Ambiguous Data Association and Entangled Attribute Estimation

David J. Trawick, Philip C. Du Toit, Randy C. Paffenroth, and Gregory J. Norgard

Numerica Corporation, 4850 Hahns Peak Drive, Suite 200, Loveland, CO, 80538, USA

ABSTRACT

This paper presents an approach to attribute estimation incorporating data association ambiguity. In modern tracking systems, time pressures often leave all but the most likely data association alternatives unexplored, possibly producing track inaccuracies. Numerica's Bayesian Network Tracking Database, a key part of its Tracker Adjunct Processor, captures and manages the data association ambiguity for further analysis and possible ambiguity reduction/resolution using subsequent data.

Attributes are non-kinematic discrete sample space sensor data. They may be as distinctive as aircraft ID, or as broad as friend or foe. Attribute data may provide improvements to data association by a process known as Attribute Aided Tracking (AAT). Indeed, certain uniquely identifying attributes (e.g. aircraft ID), when continually reported, can be used to define data association (tracks are the collections of observations with the same ID). However, attribute data arriving infrequently, combined with erroneous choices from ambiguous data associations, can produce incorrect attribute and kinematic state estimation.

Ambiguous data associations define the tracks that are entangled with each other. Attribute data observed on an entangled track then modify the attribute estimates on all tracks entangled with it. For example, if a red track and a blue track pass through a region of data association ambiguity, these tracks become entangled. Later red observations on one entangled track make the other track more blue, and reduce the data association ambiguity. Methods for this analysis have been derived and implemented for efficient forward filtering and forensic analysis.

Keywords: Attribute Estimation, Attribute Aided Tracking, Classification Aided Tracking, Multiple Hypothesis Tracking, Uncertainty Management, Dynamic Bayesian Networks, Late Data

1. INTRODUCTION

In modern tracking systems, time pressures often leave all but the most likely data association alternatives unexplored, possibly producing track state estimate inaccuracies. If a tracking system application is only interested in estimating the kinematic properties (position, velocity, etc.) of the targets it encounters, this is generally a sensible approach to computation reduction. Once the targets pass through a region of data association ambiguity and return to a region of (relatively) unambiguous data association, the estimates of the kinematic values of a target regain their accuracy as the influence of the ambiguous region's data fades away with time. Nevertheless, the state estimates during and shortly after regions of data association ambiguity may contain inaccuracies, primarily because the estimates of target properties are being made with observations corresponding to more than just the one target.

Alternatively, if a tracking system seeks to estimate some other properties of its targets, data association errors are far less benign. This is particularly true if the contribution of an observation does not fade away over time, such as observations of fixed characteristics of a target (for example, target classification or target ID). Attributes are defined as such non-kinematic discrete sample space sensor data. They may be as distinctive as aircraft ID, or as broad as a classification of a target as a friend or foe.

Attribute data may provide improvements to data association by a process known as Classification Aided Tracking (CAT)[1,2] or Attribute Aided Tracking (AAT). Indeed, certain uniquely identifying attributes (e.g. aircraft ID), when continually reported, can be used to *define* the data association (tracks are formed from all observations with the same unique

Further author information: (Send correspondence to D.J.T.)
D.J.T.: E-mail: david.trawick@numerica.us, Telephone: 1 970 461 2422 x223
P.C.D.: E-mail: philip.dutoit@numerica.us, Telephone: 1 970 461 2422 x253
R.C.P.: E-mail: randy.paffenroth@numerica.us, Telephone: 1 970 461 2422 x233
G.J.N.: E-mail: greg.norgard@numerica.us, Telephone: 1 970 461 2422 x205

aircraft ID). Figure 1 shows how this approach works. Note that in this case there is no estimation of the attribute; it is assumed that the attribute observation is always correct and always available (given with every observation).

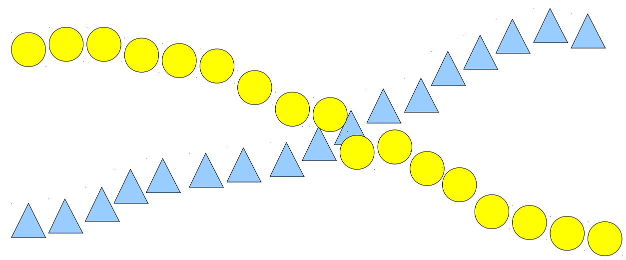

Figure 1. Attribute Aided Tracking Example. The targets are traveling from left to right. Here the attribute is shape, taking on the values of circle and triangle. The attribute is reported with every observation. The observations are placed in this picture according to the observed positions in the tracking space. The data association is defined by the attribute in this case, so that there is one circle track and one triangle track.

