Fuse-before-Track in Large Sensor Networks

STEFANO CORALUPPI
MARCO GUERRIERO
PETER WILLETT
CRAIG CARTHEL

Recent years have seen a trend towards unmanned multi-sensor surveillance networks with large numbers of cheap and limited-performance sensors. While these networks hold significant potential for surveillance, it is of interest to address fundamental limitations in large-scale implementations. We first introduce a simple analytical tracker performance model. Analysis of this model suggests that scan-based tracking performance improves with increasing numbers of sensors, but only to a certain point beyond which degradation is observed. Correspondingly, we address model-based optimization of the local sensor detection threshold and the number of sensors. Next, we propose a two-stage tracking approach (*fuse-before-track*) as a possible approach to overcoming the difficulties in large-sensor surveillance, and we illustrate promising performance results with simulated surveillance data.

Manuscript received December 18, 2008; first revision July 13, 2009, second revision October 1, 2009; released for publication October 20, 2009.

Refereeing of this contribution was handled by Ben Slocumb.

Authors' addresses: Stefano Coraluppi and Craig Carthel, NATO Undersea Research Centre (NURC), Viale S. Bartolomeo 400, 19126 La Spezia, Italy, E-mail: coraluppi@nurc.nato.int, carthel@nurc.nato.int; Marco Guerriero and Peter Willett, University of Connecticut–ECE Department, Storrs, CT 06269, E-mail: marco.guerriero@engr.uconn.edu, willett@engr.uconn.edu

1. INTRODUCTION

While multi-sensor systems hold great potential for surveillance performance, the technical challenges are significant, and include the need for effective calibration as well as a statistically-valid characterization of environmental uncertainties and contact measurement errors. Additionally, automatic tracking and fusion processing with active sensors must contend with high false contact rates and target fading effects. Issues in multi-sensor surveillance and numerous design approaches are discussed in [4, 9, 24].

In [12–13], we present model-based, simulation-based, and sea-trial tracking performance results with a track-oriented, modular multi-hypothesis tracking scheme. Of particular interest is the tradeoff between centralized and multi-stage processing: we have found that, when faced with significant target fading effects and for modest false contact rates, distributed processing can outperform centralized processing. This somewhat surprising result is based on the fundamental sub-optimality of all tracking algorithms that must contend with measurement origin uncertainty. This explains the seeming contradiction with results in the nonlinear filtering and distributed detection literature, in particular the well-known optimality of centralized processing schemes [9, 25].

Ultimately, for sufficiently low-SNR target scenarios, effective real-time automatic tracking is extremely challenging regardless of the choice of data processing architecture. One approach is to relax the real-time requirement, and to leverage powerful batch processing techniques [3]. However, such schemes are not easily amenable to real-time surveillance requirements, and generally assume non-maneuvering targets. An alternative approach in challenging scenarios is to consider enlarging the surveillance network, possibly through bootstrapping approaches that include sub-band processing techniques [19], whereby a sensor is effectively "replaced" with a number of slightly-degraded sensors.

The latter approach (enlarging the surveillance network) implicitly assumes that an increased number of like-performing, calibrated, and registered sensors are always to be preferred, i.e., more sensors are always better than fewer. Is this true in general or are there performance limits as the number of sensors becomes large? This is the issue that we address in this paper.

We start by introducing in Section 2 a simple analytical model for tracker performance. We study tracker performance as a function of local detection threshold, number of sensors, and track management criteria. The model supports the conclusion that there are performance bounds on achievable performance in large sensor networks.

Can we do better if we consider a more complex, multi-stage processing architecture? In Section 3, we describe the *fuse-before-track* (FbT) architecture and

provide motivation for its use in large sensor networks. The key insight that motivates the FbT architecture is that it couples the advantages of batch processing in the fusion step, followed by the advantages of scan-based processing (real-time processing with a maneuvering target model) in the tracking step.

It is important to note that we do not argue that FbT will outperform a *sufficiently complex* centralized processing scheme. Rather, our claim is that multistage processing with a relatively *simple* tracking module can achieve good performance results. Further, as we will see in the Monte Carlo study, this performance is achieved with a significantly lower computational effort than in centralized (single-stage) processing.

The first stage in the FbT architecture is a static fusion (or contact fusion) stage [4, 9]. In Section 4, we study perhaps the simplest approach to contact fusion that was suggested in [19] and is known as contact sifting. We analytically characterize the performance of this stage of processing, and experimentally validate the performance model. We also briefly discuss an alternative approach to contact fusion.

Section 5 presents a Monte Carlo study of multisensor tracking performance for a representative multitarget surveillance scenario. The results suggest that the FbT architecture has merit and deserves further attention by the target tracking community. Section 6 provides conclusions and directions for future work. This paper is an extended and improved version of [14–16].

2. TRACKER PERFORMANCE MODELING

Tracker performance modelling is addressed at length in [4, 8]. Extensions that address target fading effects and distributed tracking architectures are given in [10, 12–13]. For our purposes here, we introduce a simple tracker performance model that identifies a compact relationship between scan rate and performance. Scan rate is directly proportional to the number of sensors and, thus, the model will support the subsequent analysis on performance as a function of the number of sensors.

2.1. Tracker Model

Modeling parameters:

- *Target*: kinematic "nearly constant position" motion model in two dimensions with maneuvering index q $\mathrm{m^2 s^{-1}}$; fixed target SNR d;
- *Sensor*: each with scan every Δt sec; positional measurements with covariance R; surveillance region of size A $\mathrm{m^2}$, detection cells of size C $\mathrm{m^2}$, detection threshold D;
- *Tracker*: declare track on N consecutive associated detections, terminate track on K consecutive coasts (missed updates), association probability gate P_{G} and gating parameter γ (with two-dimensional measurements, $\gamma = 9.2$ corresponds to $P_{\mathrm{G}} = 0.99$; see details in [8]).

The following derived quantities are of interest.

- *Detection probability* P_{D}, where we assume Rayleigh-distributed amplitude statistics [9]:

$$P_{\mathrm{D}} = \exp\left(-\frac{D}{1+d}\right). \tag{1}$$

- *False alarm density per square meter*, where we again assume Rayleigh-distributed amplitude statistics:

$$\lambda = \frac{\exp(-D)}{C}. \tag{2}$$

- *False alarm rate per hour*:

$$\lambda_{\mathrm{FAR}} = \frac{3600 \cdot \lambda A}{\Delta t}. \tag{3}$$

- *Probability of correct association*: we assume that the statistical nearest neighbor is used for track update, that the target-originated contact is one standard deviation in each dimension from the true target location, and that the track has steady-state filter covariance based on consecutive detection events. Thus, letting S be the *innovation covariance* [4, p. 49] and letting V be the *validation region volume* [4, p. 96], we have:

$$P_{\mathrm{CA}} = \exp(-\lambda V) \tag{4}$$

$$V = \pi |S|^{1/2} \tag{5}$$

$$S = P(-) + R \tag{6}$$

$$P(-) = P(+) + Q \tag{7}$$

$$P(+) = P(-) - P(-)(P(-) + R)^{-1} P(-) \tag{8}$$

$$Q = \begin{bmatrix} q\Delta t & 0 \\ 0 & q\Delta t \end{bmatrix}. \tag{9}$$

Note that $P(-)$ denotes the filter prediction covariance, while $P(+)$ denotes the filter update covariance. Further, note that $P(-)$ is the solution to the (steady-state) *algebraic Riccati equation* (ARE) [17].

- *Probability of track update and miss* (i.e., track *coast*), where a track update requires that the current scan include target detection, successful gating to the target track, and correct association:

$$P_{\mathrm{U}} = P_{\mathrm{D}} P_{\mathrm{G}} P_{\mathrm{CA}} \tag{10}$$

$$P_{\mathrm{M}} = 1 - P_{\mathrm{U}}. \tag{11}$$

- *Average track confirmation time* (note that the expected value of the geometric distribution with parameter p is given by $1/p$), where track confirmation requires N consecutive, associated target detections:

$$\tau_C = \frac{1}{P_{\mathrm{U}}^{N-1}} \left(N - 1 + \frac{1}{P_{\mathrm{D}}}\right) \Delta t. \tag{12}$$

- *Average track hold time*, where track termination is achieved after K consecutive scans without an associated target detection:

$$\tau_H = \frac{1}{P_{\mathrm{M}}^{K-1}} \left(K - 1 + \frac{1}{P_{\mathrm{M}}} \right) \Delta t. \tag{13}$$

Note that equations (12–13) both rely on nested geometric probability distributions—that is to say, the sojourn time prior to tentative track initiation has a geometric distribution, as does the total track initiation time. It is easy to show (by linearity of the expectation operator) that equations (12–13) hold.

