathrm{I}} - x_{\mathrm{s}_1})(x_{\mathrm{II}} - x_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B3}$$

$$A_{12} = -(x_{\mathrm{I}} - x_{\mathrm{s}_1})(y_{\mathrm{II}} - y_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B4}$$

$$A_{13} = -(x_{\mathrm{I}} - x_{\mathrm{s}_1})(z_{\mathrm{II}} - z_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B5}$$

$$A_{21} = -(y_{\mathrm{I}} - y_{\mathrm{s}_1})(x_{\mathrm{II}} - x_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B6}$$

$$A_{22} = r_{\mathrm{I}} - (y_{\mathrm{I}} - y_{\mathrm{s}_1})(y_{\mathrm{II}} - y_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B7}$$

$$A_{23} = -(y_{\mathrm{I}} - y_{\mathrm{s}_1})(z_{\mathrm{II}} - z_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B8}$$

$$A_{31} = -(z_{\mathrm{I}} - z_{\mathrm{s}_1})(x_{\mathrm{II}} - x_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B9}$$

$$A_{32} = -(z_{\mathrm{I}} - z_{\mathrm{s}_1})(y_{\mathrm{II}} - y_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B10}$$

$$A_{33} = r_{\mathrm{I}} - (z_{\mathrm{I}} - z_{\mathrm{s}_1})(z_{\mathrm{II}} - z_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B11}$$

$$A_{41} = -\dot{x}_{\mathrm{I}}(x_{\mathrm{II}} - x_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B12}$$

$$A_{42} = -\dot{x}_{\mathrm{I}}(y_{\mathrm{II}} - y_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B13}$$

$$A_{43} = -\dot{x}_{\mathrm{I}}(z_{\mathrm{II}} - z_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B14}$$

$$A_{51} = -\dot{y}_{\mathrm{I}}(x_{\mathrm{II}} - x_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B15}$$

$$A_{52} = -\dot{y}_{\mathrm{I}}(y_{\mathrm{II}} - y_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B16}$$

$$A_{53} = -\dot{y}_{\mathrm{I}}(z_{\mathrm{II}} - z_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B17}$$

$$A_{61} = -\dot{z}_{\mathrm{I}}(x_{\mathrm{II}} - x_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B18}$$

$$A_{62} = -\dot{z}_{\mathrm{I}}(y_{\mathrm{II}} - y_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B19}$$

$$A_{63} = -\dot{z}_{\mathrm{I}}(z_{\mathrm{II}} - z_{\mathrm{s}_1})/r_{\mathrm{II}}, \tag{B20}$$

$$A_{71} = \frac{y_{\mathrm{II}} - y_{\mathrm{s}_2}}{(x_{\mathrm{II}} - x_{\mathrm{s}_2})^2 + (y_{\mathrm{II}} - y_{\mathrm{s}_2})^2}, \tag{B21}$$

$$A_{72} = -\frac{x_{\mathrm{II}} - x_{\mathrm{s}_2}}{(x_{\mathrm{II}} - x_{\mathrm{s}_2})^2 + (y_{\mathrm{II}} - y_{\mathrm{s}_2})^2}, \tag{B22}$$

$$A_{81} = -\frac{(x_{\mathrm{II}} - x_{\mathrm{s}_2})(z_{\mathrm{II}} - z_{\mathrm{s}_2})}{\sqrt{(x_{\mathrm{II}} - x_{\mathrm{s}_2})^2 + (y_{\mathrm{II}} - y_{\mathrm{s}_2})^2}} \times \frac{1}{(x_{\mathrm{II}} - x_{\mathrm{s}_2})^2 + (y_{\mathrm{II}} - y_{\mathrm{s}_2})^2 + (z_{\mathrm{II}} - z_{\mathrm{s}_2})^2}, \tag{B23}$$

$$A_{82} = -\frac{(y_{\mathrm{II}} - y_{s_2})(z_{\mathrm{II}} - z_{s_2})}{\sqrt{(x_{\mathrm{II}} - x_{s_2})^2 + (y_{\mathrm{II}} - y_{s_2})^2}} \times \frac{1}{(x_{\mathrm{II}} - x_{s_2})^2 + (y_{\mathrm{II}} - y_{s_2})^2 + (z_{\mathrm{II}} - z_{s_2})^2}, \tag{B24}$$

$$A_{83} = \frac{\sqrt{(x_{\mathrm{II}} - x_{s_2})^2 + (y_{\mathrm{II}} - y_{s_2})^2}}{(x_{\mathrm{II}} - x_{s_2})^2 + (y_{\mathrm{II}} - y_{s_2})^2 + (z_{\mathrm{II}} - z_{s_2})^2}, \tag{B25}$$

$$B_{11} = (x_{\mathrm{II}} - x_{s_1})(x_{\mathrm{I}} - x_{s_1})/r_{\mathrm{I}} - r_{\mathrm{II}}, \tag{B26}$$

$$B_{12} = (x_{\mathrm{II}} - x_{s_1})(y_{\mathrm{I}} - y_{s_1})/r_{\mathrm{I}}, \tag{B27}$$