However, some attribute reports: (1) may come from "specialized" sensors that provide updates only every once in a while, (2) may require significant computation time to determine, or (3) may experience significant network delays, thereby making these data either infrequent, late, or both. Such attribute data may be *critical* in the sense that they are very valuable in resolving ambiguities in tracking and combat identification; thus, the motivation to use these data is high even though there are complexities in applying it. This scenario is illustrated in Figure 2.

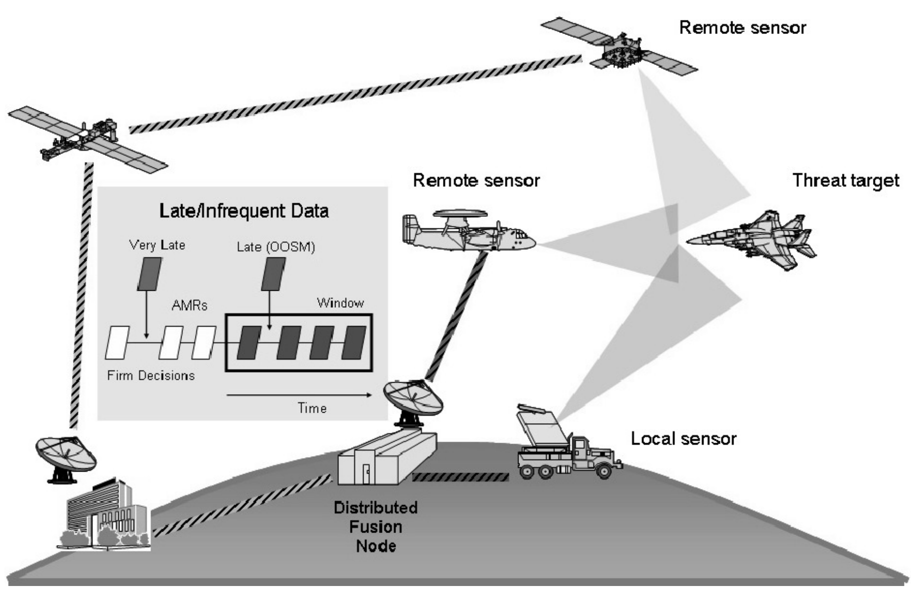

Figure 2. Diagram showing the arrival of late/infrequent data at a distributed (networked) tracking/fusion node. Data arrive at the local node from the remote airborne surveillance radar late because it has a long scan period compared to the local electronically scanned array radar. Data arrive very late from the satellite-based sensor because they must be routed via satellite links, processed at ground stations, and then forwarded to the local node. Both remote platforms can provide critical information that can be used to resolve data association and combat ID ambiguities. Note that here we are addressing issues of dealing with the very late data. The issues of more recent late data, such as Out Of Sequence Measurements (OOSM), have been considered in detail elsewhere.[3,4]

Suppose then we have the following situation. A tracker receives *critical*, unambiguous attribute data, but the data arrive infrequently and/or late. Suppose further that the tracker only retains the most likely of its ambiguous data associations,*

*Although there are some trackers that retain some data association ambiguity for a short period of time, most trackers are forced to and that this chosen data association happens to be in error. This is the situation illustrated in Figure 3 when new attribute data arrive. Note that the new attribute observations implicitly reveal that the previous data association decision was incorrect.