- *Track detection probability*, given by the fraction of time during which a target has a corresponding confirmed track:

$$P_{\mathrm{D}}^{T} = \frac{\tau_H}{\tau_C + \tau_H}. \tag{14}$$

- *Probability of false update*, given by one minus the probability that no false contacts exist in the association gate and again assuming steady-state filter covariance:

$$P_{\mathrm{FU}} = 1 - \exp(-\lambda V_\gamma) \tag{15}$$

$$V_\gamma = \gamma \pi |S|^{1/2}. \tag{16}$$

- *Probability of false track*, given by the probability that a false contact leads to a sequence of associated false contacts:

$$P_{\mathrm{FT}} = P_{\mathrm{FU}}^{N-1}. \tag{17}$$

- *False track rate per hour:*

$$\lambda_{\mathrm{FTR}} = \frac{3600 \cdot P_{\mathrm{FT}} \lambda A}{\Delta t}. \tag{18}$$

For simplicity and to minimize the number of modeling parameters, we have assumed track confirmation on N consecutive detections rather than a more general M-of-N track initiation criterion. The equations above generalize easily to the M-of-N criterion, using the binomial distribution with parameters $N-1$ and P_{U} for in the target-present case, and parameters $N-1$ and P_{FU} in the target-absent case.

We have invoked several modeling simplifications, including the impact of false updates on the true track formation and maintenance. This effect is estimated empirically in [8, pp. 207–208], as the impact is difficult to capture analytically. Here, we assume for simplicity that the impact of a false update in terms of track degradation is comparable to that of a track miss.

An illustration of the Markov chain model that corresponds to the modeling above is given in Fig. 1.

The tracker performance model introduced here shares some commonalities with the *system operating characteristics* (SOC) curve introduced in [5]. One of the differences is that the metrics of interest differ. In [5], a single, fixed time window in considered, and track detection and false track probabilities are computed. Rather, here the track detection probability is a measure of *track hold*, which answers the following question: "For a given target, what is the probability that there is a corresponding track at any given time?" The false track rate identifies the number of false objects generated by the tracker per unit time.

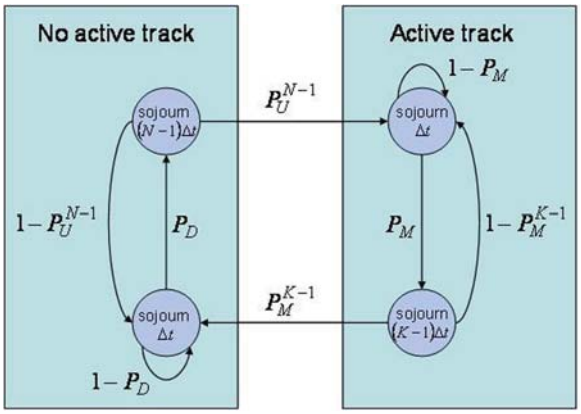


Fig. 1. Markov chain model for tracker logical state, in the target-present case. (A similar Markov chain applies to the target-absent case.)

TABLE I
Model Simulation Parameters

Parameter	Setting
Maneuverability index q	100 $\mathrm{m^2 s^{-1}}$
Target SNR d	10 dB
Scan interval $\Delta t'$	60 sec
Measurement covariance matrix R	$\begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$ $\mathrm{m^2}$
Surveillance region A	10^8 $\mathrm{m^2}$
Detection cell C	100 $\mathrm{m^2}$
Detection threshold D	5.0–9.0 dB
Track initiation N	3
Track termination K	3
Association gate γ	9.2
Gate probability P_{G}	0.99

2.2. Tracker Performance Analysis

We are interested to examine input and output performance curves (λ_{FAR} vs. P_{D}, and λ_{FTR} vs. P_{D}^{T}, respectively) as a function of the detection threshold D, and as a function of the number of sensors. The latter can be addressed by setting the scan rate to $\Delta t = \Delta t'/Z$, where Z is the number of equally-performing sensors and $\Delta t'$ is the single-sensor rate. Parameters are set as indicated in Table I, and performance curves are in Fig. 2. (Note that by *object* we mean either contact or track.)

Key conclusions are as follows:

- Tracking provides a roughly two-order-of-magnitude reduction in false objects, with comparable object detection performance.
- With a low constraint on false object rate, it is best to use few sensors.

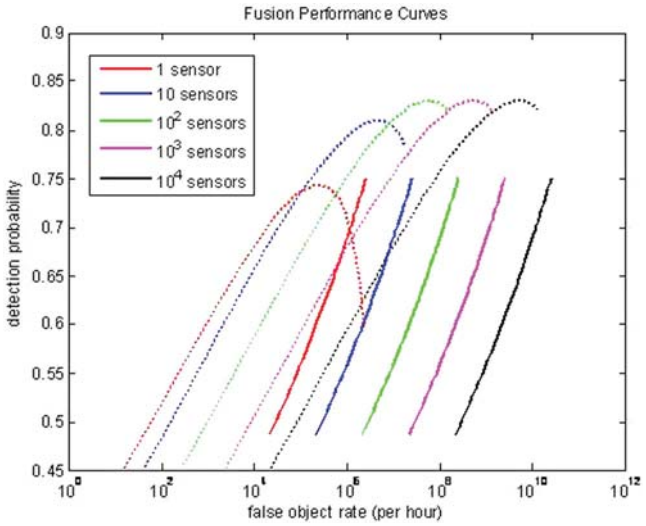


Fig. 2. Performance curves for several network-size assumptions. Solid lines characterize sensor performance, while dotted lines characterize tracker performance as characterized by the analytical model in Section 2.1.

- With a larger constraint on false object rate, it is best to use more sensors.
- For any given number of sensors, unlike the behavior of the (monotonic) input ROC curve, a maximum in track-level detection is achieved for a non-zero SNR detection threshold.

2.3. Optimal Detection Threshold and Number of Sensors

An approach to improve centralized tracking performance is to optimize the local sensor detection threshold (D) as well as the number of sensors to be processed (Z), as a function of a constraint on the false track rate. We anticipate that this will lead to a performance curve that is the envelope of the family of curves shown in Fig. 2.

For a given λ_{FTR}, we wish to optimize the local sensor detection threshold (D) as well as the sensor scan rate Δt (from which we infer the number of sensors). This optimization problem can be recast as the following *constrained* maximization problem:

$$\begin{aligned} &\max_{\Delta t, DT} \quad P_{\text{D}}^{T}(\Delta t, D) \\ &\text{s.t} \quad \lambda_{\text{FTR}}(\Delta t, D) = \alpha. \end{aligned} \tag{19}$$

Note that the dependence of P_{D}^{T} on Δt is complex, since P_{U} and P_{M} both depend on Δt. This optimization problem does not lend itself to an analytical solution. Using the same parameter settings as in Table I, the solution to equation (19) leads to the envelope of the family of curves in Fig. 2, as illustrated in Fig. 3. (Optimization is performed using the function `fmincon` in MATLAB.)

It is instructive to examine the optimal scan interval Δt^{opt} and the optimal detection threshold D^{opt} as a function of λ_{FTR}; these are illustrated in Figs. 4–5. It is interesting to note that, with increasing λ_{FTR}, the optimal P_{D}^{T} is achieved with a reduction in both Δt and D: we both increase the number of sensors and lower the detection threshold.