$$B_{13} = (x_{\mathrm{II}} - x_{s_1})(z_{\mathrm{I}} - z_{s_1})/r_{\mathrm{I}}, \tag{B28}$$

$$B_{21} = (y_{\mathrm{II}} - y_{s_1})(x_{\mathrm{I}} - x_{s_1})/r_{\mathrm{I}}, \tag{B29}$$

$$B_{22} = (y_{\mathrm{II}} - y_{s_1})(y_{\mathrm{I}} - y_{s_1})/r_{\mathrm{I}} - r_{\mathrm{II}}, \tag{B30}$$

$$B_{23} = (y_{\mathrm{II}} - y_{s_1})(z_{\mathrm{I}} - z_{s_1})/r_{\mathrm{I}}, \tag{B31}$$

$$B_{31} = (z_{\mathrm{II}} - z_{s_1})(x_{\mathrm{I}} - x_{s_1})/r_{\mathrm{I}}, \tag{B32}$$

$$B_{32} = (z_{\mathrm{II}} - z_{s_1})(y_{\mathrm{I}} - y_{s_1})/r_{\mathrm{I}}, \tag{B33}$$

$$B_{33} = (z_{\mathrm{II}} - z_{s_1})(z_{\mathrm{I}} - z_{s_1})/r_{\mathrm{I}} - r_{\mathrm{II}}, \tag{B34}$$

$$B_{41} = \dot{x}_{\mathrm{II}}(x_{\mathrm{I}} - x_{s_1})/r_{\mathrm{I}}, \tag{B35}$$

$$B_{42} = \dot{x}_{\mathrm{II}}(y_{\mathrm{I}} - y_{s_1})/r_{\mathrm{I}}, \tag{B36}$$

$$B_{43} = \dot{x}_{\mathrm{II}}(z_{\mathrm{I}} - z_{s_1})/r_{\mathrm{I}}, \tag{B37}$$

$$B_{51} = \dot{y}_{\mathrm{II}}(x_{\mathrm{I}} - x_{s_1})/r_{\mathrm{I}}, \tag{B38}$$

$$B_{52} = \dot{y}_{\mathrm{II}}(y_{\mathrm{I}} - y_{s_1})/r_{\mathrm{I}}, \tag{B39}$$

$$B_{53} = \dot{y}_{\mathrm{II}}(z_{\mathrm{I}} - z_{s_1})/r_{\mathrm{I}}, \tag{B40}$$

$$B_{61} = \dot{z}_{\mathrm{II}}(x_{\mathrm{I}} - x_{s_1})/r_{\mathrm{I}}, \tag{B41}$$

$$B_{62} = \dot{z}_{\mathrm{II}}(y_{\mathrm{I}} - y_{s_1})/r_{\mathrm{I}}, \tag{B42}$$

$$B_{63} = \dot{z}_{\mathrm{II}}(z_{\mathrm{I}} - z_{s_1})/r_{\mathrm{I}}, \tag{B43}$$

where r_{I} and r_{II} are given in (30) and (31), respectively (see Fig. 3).

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software. New York, NY, USA: Wiley, 2001.

[2] Y. Bar-Shalom, P. K. Willett, and X. Tian
Tracking and Data Fusion: A Handbook of Algorithms. Storrs, CT, USA: YBS Publishing, 2011.

[3] L. Chen, H. Ai, C. Shang, Z. Zhuang, and B. Bai
"Online multi-object tracking with convolutional neural networks,"
In IEEE Int. Conf. Image Process, Sep. 2017, pp. 645–649.

[4] W. Forstner
"On weighting and choosing constraints for optimally reconstructing the geometry of image triplets,"
In the 6th Eur. Conf. Comput. Vision, Jun. 2000, pp. 669–684.

[5] M. A. Haseeb, J. Guan, D. Ristic-Durrant, and A. Graser
"DisNet: A novel method for distance estimation from monocular camera,"
In 10th Planning Perception Navigation Intell. Vehicles, Oct. 2018.

[6] C. Jauffret, D. Pillon, and A. Pignol
"Bearings-only TMA without observer maneuver,"
In 11th Int. Conf. Inf. Fusion, Jul. 2008.

[7] K. Kanatani and H. Niitsuma
"Optimal computation of 3-D similarity: Gauss-Newton vs. Gauss-Helmert,"
Comput. Statist. Data Anal., vol. 56, no. 4, pp. 4470–4483, Dec. 2012.

[8] T. Klinger, F. Rottensteiner, and C. Heipke
"Probabilistic multi-person localisation and tracking in image sequences,"
ISPRS J. Photogrammetry Remote Sens., vol. 127, pp. 73–88, May 2017.

[9] K.-H. Lee and J.-N. Hwang
"On-road pedestrian tracking across multiple driving recorders,"
IEEE Trans. Multimedia, vol. 17, no. 9, pp. 1429–1438, Sep. 2015.

[10] E. M. Mikhail and F. E. Ackermann
Observations and Least Squares. New York, NY, USA: Harper and Row, 1976.

