New Metrics for Quantifying Data Association Performance

Mark E. Silbert
NAVAIR
Patuxent River, MD 20670
mark.silbert@navy.mil

Craig S. Agate
Toyon Research Corp.
Goleta, CA
cagate@toyon.com

***Abstract*—Numerous metrics exist for quantifying the performance of information fusion systems. Some metrics focus on estimation accuracy by comparing estimated quantities to the truth. Other metrics assess the accuracy of the estimation uncertainty by determining the *consistency* of the estimation error covariance. In this paper we define two metrics that quantify the data association algorithm's performance (whether the data are measurements or tracks). We compare the metric to a few existing metrics that quantify the effects of data association and evaluate the new metrics both with some notional examples and with some simulated data run through a track-to-track (T2T) fusion algorithm. Finally, we discuss a direct analogy between the data association problem and the information retrieval problem and reference two metrics in the information retrieval domain that are equivalent to the two metrics proposed in this paper.**

***Index Terms*—fusion performance, association performance, missed association, incorrect association, track-to-track association, precision, recall**

I. Introduction

Multi-sensor tracking systems are prevalent throughout both the military and civilian domains. The objective of these systems is to process data collected by multiple sensor systems and generate *tracks* on the entities in the surveillance region. There are two key steps in the process of tracking: data association and state estimation. In data association, newly received data are associated with tracks in the tracker's database. After the assignment process is done, the tracks to which the data were assigned are updated, meaning that their state estimates are modified to account for the newly received information. This process applies to both measurement-to-track (M2T) fusion, in which case the data are sensor measurements to be assigned to local tracks, and track-to-track (T2T) fusion, in which case the data are local track reports to be assigned to composite tracks.

The association step is key since improving the track state estimates relies on correct association of data to tracks. There are metrics that attempt to quantify the performance of the data association step. Some of these derive from measures-of-performance (MOPs) defined for the target tracking problem [1]. Examples of these MOPs are track purity, track continuity, target purity and target continuity. We shall discuss these subsequently. Another set of metrics that originate from the track fusion problem also measure the association problem, but at a "higher" level in the sense that they do not quantify all association *decisions* made by the association algorithm; rather, they measure the resulting effects of association with respect to the number of tracks per entity (e.g., track redundancy). In the next section we review these existing metrics.

II. Existing Association Metrics

The plethora of metrics that exist for quantifying the performance of fusion systems are too numerous to discuss here, so we focus our discussion on metrics related to quantifying the performance of the *data association* problem in tracking. Note that there are some tracking approaches (e.g., PHD filter, Intensity filter) that do not explicitly maintain tracks and, therefore, do not explicitly associate measurements to tracks. Association metrics do not apply to these tracking approaches, but other metrics such as the Optimal Sub-Pattern Assignment (OSPA) [2] are more appropriate to evaluate the performance of such approaches.

Several metrics related to data association came out of research into MOPs for quantifying ground moving target indicator (GMTI) tracker performance [1]. The first two are target-focused:

- **Target Continuity** - measures the number of tracks initialized on a given target. This metric measures the association algorithm's failure to associate data to an existing track.
- **Target Purity** - measures the percent of a target's detections that went into its predominant track. This metric essentially measures what percent of a target's life was spent in its main track. This gives almost no information about the association decisions.

The other two MOPs are track-focused:

- **Track Continuity** - measures the number of individual targets associated with a given track. This only indirectly measures mis-associations.
- **Track Purity** - measures the percent of a track's associated data that derive. from the predominant target [3].

Of these four metrics, track purity gives the closest measure of a tracker's mis-association performance (what we shall call incorrect associations). For more discussion of purity and further references, see [4], [5]. Fewer incorrect associations result in higher purity. While this is a useful metric, it has two key problems. First, the metric is cumulative over the track's lifetime so it does not differentiate between incorrect associations made early its history with those made late. Second, the metric can easily be defeated by placing each measurement into its own track, yielding a purity of 100%.

The first problem can be addressed by maintaining a time series on the change in purity. Thus tracks that are impure early on will show decreasing purity early in the track life but then level off towards its end. Tracks that become more impure toward the end will show no change in the tracks' early life but will show a decrease in purity at the end of the track's life. The second problem can only be addressed by including a *countering* metric. A countering metric is one that has a converse relationship to another metric. For example, probability of detection (P_d) and probability of false alarm (P_{fa}) are countering metrics. A sensor should have high P_d and low P_{fa}. However, a sensor can be made more sensitive to detections (increased P_d) at the expense of introducing more false detections.

There are also metrics to quantify the association performance of a T2T fusion system. These metrics do not directly evaluate the individual association decisions made by the algorithm, but they quantify the resulting *effects* of association in terms of the number of tracks per tracked entity. They are defined as:

- **Conciseness** - is the percent of tracked entities that have a single composite track associated with them.
- **Multiplicity** - is the percent of tracked entities that have more than one composite track associated with them. This gives a sense of the algorithm's failure to associate data with an existing track.

Track conciseness can be thought of as the countering metric for track purity. The track conciseness metric is intended to measure how many extra tracks are created by the fusion system. In [6], the term clarity is used to convey the same idea as conciseness but is further broken down into ambiguous and spurious tracks. Redundancy [7] is yet another metric that quantifies the formation of extra tracks.