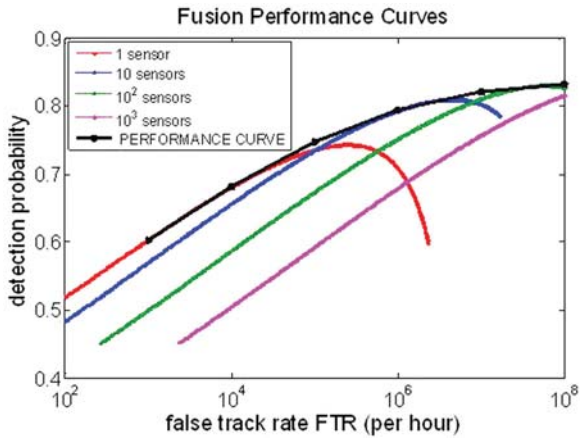


Fig. 3. Performance curve obtained solving the optimization problem (22).

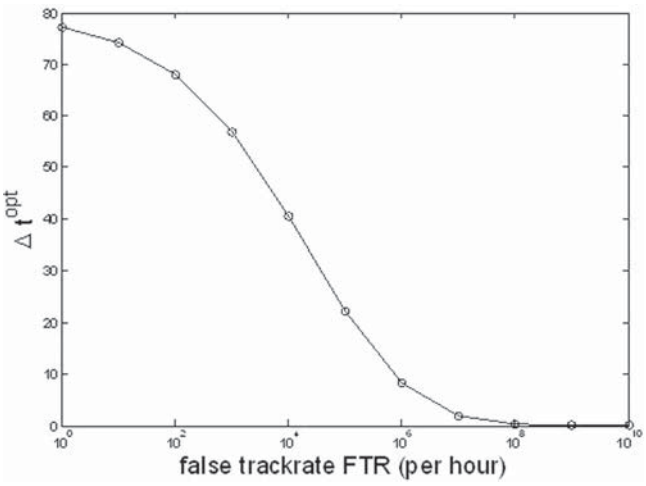


Fig. 4. Optimal scan interval (inversely proportional to number of sensors) as a function of λ_{FTR}.

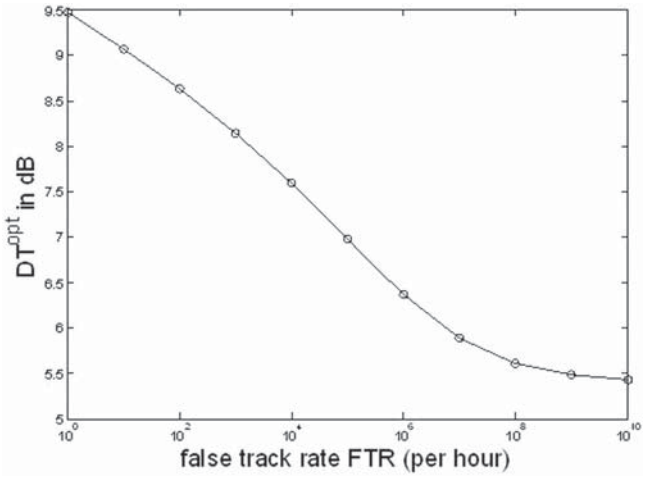


Fig. 5. Optimal detection threshold as a function of λ_{FTR}.

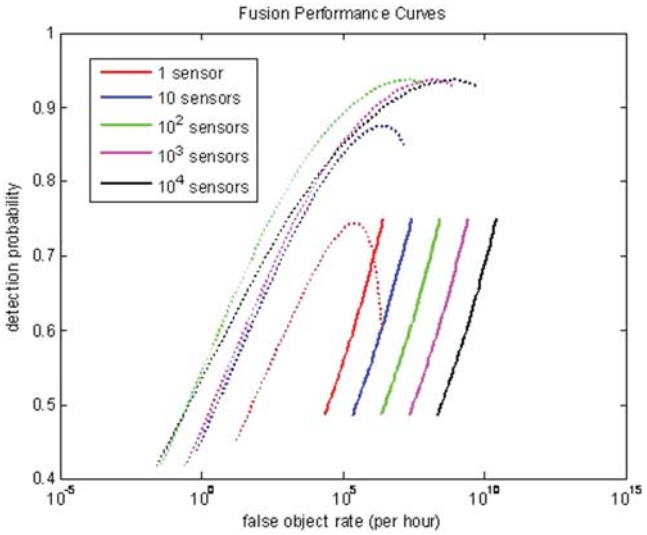


Fig. 6. Performance curves for several network size assumptions with adaptive track-management parameters. Solid lines characterize sensor performance, while dotted lines characterize tracker performance as characterized by the analytical model in Section 2.1.

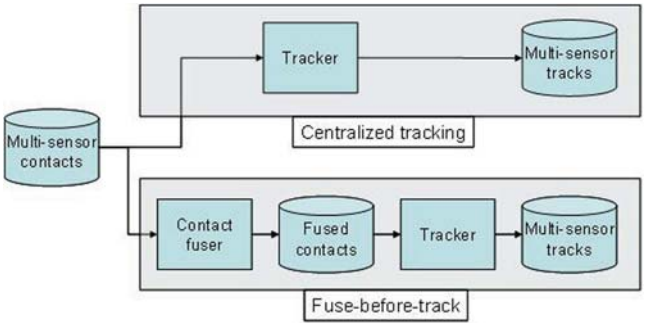


Fig. 7. Candidate fusion and tracking architectures.

TABLE II
Adaptive Track Management for the Results in Fig. 2

Number of Sensors (Z)	Track Initiation (N)
1	3
10	4
100	5
1,000	5
10,000	6

Note that our analysis has been limited to the assumptions of equally-performing sensors with identical detection thresholds. Relaxing either of these assumptions introduces the need for a more complex tracker performance model.

2.4. Tuning the Track Management Parameters

A possible objection to the results in Sections 2.2–2.3 is that we use the same track initiation and termination criteria throughout. More generally, one might wish to adapt the parameters N and K to the data rate and to the detection threshold. A fully adaptive selection of these parameters is quite complex. We may, however, seek to vary N and K as a function of the data rate only. In particular, neglecting the dependence of P_{FU} on the data rate, we can achieve a comparable false track rate by setting $N(Z)$ in the case of $Z > 1$ sensors as follows, where $N(1) = N_0$. We set the false track rate to be independent of the number of sensors:

$$\lambda_{\mathrm{FTR}} = \frac{3600 \cdot P_{\mathrm{FU}}^{N(Z)-1} \lambda A}{\Delta t / Z} = \frac{3600 \cdot P_{\mathrm{FU}}^{N_0-1} \lambda A}{\Delta t}.$$

Neglecting the dependence of P_{FU} on the scan rate leads to the following simple relationship between the track confirmation window length and the number of sensors:

$$N(Z) = N_0 + \frac{\log Z}{\log(1/P_{\mathrm{FU}})}. \tag{20}$$

We scale the parameter K in a comparable manner. As a result of adaptively-selected track-management parameters, the curves in Fig. 2 are modified to those shown in Fig. 6. The corresponding values of N (and K) are given in Table II.

We find that adaptive track management does have an impact on centralized tracking performance. In particular, note that the 100-sensor performance curve is better than the 10-sensor curve, which in turn outperforms the one-sensor curve. Nonetheless, the qualitative findings noted above still hold: significant reduction in false objects with automatic tracking, non-monotonicity in tracker performance curves, and saturation in performance benefits for a large-enough number of sensors. Correspondingly, the optimal choice for the number of sensors still depends on the false track requirement.

3. THE FUSE-BEFORE-TRACK ARCHITECTURE

Is it possible to exceed centralized tracking performance? At a conceptual level, the answer would seem to be no: every algorithmic step that is possible in multi-stage or distributed processing can be achieved in a centralized processing configuration.

A practical question of interest is whether a particular (sub-optimal) tracking module is better employed in a single-stage processing architecture or in a two-stage architecture. For the latter approach, we are interested to explore a *fuse-before-track* (FbT) processing scheme, whereby contact fusion across sensors precedes tracking over time. The two approaches are illustrated in Fig. 7.

It is important to note that the FbT architecture supports real-time processing just as the centralized architecture does: as fused contacts are produced, they provide input to the second-stage scan-based tracker.