[11] S. Y. Nam and G. P. Joshi
"Unmanned aerial vehicle localization using distributed sensors,"
Int. J. Distrib. Sensor Netw., vol. 13, no. 9, pp. 1–8, Aug. 2017.

[12] F. Neitzel
"Generalization of total least-squares on example of unweighted and weighted 2D similarity transformations,"
J. Geodesy, vol. 84, no. 12, pp. 751–762, Dec. 2010.

[13] C. Perwass, C. Gebken, and G. Sommer
"Geometry and kinematics with uncertain data,"
In the 9th Eur. Conf. Comput. Vis. Part I, May 2006, pp. 225–237.

[14] B. Schaffrin and K. Snow
"Total least-squares regularization of Tykhonov type and an ancient racetrack in Corinth,"
Linear Algebra Appl., vol. 432, no. 8, pp. 2061–2076, Apr. 2010.

[15] S. Soatto, R. Frezza, and P. Perona
"Motion estimation via dynamic vision,"
IEEE Trans. Autom. Control, vol. 41, no. 3, pp. 393–413, Mar. 1996.

[16] Z. Tang, J.-N. Hwang, Y.-S. Lin, and J.-H. Chuang
"Multiple-kernel adaptive segmentation and tracking (MAST) for robust object tracking,"

In IEEE Int. Conf. Acoust., Speech Signal Process., pp. 1115–1119, Mar. 2016.

[17] Z. Tang and J.-N. Hwang
"MOANA: An online learned adaptive appearance model for robust multiple object tracking in 3D,"
IEEE Access, vol. 7, no. 1, pp. 31934–31945, Mar. 2019.

[18] L. Wen, Z. Lei, M.-C. Chang, H. Qi, and S. Lyu
"Multi-camera multitarget tracking with space-time-view hyper-graph,"
Int. J. Comput. Vis., vol. 122, no. 2, pp. 313–333, Sep. 2016.

[19] M. Yang, Y. Wu, and Y. Jia
"A hybrid data association framework for robust online multi-object tracking,"
IEEE Trans. Image Process., vol. 26, no. 12, pp. 5667–5679, Dec. 2017.

[20] R. Yang, Y. Bar-Shalom, H. A. Jack Huang, and G. W Ng
"Interacting multiple model unscented Gauss-Helmert filter for bearings-only tracking with state-dependent propagation delay,"
In 17th Int. Conf. Inf. Fusion, Jul. 2014.

[21] R. Yang, Y. Bar-Shalom, H. A. Jack Huang, and G. W. Ng
"UGHF for acoustic tracking with state-dependent propagation delay,"
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 3, pp. 1747–1761, Jul. 2015.

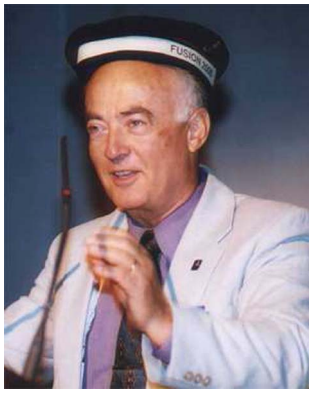
[22] R. Yang, Y. Bar-Shalom, and G. W. Ng
"Bearings-only tracking with fusion from heterogenous passive sensors: ESM/EO and acoustic,"
In the 18th Int. Conf. Inf. Fusion, Jul. 2015.

[23] R. Yang, Y. Bar-Shalom, and G. W. Ng
"Bearings-only tracking with fusion from heterogenous passive sensors: ESM/EO and acoustic,"
J. Adv. Inf. Fusion, vol. 12, no. 3, pp. 3–17, Jul. 2017.

[24] R. Yang, Y. Bar-Shalom, C. Jauffret, A.-C. Perez, and G. W. Ng
"Maneuvering target tracking using continuous wave bistatic sonar with propagation delay,"
J. Adv. Inf. Fusion, vol. 13, no. 1, pp. 36–49, Jun. 2018.

[25] T. Yuan, Y. Bar-Shalom, and X. Tian
Heterogeneous track-to-track fusion.
In 14th Int. Conf. Inf. Fusion, Jul. 2011.

Rong Yang received the B.E. degree in information and control from Xi'an Jiao Tong University, China, in 1986, the M.Sc. degree in electrical engineering from National University of Singapore, in 2000, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. She is currently a Principal Member of Technical Staff at DSO National Laboratories, Singapore. Her research interests include passive tracking, low observable target tracking, GMTI tracking, hybrid dynamic estimation, and data fusion. She was Publicity and Publication Chair of FUSION 2012 and received the FUSION 2014 Best Paper Award (First runner up).



Yaakov Bar-Shalom (F'84) received the B.S. and M.S. degrees from the Technion, in 1963 and 1967, and the Ph.D. degree from Princeton University, in 1970, all in EE. Currently he is Board of Trustees Distinguished Professor in the ECE Dept. and Marianne E. Klewin Professor at the University of Connecticut. His current research interests are in estimation theory, target tracking, and data fusion. He has published over 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 Associate Editor of 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 during 2004–2013. Since 1995, he is a Distinguished Lecturer of the IEEE AESS. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE TAE Systems 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.