While purity, conciseness, multiplicity, redundacy, and other similar metrics are useful, they do not *directly* evaluate the individual association decisions made by the association algorithm; they evaluate how those decision affect the resulting tracks. What is needed are metrics that directly evaluate the association decisions made by the algorithm.

After developing our two metrics for evaluating measurement-to-track and track-to-track associaiton, we became aware of analogous metrics used in the domains of *information retrieval* and *machine learning*. These metrics are called *precision* and *recall* [8], [9] and are equivalent to the metrics that we have developed here. We shall defer a discussion of these two metrics (see Section III-E) until after we have presented the two metrics proposed in this paper.

III. Association Metrics

Before defining the association metrics that we propose to use for evaluating a T2T or M2T fusion system, it will be helpful to define some terms. First, we define an *association event* as a *decision* that two pieces of information should or should not be associated[1] An association event can be categorized as one of two types. A *positive association* is a decision to associate two pieces of information, while a *negative association* is a decision *not* to associate two pieces of information.

The above definitions characterize the decisions made by an association algorithm. In order to evaluate these decisions, the *truth entity* corresponding to that data (e.g., for a measurement, this would be the truth entity from which that measurement was derived). Establishing the truth entity of a piece of information is referred to as the "truth-to-track assignment problem," and various methods for determining the truth entity are given in [10].

With these definitions we are now ready to define the following joint events (over truth and algorithm decisions):

- **Correct Positive Association**: a positive association in which the truth entities of the associated elements are the same.
- **Incorrect Positive Association**: a positive association in which the truth entities of the associated elements are *not* the same.
- **Missed Association (or incorrect negative association)**: a negative association in which the truth entities of the elements are the same.
- **Correct Non-Association**: a negative association in which the truth entities of the two elements are *not* the same.

The above joint event characterizations can be represented in a two-dimensional table as shown in Table I. The columns of the table reflect the truth, and the rows represent the decisions made by the fusion system. Now that the association events are defined and characterize the association decisions, the associations made by the M2T/T2T algorithm can be quantified.

TABLE I
Association event characterizations.

	True Assoc.	**False Assoc.**
Positive Assoc.	Correct Assoc.	Incorrect Assoc.
Negative Assoc.	Missed Assoc.	Correct Non-Assoc.

A. Pairwise Association Events

Consider a process in which local tracks are received from two different sources (sources R and S) at various points in time, and there are two truth entities being tracked (entities a and b). The association algorithm associates received local tracks with previously received local tracks (and fuses them for improved estimates). Typically, the previously received and fused local tracks are represented by system or composite tracks, but it will be useful to explicitly designate the previously received tracks. For the development that follows we

[1] Most association algorithms make "hard" associations at some point. There are some "soft" association algorithms that do not make explicit associations, so this approach would not apply to these.

use $X_i^y(t)$ to denote the i^{th} track from Source X at time t with associated truth type y.

Suppose a local tracker (source R) is initially tracking the two entities with tracks R_1 and R_2. Additionally, there is a second local tracker (source S) tracking these two entities with tracks S_1 and S_2. Unbeknownst to the track fusion system, R_1 is updated at time t_4 with a false alarm (and is, therefore, not following entity a nor b). Local tracker R then forms a new local track on entity a. Shown in Table II is a history of received local tracks along with the associations made by the T2T association algorithm.

TABLE II
RECEIVED LOCAL TRACKS AT DIFFERENT TIMES AND FORMED COMPOSITE TRACKS.

Composite Track	$\mathbf{t_1}$	$\mathbf{t_2}$	$\mathbf{t_3}$	$\mathbf{t_4}$
C_1	$R_1^a(t_1)$	$R_1^a(t_1)$ $S_1^a(t_2)$	$R_1^a(t_1)$ $S_1^a(t_2)$ $R_1^a(t_3)$	$R_1^a(t_1)$ $S_1^a(t_2)$ $R_1^a(t_3)$ $R_1^?(t_4)$
C_2	$R_2^b(t_1)$	$R_2^b(t_1)$ $S_2^b(t_2)$	$R_2^b(t_1)$ $S_2^b(t_2)$	$R_2^b(t_1)$ $S_2^b(t_2)$ $R_2^b(t_4)$
C_3			$R_2^b(t_3)$	$R_2^b(t_3)$
C_4				$R_3^a(t_4)$

Consider the association problem at the fusion node at time t_3. The fusion node receives local updates for tracks $R_1(t_3)$ and $R_2(t_3)$ and must associate them with existing composite tracks $C_1(t_3)$ and $C_2(t_3)$ or create new composite track(s) as it does in this example (i.e., $C_3(t_3)$ is created using $R_2(t_3)$).

The track associations can be represented using an *association matrix*. Shown in Figure 1 is an upper triangular matrix with each element of the matrix representing a particular pair of tracks. An "**X**" in any location indicates a positive association of the corresponding pair of tracks (by implication, any location without an "**X**" indicates a negative association). If there are n total tracks to be partitioned, then there are C_2^n (a standard notation for "n-choose-2," which is the number of ways that n objects can be taken two at a time without repetition) or $n(n-1)/2$ possible pair-wise associations. Finally, some elements of the matrix have a circle in them; these represent positive associations that *should* have been made. We shall make use of these circles when we define the incorrect and missed associations.