For the purposes of this discussion, the particular choice of tracking module (here, a track-oriented multi-

hypothesis tracker [12–13]) is not critical; however, what *is* critical is that our tracker is representative of a real-time, scan-based algorithm that necessarily discards data-association hypotheses (in the case of non-Bayesian tracking) or that combines hypotheses (in the case of Bayesian tracking). Indeed, while amenable to real-time surveillance requirements, scan-based tracking approaches lack the performance potential of batch processing schemes such as [3].

For simplicity, we assume that the (active) sensors are (nearly) synchronized; that is, we assume that the scans of contact-level data are acquired for the same sequence of times, for all sensors. An alternative time-series representation of the two architectures in Fig. 7 is illustrated in Fig. 8.

The motivation for investigating the FbT architecture is as follows. The static fusion stage is not hindered by the requirement for scan-based processing, since all the sensors scan the surveillance region simultaneously. Thus, for large sensor networks, the two-stage architecture leverages the strength of batch processing in the fusion stage, while maintaining the real-time surveillance requirement with scan-based tracking.

Let us return for a moment to the argument that the same processing results obtained with FbT are in principle achievable with single-stage or centralized processing. After all, as we saw in the model-based results documented in Section 2, there is an advantage to appropriately scaling the track-management parameters with the sensor data rate. However, key data association parameters *do not* scale well with increasing data rate. For instance, for computational reasons, the *n-scan* track hypothesis depth parameter that is common in multi-hypothesis tracking (MHT) approaches cannot be scaled with the data rate. A similar consideration holds for multiple-model filters. Thus, as we will see in Section 5, an adjustment to track-management parameters for centralized processing is insufficient to match the promising performance results exhibited by the FbT architecture.

4. STATIC FUSION AND THE CONTACT SIFTING APPROACH

We focus now on the static fusion problem, which represents the first stage in the *fuse-before-track* (FbT) architecture. We will introduce the concept of *probability of localization* and use this concept in studying the (simple) contact sifting approach and its performance characteristics; subsequently, we will briefly discuss an alternative approach to static fusion.

4.1. Probability of Localization

Assume we have a surveillance region A composed of N_{cell} detection cells of equal size. Assume further that detection statistics in both the target-present and target-absent cases follow Rayleigh statistics, and that the expected target SNRs are given by d. Given a detection threshold D, it can be shown that targets have the following detection probability (same as equation (1)):

$$P_{\text{D}} = \exp\left(-\frac{D}{1+d}\right). \tag{21}$$

Further, the probability of false contact in any cell is given by:

$$P_{\text{FA}} = \exp(-D). \tag{22}$$

Accordingly, the number of false contacts is Poisson with parameter $\lambda_A = P_{\text{FA}} N_{\text{cell}}$. Further, the false contacts are uniformly distributed in the surveillance region.

The well-known classical ROC curve in this case is given by varying D over a range of values, with the following relationship between P_{D} and P_{FA}:

$$P_{\text{D}} = (P_{\text{FA}})^{1/(1+d)}. \tag{23}$$

A slightly modified ROC curve provides P_{D} as a function of the expected number of false contacts (same as λ_A):

$$N_{\text{FA}} = P_{\text{FA}} N_{\text{cell}}. \tag{24}$$

From an operational perspective, we are interested in detecting and localizing targets: target detections that are distant from the true target location are indistinguishable from false contacts; similarly, false contacts that fortuitously are close to the true target location are indistinguishable from target detections. Thus, we introduce the notion of *probability of localization*: the probability that a contact exists close enough to the target. This notion couples detection and localization metrics into a single quantity of interest; related work on the coupling between detection and localization objectives is found in [21].

Assume that our sensor provides two-dimensional Cartesian positional measurements of target location, with uncorrelated and identically distributed Gaussian errors in x and y, and variance σ_x^2. Let ε be the maximum distance for acceptable target localization. This defines a circular validation region around the target of area $B = \pi\varepsilon^2$. False contacts are uniformly distributed in this region with parameter $\lambda_B = \lambda_A B/A$. Define $\xi = \varepsilon/\sigma_x$.

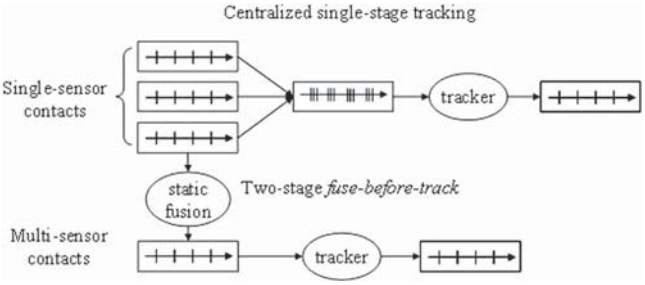


Fig. 8. FbT includes static fusion and scan-based tracking.

The probability of localization P_L is the following:

$$P_L = P_D\left(1-\exp\left(-\frac{\xi^2}{2}\right)\right) + (1-\exp(-\lambda_B)) - P_D\left(1-\exp\left(-\frac{\xi^2}{2}\right)\right)(1-\exp(-\lambda_B)). \quad (25)$$

The first term is the probability that a target is detected, with the detection lying in the validation region. The second term is the probability that at least one false contact is in the validation region; it can be derived by denoting $C = (1 - B/A)$ and observing that the probability that no false contact is in the validation region is given by:

$$\sum_{k=0}^{\infty}\frac{\lambda_A^k}{k!}\exp(-\lambda_A)C^k = \exp(-\lambda_A(1-C)) \times \sum_{k=0}^{\infty}\frac{(\lambda_A C)^k}{k!}\exp(-\lambda_A C) = \exp(-\lambda_A B/A) = \exp(-\lambda_B).$$

The third term in equation (25) is the probability that both target and non-target contacts are present in the validation region.

Contact sifting performance is defined analytically through equations (21–25). Note that these equations are based on some simplifying assumptions:

- Targets are not closely spaced: the model does not account for the presence of contacts from one target in the validation region of another.
- Targets are not near the edge of the surveillance region: we consistently use B as the size of the validation region.
- The impact of imperfect sensor resolution (grid cell size) is neglected: false contacts are assumed to be uniformly distributed in the surveillance area, and target detections are Gaussian distributed and centered on true target location.
- False contact statistics are based on the contact-absent case. That is, we do not include target contacts outside the validation region in the model for the number of false contact: the impact is minimal for non-trivial false contact rates.

Note that in the limit $\varepsilon \to 0$ and $\xi \to 0$ (i.e., $\sigma_x \to 0$ faster than $\varepsilon \to 0$), we see from equation (25) that $P_L \to P_D$. (One can think of P_L as a generalization of P_D in the presence of localization error.)

It is important to use the *probability of localization* as defined here (i.e., detection *and* localization) as a performance measure, rather than simply the *probability of detection*. Indeed, we are interested in comparing the performance at the input and output of the static fusion process. Accordingly, it is important to measure performance consistently: at the output of static fusion processing, we define target-induced and false contacts based on a localization threshold.

4.2. Contact Sifting Performance

Assume we have N independent identical sensors with performance as described in the previous section. Note that N in this section should not be confused with the track-initiation window size in Section 2: the two may be different in general.

Indeed, note that in Section 2 we reasoned over both Z sensors and the temporal window of N scans for track initiation. Here, we study the static fusion problem, where we can alternatively think of having a set of synchronous sensors, or a set of scans from the *same* sensor. That is, we only have Z or N to consider. We have chosen to denote this buffer size by N, as this buffer size relates directly to the previous track-initiation discussion.

Surveillance performance will depend critically on the specific fusion algorithm that we employ to combine N sets of contacts into a single set. In this section, we consider the simple batch-processing approach.

Even with the assumption of like-performing sensors, one might ask whether optimal multi-sensor performance will be achieved by requiring that all sensors use the same detection threshold D. It is known from the distributed detection literature that the assumption of equal local thresholds may lead to sub-optimal performance; nonetheless, in certain cases optimality is indeed achieved with identical local thresholds [25]. Further, under this assumption the form of the optimal fusion rule is known to be of the form K-of-N, though the optimal choice of K will depend on the local threshold, i.e., a fixed K-of-N fusion rule is not optimal in general. ROC performance of the K-of-N fusion rule is the following [22]:

$$P_D(K,N) = \sum_{j=K}^{N}\binom{N}{j}P_D^j(1-P_D)^{N-j} \quad (26)$$

$$P_{FA}(K,N) = \sum_{j=K}^{N}\binom{N}{j}P_{FA}^j(1-P_{FA})^{N-j}. \quad (27)$$

Note that K in this section should not be confused with the track-termination threshold defined in Section 2.

As in the single-sensor case, we are interested in a slightly modified definition of the classical ROC curve, where we replace $P_{FA}(K,N)$ with $N_{FA}(K,N)$; these are related as follows:

$$N_{FA}(K,N) = P_{FA}(K,N)N_{cell}. \quad (28)$$

For simplicity, we proceed with an assumption of equal local sensor statistics and local detection thresholds. The *contact sifting* approach relies on a sifting grid in measurement space. In each sifting cell, we sum the number of contacts over all N sensors. Sifting cells in which the number of contacts exceeds K lead to a fused contact, with localization based on appropriate averaging over the location of the single-sensor contacts (accounting for contact uncertainties).

In order to develop a simple model for contact sifting performance, we introduce a number of simplifying approximations, in addition to those previously introduced. The impact of these simplifying approximations will be evaluated experimentally.

- Assume targets are located at the center of sifting cells.
- Assume sifting cell size and shape is same as the validation region B introduced previously.
- Neglect overlaps among circular grid cells, and assume the cells fully cover the surveillance region. This simplifies the computation of false contact statistics, and ensures that successful localization corresponds precisely to fused contacts in target-present cells.

Let the random variables $N_{\mathrm{T}}^{\mathrm{cell}}$ and $N_{\mathrm{FA}}^{\mathrm{cell}}$ denote the number of target and non-target contacts in a particular sifting cell, respectively. Then, the probability of localization $P_{\mathrm{L}}(K,N)$ can be expressed by leveraging the following decomposition:

$$\begin{aligned} P_{\mathrm{L}}(K,N) &= P(N_{\mathrm{T}}^{\mathrm{cell}} + N_{\mathrm{FA}}^{\mathrm{cell}} \geq K) \\ &= P(N_{\mathrm{T}}^{\mathrm{cell}} \geq K) + \sum_{j=0}^{K-1} P(N_{\mathrm{T}}^{\mathrm{cell}} = j) P(N_{\mathrm{FA}}^{\mathrm{cell}} \geq K - j). \end{aligned}$$

Accordingly, letting $\tilde{P}_{\mathrm{D}} = P_{\mathrm{D}}(1 - \exp(-\xi^2/2))$, we have

$$\begin{aligned} P_{\mathrm{L}}(K,N) = &\sum_{j=K}^{N} \binom{N}{j} \tilde{P}_{\mathrm{D}}^{j} (1 - \tilde{P}_{\mathrm{D}})^{N-j} \\ &+ \sum_{j=0}^{K-1} \left[\binom{N}{j} \tilde{P}_{\mathrm{D}}^{j} (1 - \tilde{P}_{\mathrm{D}})^{N-j} \sum_{i=K-j}^{\infty} \frac{\lambda_B^i}{i!} \exp(-\lambda_B) \right]. \end{aligned} \tag{29}$$

The expected number of false contacts $\tilde{N}_{\mathrm{FA}}(K,N)$ is given by the following, where $N_{\mathrm{cell}}^{\mathrm{sift}}$ is the number of sifting cells that is not to be confused with the number of sensor grid cells N_{cell}:

$$\tilde{N}_{\mathrm{FA}}(K,N) = N_{\mathrm{cell}}^{\mathrm{sift}} \sum_{j=K}^{\infty} \frac{(N\lambda_B)^j}{j!} \exp(-N\lambda_B). \tag{30}$$

More compact analytical expressions for (29–30) based on the incomplete gamma function [1] are given as follows:

$$\begin{aligned} P_{\mathrm{L}}(K,N) = &\sum_{j=K}^{N} \binom{N}{j} \tilde{P}_{\mathrm{D}}^{j} (1 - \tilde{P}_{\mathrm{D}})^{N-j} \\ &+ \sum_{j=0}^{K-1} \left[\binom{N}{j} \tilde{P}_{\mathrm{D}}^{j} (1 - \tilde{P}_{\mathrm{D}})^{N-j} \Gamma(K-j, \lambda_B) \right] \end{aligned} \tag{31}$$

$$\tilde{N}_{\mathrm{FA}}(K,N) = N_{\mathrm{cell}}^{\mathrm{sift}} \Gamma(K, N\lambda_B). \tag{32}$$

Further, when N is large enough and λ_B small, $P_{\mathrm{L}}(K,N)$ given by equation (29) can be calculated by using the Laplace-De Moivre approximation [20]:

$$P_{\mathrm{L}}(K,N) = Q\left(\frac{K - N\tilde{P}_{\mathrm{D}}}{\sqrt{N\tilde{P}_{\mathrm{D}}(1 - \tilde{P}_{\mathrm{D}})}} \right). \tag{33}$$

$Q(\cdot)$ is the complementary distribution function of the standard normal random variable:

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt. \tag{34}$$

It is of interest to compare ROC curves based on equations (28–30) with performance curves based on equations (31–32). Letting $N_{\mathrm{cell}}^{\mathrm{sift}} = N_{\mathrm{cell}}$, we have $P_{\mathrm{FA}} = \lambda_B$. For large N, we have exact equivalence of equations (28) and (30); we have reasonable agreement for modest values of N. Next, we examine the limit $\varepsilon \to 0$ and $\xi \to 0$ (i.e., $\sigma_x \to 0$ faster than $\varepsilon \to 0$) in equation (29). Note that this limit impacts both the validation region and the sifting cell size, as we have fixed these to be the same. We have both $\tilde{P}_{\mathrm{D}} \to P_{\mathrm{D}}$ and $\lambda_B \to 0$. It follows immediately that $P_{\mathrm{L}}(K,N) \to P_{\mathrm{D}}(K,N)$.

Thus, contact sifting with $N_{\mathrm{cell}}^{\mathrm{sift}} = N_{\mathrm{cell}}$, extremely small localization errors ($\sigma_x \to 0$), and extremely large number of sensors ($N \to \infty$) corresponds precisely to the distributed-detection problem characterized by equations (26–28). Even for finite N, there is close agreement with the analytical performance curve given by equations (29–30).

The choice of contact sifting cell size has inherent tradeoffs. Large cells will reduce the contrast between target-absent and target-present statistics. Likewise, a small cell containing a target is less likely to contain those target-originated contacts that incur significant localization errors. In addition to cell size, the choice of threshold parameter includes non-trivial tradeoffs. As in distributed detection theory (and as noted earlier), in general the optimal contact-sifting fusion rule will dependent on the single-sensor (local) detection threshold.

Experimental validation of contact sifting performance (equations 29–30) is documented in [14].

4.3. Other Approaches to Contact Fusion

The *contact sifting* algorithm is not effective in close-target scenarios. Another approach to the problem is

to consider a *multi-sensor probabilistic data association* algorithm. Further details of this approach may be found in [16].

Under this approach, we take a *generalized likelihood ratio test* (GLRT) approach for both the detection and estimation problems. For each hypothesized target, we find its location estimate that maximizes the likelihood function, and choose the hypothesis that has the largest likelihood. This results in a procedure that maximizes the likelihood function with respect to the number of targets and their respective locations.

A constraint is imposed on the maximum allowable number of targets present in the surveillance region. A sequential search over the number of targets is used for a computationally feasible solution. The technique provides location estimates as part of the detection process. The location estimates can always be further refined by an estimation process. This approach in a different context is developed in [6–7]. In [7], comparisons are made between the proposed method and the unstructured and structured techniques based on Akaike information theoretic criteria (AIC) [2], minimum description length (MDL) [23], and Bayesian predictive density [11].

A known limitation of the GLRT approach is the *runaway* degree-of-freedom phenomenon. In [16], by not considering a penalty factor as prescribed by the AIC or *Bayesian Information Criterion* (BIC) approaches, we did face the problem of overestimating the number of targets.

Compared to the more complex approaches to contact fusion, the contact-sifting algorithm is simple and handles situations where we have large number of targets, albeit not closely spaced.