B. Venn Diagram Representation of Pairwise Associations

Figure 1 shows how the association of data can be represented as a collection of pairwise associations. We define the probabilities of missed association and incorrect association and calculate them using their frequencies of occurrence, which requires that we define a number of event sets. The first event set will be the set of all possible pair-wise associations between the collection of local tracks received over some time period of interest. Given a collection of n tracks, let $\mathcal{A}$ denote the set of all possible pairwise associations, and let $|\mathcal{A}|$ denote

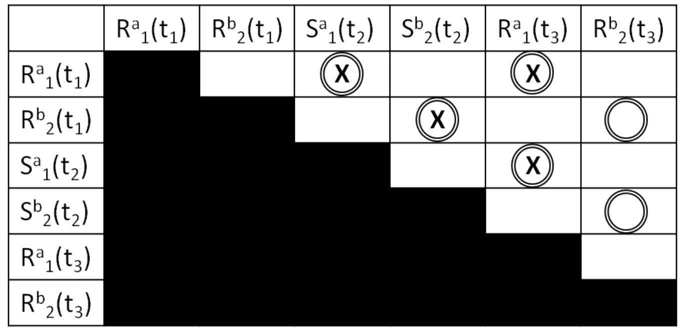

Fig. 1. Partition over all received local tracks (up through t_3) represented as a pair-wise association matrix.

its size or cardinality. As shown in Figure 1, for n tracks, $|\mathcal{A}| = n(n-1)/2$. We shall divide this set into two disjoint subsets:

- Let $\mathcal{A}_T$ be the set of all *true* associations based on the truth entities corresponding to each data item in a pair. Thus, $\mathcal{A}_T$ contains the pairs of data items that *should* be associated because they are both associated with the same truth entity (i.e., the circled cells in Figure 1).
- Let $\mathcal{A}_F$ be the set of all *false* associations based on the truth types corresponding to each track in a pair. Thus, $\mathcal{A}_F$ contains the pairs of data items that *should not* be associated because they are each associated with a different truth type.

As defined, $\mathcal{A}_T \bigcup \mathcal{A}_F = \mathcal{A}$ and $\mathcal{A}_T \bigcap \mathcal{A}_F = \emptyset$. The Venn diagram in Figure 2 shows the set of all pairwise associations partitioned into the sets $\mathcal{A}_T$ and $\mathcal{A}_F$.

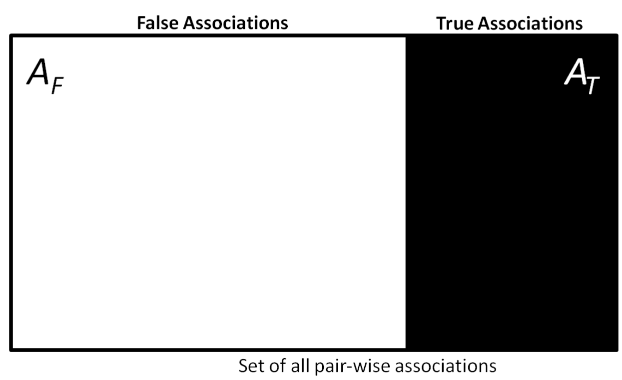

Fig. 2. Partition of all pairwise associations into true and false associations.

Next, consider a partition of the space of all pairwise associations into the set of positive associations, $\mathcal{P}$ (i.e., those associations made by the association algorithm) and the set of negative associations, $\mathcal{N}$ (i.e., those associations *not* made by the algorithm). These are shown in Figure 3.

Finally, putting all these sets into a single Venn diagram as shown in Figure 4 allows us to mathematically define the probabilities of missed and incorrect association. The figure shows the four sets of interest that correspond to the events

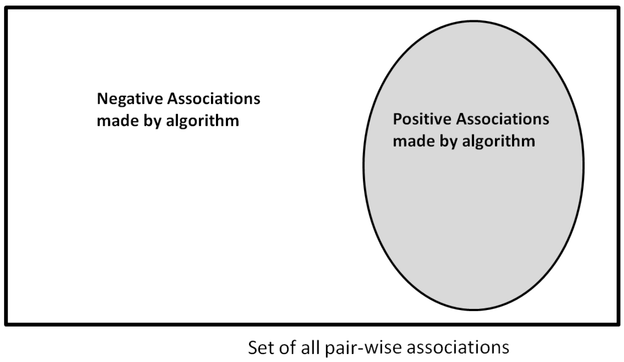

Fig. 3. Partition of all pairwise associations into positive and negative associations (i.e., those made by association algorithm).

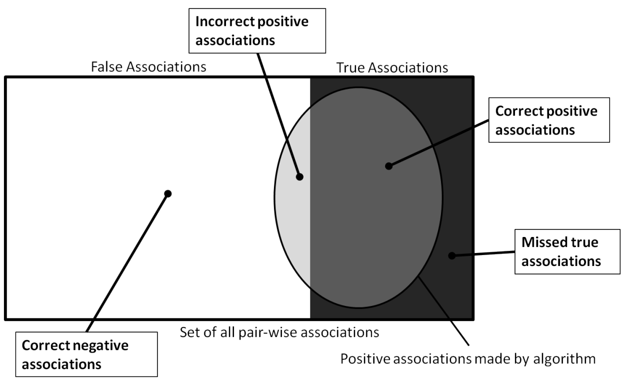

Fig. 4. Partition of all pairwise associations into true and false assocaitions.

defined in Section III, and we now define them in terms of set operations:

- Correct Positive Associations are also referred to as *correct associations* and defined as:

$$\text{Correct Positive Associations } = \mathcal{P} \bigcap \mathcal{A}_T \quad (1)$$

- Missed True Associations are also referred to as *missed associations* and defined as:

$$\text{Missed True Associations } = \mathcal{N} \bigcap \mathcal{A}_T \quad (2)$$

- Incorrect Positive Associations are also referred to as *incorrect associations* and are defined as:

$$\text{Incorrect Positive Associations } = \mathcal{P} \bigcap \mathcal{A}_F \quad (3)$$

- Correct Negative Associations are defined as:

$$\text{Correct Negative Associations } = \mathcal{N} \bigcap \mathcal{A}_F \quad (4)$$

C. Missed and Incorrect Association Probabilities

We have defined the missed and incorrect association sets. Each of these sets includes all pairwise associations that are categorized as missed or incorrect. To convert these to probabilities, we must define the space over which these pairwise association events occur; these would then be the denominators for finding the probabilities. We define the probabilities of missed and incorrect association as follows:

$$\begin{aligned} P(\text{Incorrect Association}) &\equiv P(\text{False Assoc.}|\text{Pos. Assoc.}) \\ &= \frac{|\mathcal{P} \bigcap \mathcal{A}_F|}{|\mathcal{P}|} \\ P(\text{Missed Association}) &\equiv P(\text{Neg. Assoc.}|\text{True Assoc.}) \\ &= \frac{|\mathcal{N} \bigcap \mathcal{A}_T|}{|\mathcal{A}_T|} \end{aligned} \quad (5)$$

Note that these are conditional probabilities since the denominator is not the set of all possible pairwise associations, $\mathcal{A}$. By choosing different spaces over which to calculate the probablities we are "grading" the association algorithm in two different ways: the incorrect associations are evaluated based on what the algorithm *did* do, while the missed associations are evaluated based on what the algorithm should have done.

An alternative is to use the set of all possible pairwise associations for both denominators, which grows as n^2 where n is the number of tracks, while the number of associations that should be made merely grows as n. Thus, an algorithm could fuse nothing (i.e., make each local track correspond to its own composite track) and the probability of missed association would go to zero as the number of local tracks increases, which is clearly undesirable.

With these definitions we can now calculate the probabilities of missed and incorrect association using the pairwise association matrix discussed in Section III-A. Refering to Figure 1, note that everything is there to calculate the probabilities. The **X**'s indicate the positive associations (the set $\mathcal{P}$), and the circles indicate the true associations (the set $\mathcal{A}_T$). The missed associations are those circles that do not have an **X** in them, and the incorrect associations are those **X**'s that do not fall within a circle. For this example:

$$\begin{aligned} P(\text{Incorrect}) &= \frac{0}{4} = 0.0 \\ P(\text{Missed}) &= \frac{2}{6} = 0.33 \end{aligned} \quad (6)$$

For comparison, consider the purity and conciseness metrics. At time t_3 we note from Table II that there is a single composite track (C_1) on Entity a but *two* composite tracks (C_2 and C_3) on Entity b. Thus, the conciseness is 50%. Each of the three composite tracks comprise local tracks associated with the same truth entities, so the purity of all three tracks is 100% and, therefore, the overall purity is 100%.

Continuing with the same example, the pairwise association matrix at time t_4 is shown in Figure 5. Using this matrix the probabilities of missed and incorrect association are:

$$\begin{aligned} P(\text{Incorrect}) &= \frac{3}{9} = 0.33 \\ P(\text{Missed}) &= \frac{6}{12} = 0.5 \end{aligned} \quad (7)$$

Consulting Table II again, there are two composite tracks (C_2 and C_3) on Entity b. Depending on how the truth entity of composite track C_1 is evaluated, it may be associated with

	$R^a_1(t_1)$	$R^b_2(t_1)$	$S^a_1(t_2)$	$S^b_2(t_2)$	$R^a_1(t_3)$	$R^b_2(t_3)$	$R^?_1(t_4)$	$R^b_2(t_4)$	$R^a_3(t_4)$
$R^a_1(t_1)$			⊗		⊗		x		○
$R^b_2(t_1)$				⊗		○		⊗	
$S^a_1(t_2)$					⊗		x		○
$S^b_2(t_2)$						○		⊗	
$R^a_1(t_3)$							x		○
$R^b_2(t_3)$								○	
$R^?_1(t_4)$									
$R^b_2(t_4)$									
$R^a_3(t_4)$									

Fig. 5. Association matrix at time t_4 showing which associations were made by the algorithm and which should have been made by the algorithm.