5. FUSE-BEFORE-TRACK PERFORMANCE STUDY

We have argued in Section 3 that the *fuse-before-track* architecture holds potential for target tracking in large sensor networks. Here, we provide results of Monte Carlo experimental validation. For both the centralized and FbT architectures, we use a track-oriented multi-hypothesis tracker [12–13]. Simulation and algorithmic parameters are in Table III.

We have stochastically-generated target ground truth based on a nearly constant velocity motion model, for which positional measurements are obtained from a number of like-performing sensors that are synchronized in time and with a fixed sensor revisit time. All target trajectories are initiated at scenario initiation and target death results if a target exits the scenario area. Initial target location is uniformly distributed in the surveillance region. The tracker is assumed to have knowledge of target motion and sensor parameters; by this we mean that the statistical characteristics of target motion and of the sensor measurement error are known.

Performance evaluation relies on track classification, whereby those tracks with sufficiently large average localization error from all target trajectories are classified as false. Otherwise, the closest trajectory is identified. For tracks that extend in time beyond a given target death, the last target location is used for positional comparison.

TABLE III
Parameters for Single-Sensor Tracking, Multi-Sensor Tracking, and FbT Simulation-Based Performance Evaluation

Parameter	Setting
Monte Carlo realizations	500
Number of targets	10
Target SNR	13 dB
Target maneuverability index	0.01 m^2s^{-3}
Initial velocity std. dev.	1 ms^{-1}
Sensor threshold	10.5 dB
Contact measurement error std. dev. (in both x and y dimensions)	10 m
Number of sensors	10
Sensor revisit time	10 sec
Scenario duration	3 min
Surveillance region	$(1.5\ km)^2$
Detection cell size	$(1\ m)^2$
Sifting cell size	$(30\ m)^2$
Sifting threshold (number of contacts)	3
Track initiation (FbT)	4-of-4
Track initiation (centralized)	12-of-40
FbT track termination (allowed misses)	3
Centralized track termination (allowed misses)	20
Hypothesis tree depth (n-scan)	2
Track classification distance threshold	14.14 m

Note that the track-initiation setting for the multi-sensor (centralized) configuration is different than in FbT, so as to have a comparable track rate at the processing output. Indeed, a concatenation of M-of-N track initiation criteria is roughly comparable to a rule where the Ms are multiplied together, and likewise for the Ns. Similarly, it can be verified that a track that is kept alive with a concatenation of M-of-N rules (with $M = 1$ in the second stage) has maintenance statistics comparable to a single (centralized) 1-of-N with an appropriate choice of N (in our case, 20 allowed missed detections).

Similarly, track termination is based on the maximum time since the last track update, rather than on the number of missed updates.

The sensor threshold and target SNR settings above lead to a target probability of detection (P_D) of 0.62. The sensor threshold, detection cell size, and surveillance region sizes lead to a contact false alarm rate (λ_{FAR}) of 30 contacts per scan. Given the scenario revisit time and scenario duration, for each Monte Carlo realization there are 18 contact files for each sensor, and 180 in total, leading to 18 fused-contact files.

Given the sifting cell size and sifting thresholds above, the first stage of FbT processing generally leads to approximately 20 fused contacts per scan. The fused-contact location is given by the mean of the contacts in the sifting cell and, correspondingly, the fused-contact measurement covariance is smaller than that of single-sensor contacts. These statistics follow directly from

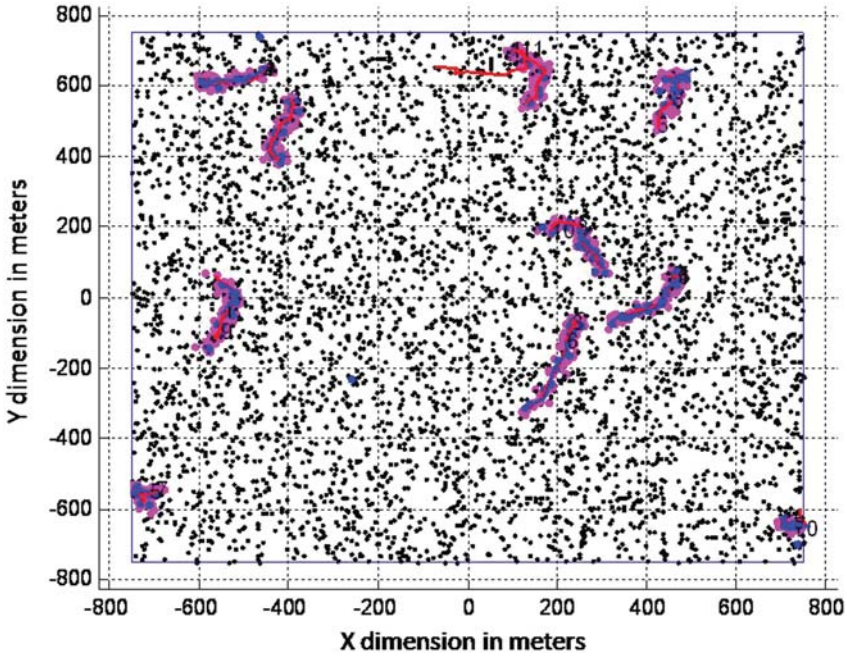


Fig. 9. One realization in the simulation-based analysis of centralized and fuse-before-track processing. False contacts are black dots, target-originated contacts are magenta dots, target trajectories are magenta, fused contacts in FbT processing are blue dots, FbT tracks are blue, and centralized tracks are red.

TABLE IV
Tracker Performance Results

Metric	Centralized	FbT
P_T	0.90	0.60
λ_{FTR}	186	28
Execution time ratio	0.78	0.013

least-squares estimation, and may alternatively be interpreted as the result of Kalman filtering of a sequence of contacts at the same measurement time.

Performance results are given in Table IV. Note first that the overall input false contact rate is roughly 108,000 contacts per hour. Thus, in both centralized and FbT configurations, target tracking provides a significant data reduction.

The key track detection statistics are *track hold* (P_T), the average fraction of time during which an active track exists on a target, and the *false track rate* (λ_{FTR}), the number of false tracks produced per unit time. The centralized tracking and FbT results are comparable: the former has better track hold, while the latter has a lower false track rate. Both achieve a dramatic reduction in false objects, for a track hold that is comparable or better than the target probability of detection.

The results documented here are encouraging, since the first stage of FbT processing (static fusion) could be improved further, as discussed in Section 4.3. Thus, the FbT architecture is promising and deserves further investigation.

Perhaps a more compelling motivation for the FbT architectures is the dramatic improvement in processing time. Indeed, contact sifting provides a dramatic reduction in the contact data rate: fused contact files include approximately 20 contacts. This, combined with the tenfold reduction in the number of contact files, leads to considerable computational savings in MHT processing. The savings in execution time are a combination of reduced computational load as well as saving in reading and writing a much small number of input and output files, respectively.

The *execution time ratio* in Table IV is the ratio of average tracker processing time and scenario duration. We see that centralized tracking achieves slightly faster than real-time processing, while FbT requires only a small fraction of the processing time. (Results are generated on a DELL OPTIPLEX GX620 with Intel Pentium D processor.)

Fig. 9 provides an illustration of a realization of contact-level and track-level data, while Figs. 10–12 illustrate some examples in detail. As indicated in Table III, generally we find that the FbT approach exhibits good track stability, at the cost of a longer track initiation time that in turn induces a lower track hold than in centralized tracking. Centralized tracking exhibits poorer tracking stability and, correspondingly, a higher false track rate.

The example in Fig. 12 shows two target trajectories that start in close proximity. While it takes longer to initiate tracks with the FbT approach, this is achieved without false track formation.