Entity a or the false entity. If it is associated with Entity a, then there would be two composite tracks on Entity a, and the conciseness would be 0%; otherwise, the conciseness would be 50%. Note that C_2, C_3, and C_4 are 100% pure while C_1 is 75% pure, yielding a total purity of 93.75%.

D. Growth of Pairwise Association Matrix

As is evident from the above discussion and example, as time progresses and more local tracks are received, the size of the pairwise association matrix increases. Not only is this a computational problem, it will also diminish the value of the metrics in the sense that at any point in time, the probabilities of missed and incorrect association include all association decisions made since the start of the fusion process. The metrics should quantify the performance of the algorithm around the time at which the metrics are calculated.

Consider a case in which a composite track was associated with truth Entity a for some period of time before switching to truth Entity b. During the time period around the switch, the probability of incorrect association should reflect this mistake. If, however, the accumulation of local tracks persists in the association matrix, the metrics will still include these incorrect associations long after they occur. Thus, the association matrix should consider only local tracks received during a sliding time window. In this way, calculating missed and incorrect association probabilities at time t_j will quantify the performance of the association algorithm around time t_j and not include decisions made since the beginning of the tracking problem. The size of the time window is a design parameter that should be a function of the update rate of the local tracks. A graphic illustrating the sliding window is shown in Figure 6.

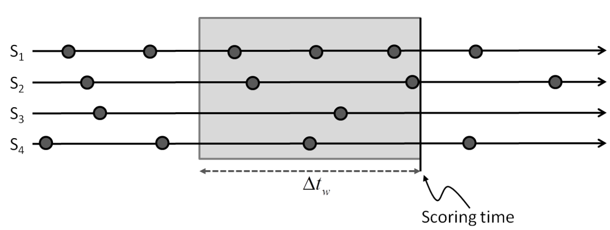

Fig. 6. Timeline of four sources of local track updates and the sliding window used to calculate the association metrics.

E. Analogous Metrics in Information Retrieval Domain

We now discuss performance metrics defined in the information retrieval domain; there is a direct analogy between the data association problem (in tracking) and the information retrieval problem. In the information retrieval domain, an algorithm searches for and retrieves documents that are deemed relevant given one or more keywords of interest to the user. Metrics were developed to quantify how well the information retrieval algorithm performed. The space of documents is divided into two sets in two different ways. First, the documents are divided into the set of *relevant documents* and *not relevant* documents. Where "relevant" means that the document is of interest to the user based on the search keywords. Next, the documents are divided into the set of *retrieved* documents (those gathered by the algorithm for the user) and the set of *not retrieved* documents. Given these set definitions, *precision* is defined as the fraction of retrieved documents that are relevant, and *recall* is defined as the fraction of relevant documents that are retrieved by the algorithm ([8], [9]).

These two metrics are analogous to our probability of incorrect and probability of missed association as we now show. First, note that the items that we are associating are measurements to tracks or tracks to tracks. In information retrieval, the items to be associated are documents and the keywords used in the search. The following indicate the equivalence in terminology between the two domains:

$$\begin{aligned} \text{Documents} &\iff \text{Measurements/Tracks} \\ \text{Retrieved documents} &\iff \text{Positive Associations} \\ \text{Relevant documents} &\iff \text{True Associations} \end{aligned}$$

Based on these equivalencies, the following relationships relate our metrics and the information retrieval metrics:

$$\begin{aligned} \text{Precision} &= 1 - P(\text{Incorrect Association}) \\ \text{Recall} &= 1 - P(\text{Missed Association}) \end{aligned}$$

One difference between the tracking and information retrieval problems is that the tracking problem has a time element to it as associations are made repeatedly as sensor data is received over time, while the information retrieval is really a *single event*, i.e., more of a static problem.

IV. Example Cases for Calculating the Association Metrics

In this section we present some examples of calculating the metrics. Some are notional and some are from running a simulation, generating local tracks, and feeding the local tracks to a T2T fusion algorithm. We first consider two notional examples that represent extremes of association algorithms.

A. Two Notional Examples of Association

When studying the performance of an algorithm or metric, it is often useful to consider theoretical scenarios that are at the extremes of performance. In this section we shall do just that by considering an association algorithm that associates *all* local tracks together and an association algorithm that does not associate *any* tracks together. For both of these examples, we shall consider the following scenario. Suppose that there are m entities being tracked in the surveillance region, and that there are k local trackers, each of which have one track on each of the m entities. Suppose that the time window discussed previously is such that within the window there is a set of tracks on all entities for each local tracker. The association matrix would have dimensions $mk \times mk$. For example for three entities and two trackers, the pairwise association matrix would appear as in Figure 7. There are a total of $mk(mk-1)/2$ pairwise associations possible. The number of true associations (i.e., those that should be made) can easily be calculated as $m\sum_{i=1}^{k-1} i = \frac{mk(k-1)}{2}$.