It should be noted that this study has been limited to random target tracks in a fairly wide surveillance region, which rarely leads to dense multi-target instantiations. These would challenge the FbT architecture as implemented here, as the first-stage contact-sifting

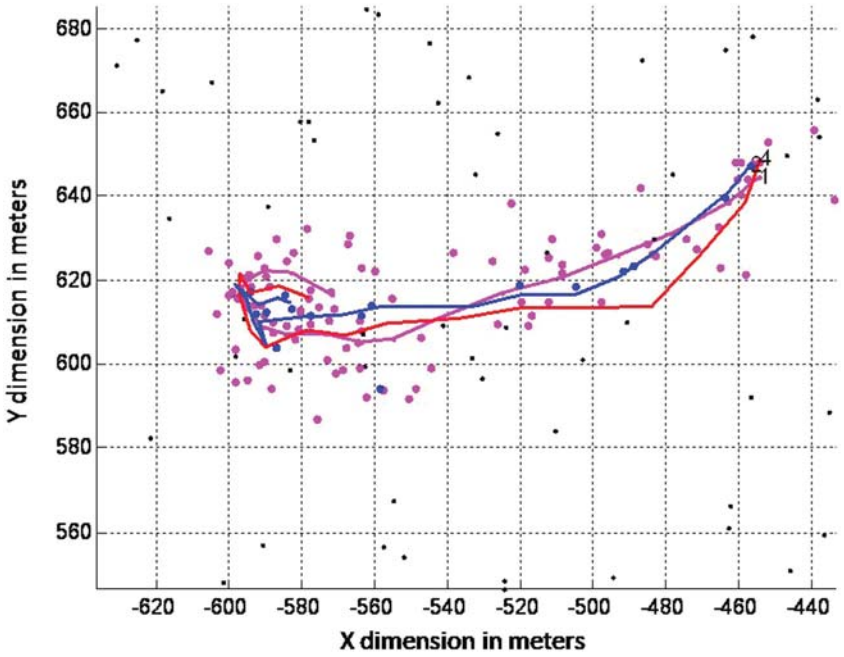


Fig. 10. An instance of comparable performance of centralized (red) and FbT (blue) tracks.

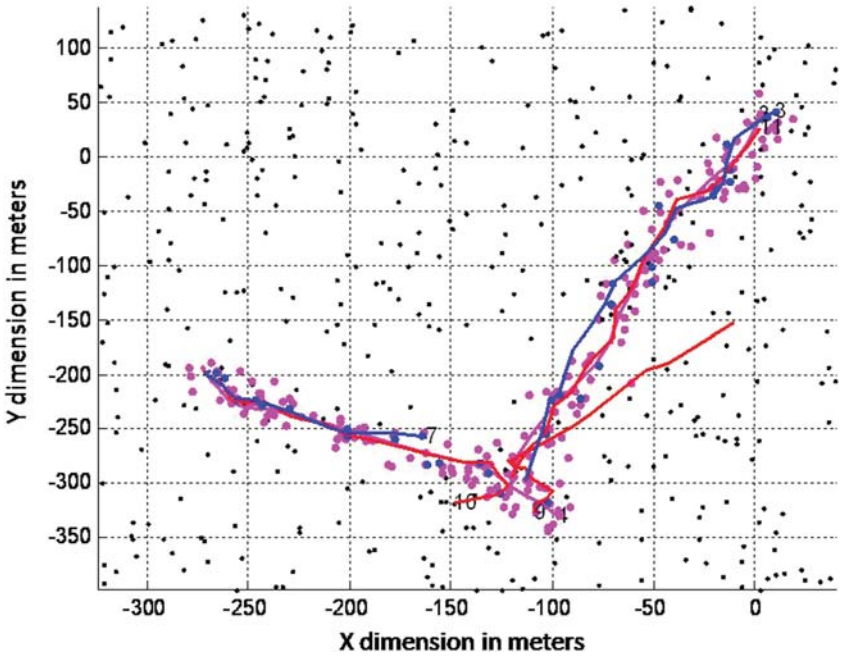


Fig. 11. An illustration of lower track hold but with more stable tracks in the FbT (blue) approach.

algorithm is known to be inappropriate for close-target cases. Likewise, in a computational sense, dense-target scenarios would challenge the centralized tracker more severely than it would in the FbT approach.

6. CONCLUSIONS

While large sensor networks hold great potential for surveillance performance, current scan-based target tracking technology by itself may not offer the best processing paradigm. Conversely, existing batch processing approaches do not provide real-time surveillance outputs. Thus, we believe a two-stage architecture that leverages the strengths of both batch and scan-based processing holds great potential for effective surveillance performance. In particular, contact fusion for a large number of nearly simultaneous sensor scans may be followed quite effectively by scan-based tracking.

This paper has addressed these contributions. First, in Section 2 we introduced an analytical performance model for scan-based tracking, and studied the performance limitations that the model suggests for increasing date rates (or number of sensors). Next, in Section 3 we introduced the *fuse-before-track* (FbT) architecture

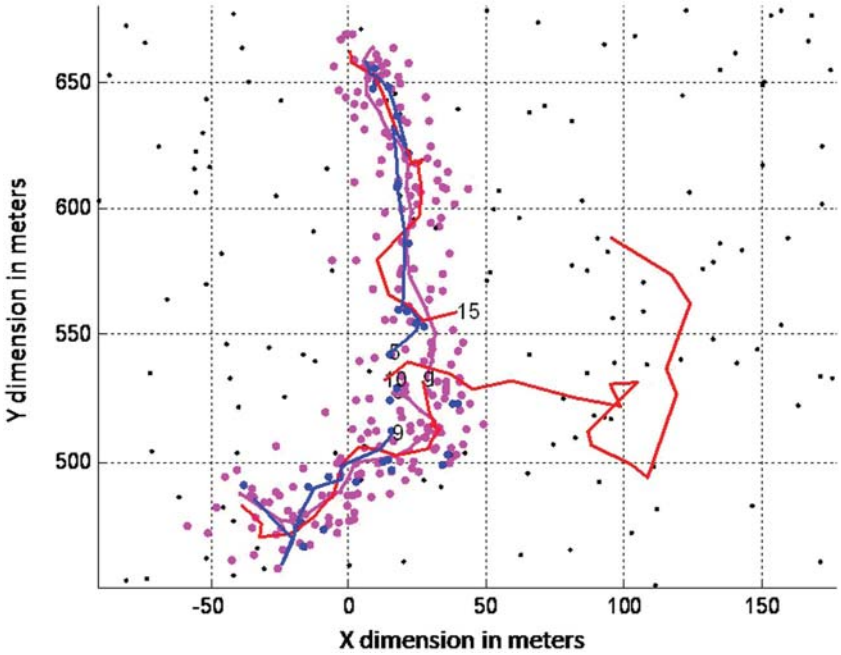


Fig. 12. Another illustration of reduced FbT (blue) track hold, but without false track confirmation as observed in the centralized case (red).

for automatic tracking in large sensor networks, which includes contact fusing followed by scan-based tracking; the specific FbT instantiation that we have implemented utilizes the *contact sifting* algorithm followed by an MHT tracker. After validating an analytical performance model for contact sifting and briefly describing another approach to contact fusion in Section 4, in Section 5 we describe simulation results that compare centralized and FbT processing results. We found that both approaches hold some merit and indeed both provide a dramatic reduction in false object rates. The FbB provides considerable computational savings, good track stability, and a lower false track rate, at the cost of reduced track hold.

A number of directions for future work exist. Principally, and in addition to a more effective first stage in the FbT architecture to handle closely-spaced target scenarios, the future direction includes an analysis of the impact of synchronized vs. staggered sensor sampling times [18, 26], for which analysis in the large-sensor case is lacking.

ACKNOWLEDGMENTS

Peter Willett was supported under ONR Contract N00014-07-1-0055. Marco Guerriero was partially supported through a six-month NURC visiting research appointment in 2007.

REFERENCES

[1] M. Abramowitz and I. Stegun
Handbook of Mathematical Functions.
Dover, New York, 1970.

[2] H. Akaike
A new look at the statistical model identification.
IEEE Transactions on Automatic Control, **19** (Dec. 1974).

[3] D. Avitzour
A maximum likelihood approach to data association.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 2 (Apr. 1992).

[4] Y. Bar-Shalom and X. Li
Multitarget-Multisensor Tracking.
YBS Publishing, 1995.

[5] Y. Bar-Shalom, L. Campo and P. Luh
From receiver operating characteristic to system operating characteristic: Evaluation of a track formation system.
IEEE Transactions on Automatic Control, **35**, 2 (Feb. 1990).

[6] R. Bethel and K. Bell
A multi-hypothesis GLRT approach to the combined source detection and direction of arrival estimation problem.
In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP '01), Salt Lake City, UT, May 2001.