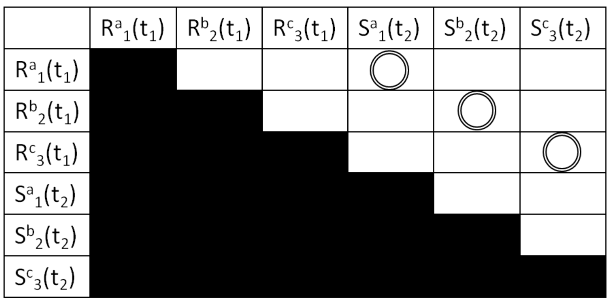

Fig. 7. An example pairwise association matrix for three entities and three local trackers (and a window that includes one set of updates from both trackers).

For the case of an aggressive association approach there would be $\mathbf{X}$'s everywhere since all tracks are fused into a single track. In this case the probability of missed is zero, and the probability of incorrect is given by:

$$P(\text{Incorrect Association}) = \frac{\frac{mk(mk-1)}{2} - \frac{mk(k-1)}{2}}{\frac{mk(mk-1)}{2}} = \frac{k(m-1)}{km-1} \approx \frac{m-1}{m} \quad (8)$$

Note that for the case of aggressive fusion, the probability of incorrect goes to one as the number of entities, m, gets very large. This is expected since as these increase, there are many more false associations than the number of true associations. On the other hand, for the case of extremely conservative fusion in which the association algorithm never combines tracks from difference sources (this is really "no fusion"), the probability of incorrect is 0, but the probability of missed association is 1.0.

B. Examples from Simulated Data

We now calculate the new metrics from results of a few different scenarios. In these examples, a set of surface entities are simulated, along with simulated sensor measurements from multiple sensors. Local trackers process the simulated measurements to generate local tracks, which are then processed by a T2T fusion algorithm to produce composite tracks. The new association metrics are calculated based on the performance of the T2T associaton algorithm. One of the scenarios is characterized by a failure to associate local tracks with existing composite tracks, while the other scenario is characterized by both incorrect and failed associations as we now describe.

1) Example 1: No Multi-Source Fusion: In this example, we ran a simulation with 20 surface entities that were widely separated, avoiding any incorrect associations. We simulated radar data and Automatic Identification System (AIS) data on these entities and processed the measurement data with local trackers. There are two platforms, each producing its own set of radar and AIS tracks. Platform A collects sensor data for the first four hours of the scenario, after which it stops collecting, and Platform B begins collecting data until the end of the scenario. The local radar tracks and AIS tracks were processed by a T2T fusion algorithm. The results of the association were evaluated using the new metrics. The T2T fusion algorithm was configured to avoid associating local tracks from different local trackers. There is still fusion taking place in that the T2T association algorithm does fuse local tracks to the composite tracks previously created on those local tracks, i.e., each local radar track and each local AIS track have a corresponding composite track to which all future updates on each local track are fused. Thus, there are generally 40 composite tracks; 20 that correspond to the 20 AIS tracks and 20 that correspond to the 20 radar tracks.

For this scenario, the percent incorrect and missed associations are plotted as a function of time for the 8.5 hour scenario in Figure 8. We chose a window size of 600 seconds for including tracks in the association matrix. There are two points about the graph worth noticing. First, the plot generally hovers around 18% missed association rate. Second, there is a spike that occurs at four hours. It is easy to determine that if the update rate of the radar tracks and the update rate of the AIS tracks is approximately the same, then the missed association percent averages 50%. In this scenario, the AIS update rate is approximately every 10 seconds, while the update rate for the radar is approximately 80 seconds, so the actual probability of missed association is not 50% but down near 18%.

The spike in missed associations at four hours occurs due to the receipt of a new set of local tracks from Platform B as it first comes online. As configured, the T2T fusion system will not fuse the initally received tracks from Platform B to any existing composite tracks, which is to say that it will not associate the newly received local tracks with the local tracks received from Platform A. Thus, for a short time period just after four hours, there are extra composite tracks on each entity, leading to a spike in the percent of missed associations. Since Platform A goes offline at four hours, the local tracks previously received from Platform A do not continue to get updated. As the local tracks from Platform A move outside the sliding window for evaluating association decisions, they drop out of the association matrix and the spike in missed associations disappears. The missed association percent returns to what it was during the first four hours of the scenario.

2) Example 2: Increased Association Ambiguity: In this second example, we simulated 50 entities that move as a group so that T2T association ambiguity exists and leaves the

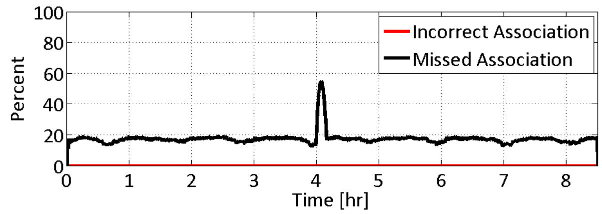

Fig. 8. Percent incorrect and missed associations plotted as a function of time for the case of no multi-source fusion. The spike results from newly received tracks from Platform B not being associated with existing tracks from Platform A.

algorithm prone to association mistakes. As in the previous example, Platform A is online for the first four hours of the scenario and generates two sets of local tracks. One set is based on simulated radar measurements, while the other set is based on simulated bearings-only measurements. At four hours Platform B takes over surveillance and also sends two sets of tracks to the T2T fusion node: radar tracks and tracks derived from bearings-only measurements. Depending on the size of the local track covariances relative to the entity spacing, there are times in the scenario where T2T association uncertainty exists, resulting in incorrect associations as shown below. The general behavior of the entities is:

- Hours 1 to 3.25: The entities start very close together and spread out as they move as a group on a linear path.
- Hours 3.25 to 5: Entities slow down towards zero. Many are stopped during this period. They also move farther apart.
- Hours 5 to 7.5: Entities turn around (rather abruptly) and return to where they started.
- Hours 7 to 8.5: Entities slow down to a stop. Most are stopped for a large part of the time.

A plot of the speed versus time for all 50 entities is shown in Figure 9. It is important to note that when the entities slow down, the probability of detection for the radar is quite reduced. In fact, many of them go undetected. Examining the local tracks we found that 30 of the entities do not have any radar track on them between 3.5 and 5 hours.

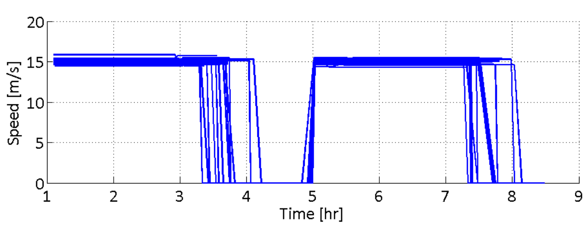

Fig. 9. Plot of all 50 entity speed profiles over time. Most slow down towards zero during the three to five hour interval.

We ran the resulting radar and bearings-only tracks through the T2T fusion algorithm and calculated the new metrics to evaluate the T2T association. The results are shown in Figure 10. During the first hour only radar tracks are generated on the entities, and there is a small number of missed associations that result from a few composite track breaks. At approximately 2.25 hours, the bearings-only tracks begin arriving at the fusion center and, due to the large uncertainty in the track state estimates (relative to the entity spacing, which is fairly tight), there is significant T2T association ambiguity that causes significant incorrect association.

To help highlight the amount of T2T association uncertainty at various times in the scenario, we have generated two plots. In Figure 11 is a plot of the square root of the trace of the horizontal position covariance for all 50 tracks, which assesses the uncertainty in the bearings-only derived tracks. The black lines correspond to the tracks from Platform A, while the grey lines correspond to the tracks from Platform B. The second plot is a measure of the distance between the entities. To quantify the spacing of the entities, we simply found the mean position of a collection of entities at a particular time, calculated the distance from each entity to the mean, and found the average over all these distances. A plot of this is shown in Figure 12. We do not always have truth positions on all 50 entities at all times; therefore, we sometimes get a subset of the entities that are closer (this accounts for some points on the plot that appear below the "line"). Note that between two and three hours, the entities are very close (Figure 12) resulting in the rapid increase in incorrection associations as the entities start moving between two and three hours.

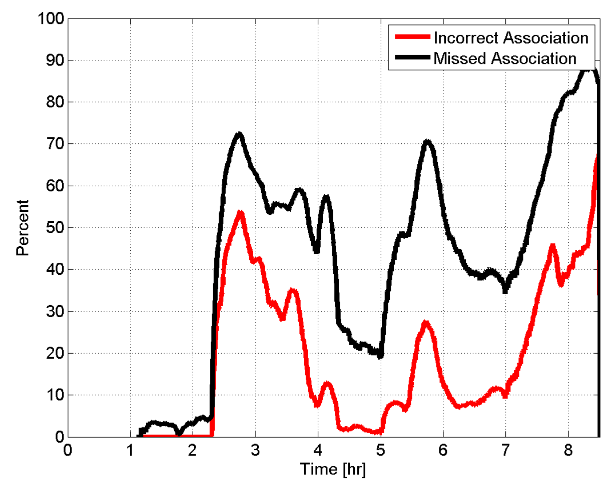

Fig. 10. Large uncertainty in bearings-only tracks leads to an increased rate of incorrect associations.

The percent of incorrect associations begins dropping just before three hours and continues to drop until five hours, at which point the incorrect associations are close to zero. There are two reasons for this. One reason is that many of the entities stop being detected by the radar, resulting in many less radar tracks that have to be associated and, therefore, less chance for incorrect or missed associations. More importantly, the entities are farthest apart, as shown in Figure 12, thereby reducing the T2T association ambiguity. Indeed, the precipitous drop in incorrect associations from three to five hours is accompanied by a rapid increase in the spacing of the entities.

The curve for missed associations generally follows the shape of the curve for the incorrect associations. Often (though not always), when we incorrectly associate local tracks with composite tracks, we simultaneously fail to associate these local tracks with the correct composite tracks. The missed associations are, thus, of two types. First, newly received local tracks from Platform B fail to associate to existing composite tracks and are used to create new composite tracks. Second, newly received local tracks from Platform B *are* associated with existing composite tracks, but the association is incorrect; thus adding to both the incorrect associations and missed associations.