[7] R. Bethel and K. Bell
Maximum likelihood approach to joint array detection/estimation.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 3 (July 2004).

[8] S. Blackman
Multi-Target Tracking with Radar Application.
Artech House, 1986.

[9] S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems.
Artech House, 1999.

[10] W. Blanding, P. Willett, Y. Bar-Shalom and S. Coraluppi
Multisensor track termination for targets with fluctuating SNR.
In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Honolulu, HI, Apr. 2007.

[11] C. Cho and P. Djuric
Detection and estimation of DOAs of signals via Bayesian predictive densities.
IEEE Transactions on Signal Processing, **42**, 11 (Nov. 1994).

[12] S. Coraluppi and C. Carthel
Distributed tracking in multistatic sonar.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 3 (July 2005).

[13] S. Coraluppi, C. Carthel, D. Grimmett and O. Gerard
Multistatic sonar tracking: Algorithmic advances and sea trial evaluation.
U.S. Navy Journal of Underwater Acoustics, **56**, 3 (July 2006).

[14] S. Coraluppi, M. Guerriero and P. Willett
Contact fusion in large sensor networks: Operational performance analysis.
In *Proceedings of the NATO RTO SET Panel Symposium on Sensors and Technology for Defence Against Terrorism*, Mannheim, Germany, Apr. 2008.

[15] S. Coraluppi, M. Guerriero and P. Willett
Optimal fusion performance modeling in sensor networks.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.

[16] M. Guerriero, S. Coraluppi and P. Willett
Analysis of scan and batch processing approaches to static fusion in sensor networks.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, Orlando FL, Mar. 2008.

[17] P. Kumar and P. Varaiya
Stochastic Systems.
Prentice-Hall, 1986.

[18] R. Niu, P. Varshney, K. Mehrotra and C. Mohan
Temporally staggered sensors in multi-sensor target tracking systems.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 3 (July 2005).

[19] M. Prior and A. Baldacci
The physical causes of clutter and its suppression via subband processing.
In *Proceedings of OCEANS 2006*, Boston, MA, 2006.

[20] A. Papoulis
Probability, Random Variables, and Stochastic Processes.
McGraw-Hill, New York, 1984.

[21] C. Rago, P. Willett and M. Alford
Predetection fusion: Resolution cell grid effects.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 3 (July 1999).

[22] A. Riebman and L. Nolte
Optimal detection and performance of distributed sensor systems.
IEEE Transactions on Aerospace and Electronic Systems, **23**, 1 (Jan. 1987).

[23] J. Rissanen
Modeling by shortest data description.
Automatica, **14** (1978).

[24] L. Stone, C. Barlow and T. Corwin
Bayesian Multiple Target Tracking.
Artech House, 1999.

[25] J. Tsitsiklis
Decentralized detection.
Advances in Signal Processing, JAI Press, 1993.

[26] X. Zhang, P. Willett and Y. Bar-Shalom
Uniform versus nonuniform sampling when tracking in clutter.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 2 (Apr. 2006).

Stefano Coraluppi (SM'XX) received B.S. degrees in electrical engineering and in mathematics from Carnegie Mellon University, Pittsburgh, PA, in 1990, and M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, in 1992 and 1997.

He has held research staff positions at ALPHATECH Inc., Burlington, MA (1997–2002) and at the NATO Undersea Research Centre in La Spezia, Italy (since 2002). He has contributed to numerous programs in ground, undersea, and maritime surveillance for security and defense applications. His research interests include detection and estimation theory, target tracking, data fusion, and stochastic control.

Dr. Coraluppi served as General Cochair of the 9th ISIF/IEEE International Conference on Information Fusion (FUSION) in Florence, Italy, in 2006 (with Peter Willett) and Program Chair of the NATO Workshop on Data Fusion and Anomaly Detection for Maritime Situational Awareness in La Spezia, Italy, in 2009. He has served as member of the NATO Research and Technology Organization (RTO) Sensors and Electronics Technology (SET) Panel, associate editor for both the *IEEE Journal of Oceanic Engineering* and the *ISIF Journal of Advances in Information Fusion* (JAIF), and lecturer for the NATO RTO, the Italian Naval Academy, and the University of Pisa. Currently, he is associate editor for Target Tracking and Multisensor Systems for the *IEEE Transactions on Aerospace and Electronic Systems*, area editor for Tracking for the ISIF JAIF, and member of the Board of Directors of the International Society of Information Fusion (serving as president in 2010).

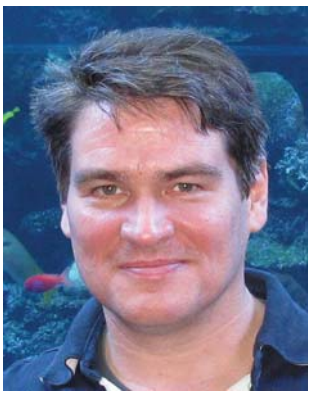
Marco Guerriero was born in Salerno, Italy, on June 18, 1981. He received his B.A.Sc. and M.Sc. (electrical engineering) from the University of Salerno, Italy, in 2002 and 2005, respectively and his Ph.D. degree from University of Connecticut in 2009.

In the fall of 2009 he was a visiting scientist at the NATO Undersea Research Centre (NURC) La Spezia, Italy. He is now a system analyst at the Research and Advanced System Design Department at ELETTRONICA S.p.A. in Rome. His research interests lie in the areas of signal processing, with particular focus on distributed detection and estimation in sensor networks, target tracking and data fusion and radar signal processing.

Peter Willett (F'03) received his B.A.Sc. (engineering science) from the University of Toronto in 1982, and his Ph.D. degree from Princeton University in 1986.

He has been a faculty member at the University of Connecticut since 1986, and since 1998 has been a professor. He has published 135 journal articles (13 more under review), 290 conference papers, and 9 book chapters. His primary areas of research have been statistical signal processing, detection, machine learning, data fusion and tracking. He has interests in and has published in the areas of change/abnormality detection, optical pattern recognition, communications and industrial/security condition monitoring.

He is editor-in-chief for *IEEE Transactions on Aerospace and Electronic Systems*, and until recently was 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*, parts A and B. He is also associate editor for the IEEE AES Magazine, editor of the AES Magazine's periodic Tutorial issues, associate editor for ISIF's electronic *Journal of Advances in Information Fusion*, and is a member of the editorial board of IEEE's Signal Processing Magazine. He was a member of the IEEE AESS Board of Governors 2003–2009. He was general cochair (with Stefano Coraluppi) for the 2006 ISIF/IEEE Fusion Conference in Florence, Italy, Program Co-Chair (with Eugene Santos) for the 2003 IEEE Conference on Systems, Man & Cybernetics in Washington, D.C., and program cochair (with Pramod Varshney) for the 1999 Fusion Conference in Sunnyvale. He was coorganizer of the tracking subsession at the 1999 IEEE Aerospace Conference, and has been organizer of the Remote Sensing Track of that conference 2000–2003. Jointly with T. Kirubarajan he has coorganized the SPIE "System Diagnosis and Prognosis: Security and Condition Monitoring Issues" Conference in Orlando, 2001–2003. He has been a member of the IEEE Signal Processing Society's Sensor-Array & Multichannel (SAM) Technical Committee since 1997, and both serves on that TC's SAM Conferences' Program Committees and maintains the SAM website.



Craig Carthel received B.S. degrees in physics and mathematics in 1988, a M.S. in mathematics in 1992, and a Ph.D. in mathematics in 1995, all from the University of Houston, where he did research in numerical analysis and optimization theory.

He is a senior scientist in the Applied Research Department at the NATO Undersea Research Centre in La Spezia, Italy, where he works on military operations research, simulation, optimization, and data fusion problems associated with maritime environments. From 1995 to 1997, he worked at the Institute for Industrial Mathematics at Johannes Kepler University, in Linz, Austria on parameter identification and inverse problems. From 1998 to 2002, he was a senior mathematician at ALPHATECH Inc. in Burlington, MA, where he worked on image processing, multisensor data fusion and ground target tracking. In 2006, he served as the technical program chair for The 9th International Conference on Information Fusion.