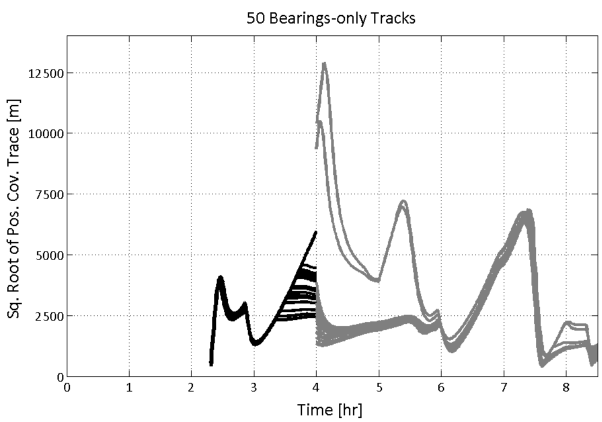

Fig. 11. Square root of the trace of the bearings-only track position covariances plotted for all 50 entity tracks over time.

At five hours, the entities begin to move a bit closer together as seen in Figure 12 and then stay at about the same spacing until 7.25 hours. The entities moving closer probably contributes to the increase in incorrect associations between five and six hours. Also contributing to the incorrect associations are two tracks that appear to have much larger covariances than the others as seen in Figure 11. The incorrect associations start to increase between 6 and 7.25 hours. While the entity spacing does not change much during this time, note that there is a rapid increase in the covariance size for all 50 tracks.

Finally, the size of the covariances drop rapidly around 7.5 hours, though the percent of incorrect still increases. The explanation is that while the covariances get smaller, the entities move much closer together from 7.3 hours to 8 hours, again causing association ambiguity and the continued rise in incorrect associations. Additionally, it turns out that there are also many newly created composite tracks during this last hour of the scenario, which is responsible for the increase in missed associations.

V. Conclusions

We have defined two new metrics for quantifying the performance of a data association algorithm that applies to both M2T fusion and T2T fusion algorithms. The probability of missed association and probability of incorrect association *directly* evaluate the decisions made by the association algorithm

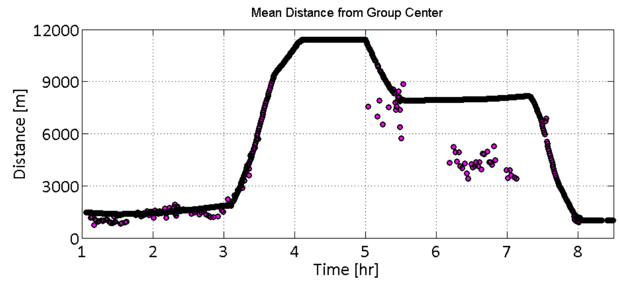

Fig. 12. Plot of the mean distance of an entity in the group from the mean position of the group. This gives a sense of the spread of entities.

and give the algorithm developer the ability to analyze the detailed behavior of the association algorithm at any point in time during a tracking scenario. We also showed how these metrics relate to the precision and recall metrics used in the information retrieval domain. As with any metric, usage over time will determine whether the new association metrics defined in this paper are more useful to algorithm developers than existing metrics. In presenting examples, notice how the plot of the metrics was used to help direct analysis of results in order to understand what happened and why it happened; this is how some metrics also serve as a great aid to performance analysis. Continued work will involve running the metrics on more tracking scenarios and comparing the results to the other metrics discussed here, purity and conciseness.

References

[1] J. Brandstadt, M. Kovak, J. Jones, and R. Gassner, "GMTI tracking performance," 2000. [Online]. Available: http://people.csmail.mit.edu/bushl2/rpi/portfolio/4.p4-NVA/ll/brandstadt_00.pdf

[2] D. Schuhmacher, B. Vo, and B. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. SP-56, no. 8, pp. 3447–3457, Aug. 2008.

[3] S. Mori, K. Chang, C. Chong, and K. Dunn, "Tracking performance evaluation - prediction of track purity," in *Proceedings of the SPIE AeroSense Conference on Signal Data Processing of Small Targets*, vol. 1096, Orlando, FL, March 1989.

[4] E. Blasch and P. Valin, "Track purity and current assignment ratio for target tracking and identification evaluation," in *Proc. of the 14th International Conference on Information Fusion*, Chicago, IL, July 2011.

[5] E. Blasch, O. Straka, C. Yang, D. Qui, M. Simandl, and J. Ajgl, "Distributed tracking fidelity-metric performance analysis using confusion matrices," in *Proc. of the 15th International Conference on Information Fusion*, Singapore, July 2012.

[6] "Single integrated air picture (SIAP) attributes," System Engineering Task Force (SETF), Single Integrated Air Picture SIAP, Tech. Rep., June 2001.

[7] K. Tarplee, D. Trawick, and S. Herman, "Distributed multiple-hypothesis correlation and feedback with applications to video," in *Proceedings of the SPIE AeroSense Conference on Signal and Data Processing of Small Targets*, vol. 6969, Orlando, FL, April 20008.

[8] "Precision and recall," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Precision_and_recall

[9] D. Powers, "Evaluation: From precision, recall, and f-factor to roc, informedness, markedness, and correlation," School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep., 2007.

[10] M. Silbert, "A robust method for computing truth-to-track assignments," in *Proceedings of the 12th International Conference on Information Fusion*, Seattle, WA, July 2009.