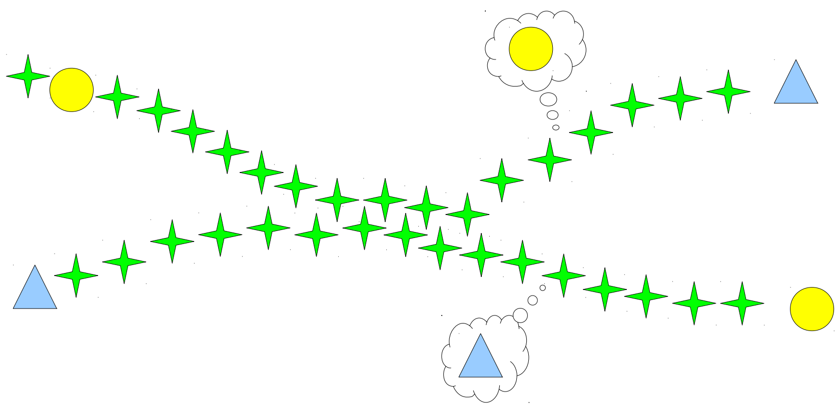

Figure 3. Issues arising from firm data association decisions. Suppose we have the scenario similar to Figure 1, except that now the attributes are reported infrequently. The four-pointed stars are the kinematic only observations. A single, firm data association (no crossing) was chosen incorrectly. The attribute estimates (in the clouds) have been propagated through the chosen data association. The next data association decision is problematic. For example, a triangle is forced to be on a circle track and a circle is forced to be on a triangle track.

What is the tracker to do now? The kinematic evidence for how to associate the new observations to the existing tracks is in direct contradiction to the attribute evidence, and both are nearly certain. A data association decision based only on the attribute evidence will try to undo the previous data association error, leading to strange kinematic results. A data association decision based only on the kinematic evidence will corrupt the attribute estimates by effectively forcing the combination of attribute reports from different targets.

To summarize: attribute data arriving late or infrequently, combined with data associations forced to be unambiguous and resulting in error, can produce incorrect attribute and/or kinematic state estimation. However, if we retain the fact that a previous data association was ambiguous, and modify our attribute estimates to reflect this uncertainty, we can be prepared for whatever and whenever new observation data arrives.

The appropriate modeling of this situation requires the attribute estimate to correctly capture the full attribute uncertainty distribution. Furthermore, correct attribute modeling leads to attribute estimates for a track that depend directly on the attribute estimates of other tracks, where the tracks involved become linked together by their history of ambiguous data associations. We call such ambiguously associated tracks *entangled*.

The rest of this paper is organized as follows. First, we describe Numerica's Tracker Adjunct Processor (TAP), which contains our methods for analyzing and managing data association uncertainty and handling late or infrequent data. Next we show a detailed analysis of a simple example of attribute estimation through a region of data association ambiguity. Then in Section 4, we generalize this analysis and make it suitable for a recursive implementation. Next, we show some examples of the algorithm in action. Finally, we summarize the results and draw some conclusions.

2. BACKGROUND

Identifying and quantifying the regions of data association ambiguity is a necessary first step in mitigating the possible errors resulting from such ambiguity. Numerica's Bayesian Network Tracking Database (BNTD), a key part of its Tracker Adjunct Processor (TAP), captures and manages the data association ambiguity for further analysis and possible reduction/resolution using subsequent data.

One requirement of modern tracking systems is that they provide the best overall situational awareness given the data and the processing time available. These constraints imply that a lot of interesting, even critical, analysis cannot practically

choose the most likely of their ambiguous data associations at some point.

be performed by the tracker. The key design innovation needed is then to divide the responsibilities of a tracking system into those that must always be kept up-to-date and those that involve non-real-time decisions. This organization allows us to process certain computations outside the tracking system proper, either as time allows or as needed by the tracking system. We have identified this tracking system support process as the *Tracker Adjunct Processor*,[5] or TAP.

Some examples of non-real-time tracking tasks suitable for the TAP include the following: (1) handling of critical, late or infrequently reported data (the focus of this effort); (2) reconnection of broken tracks; (3) termination of tracks no longer being observed by the sensor; (4) identification of redundant or spurious tracks caused by not correctly associating local sensor data to system tracks; (5) detection of the "incompatible" fusion of multi-sensor data to system/network tracks based on incongruous kinematic or feature properties.

The TAP has as its core component Numerica's *Bayesian Network Tracking Database* (BNTD).[6,7] The BNTD uses an analysis from an associated Multiple Hypothesis Tracking (MHT) system† to form a Bayesian Network[10–12] representing the input observation data and the ambiguous multiple data association hypotheses from the tracker. It can then use Bayesian network methods to propagate the impact of new evidence throughout the network, and report on the consequences of such new data or hypothesized data to the track picture. This forms a very flexible and general approach for addressing the non-real-time issues described above.

The core Bayesian network in use by the BNTD is a dynamic Bayesian network, meaning that the network is designed to model dynamic processes, specifically the kinematic state estimates of its targets. As such, it uses dynamic system models in much the same way that the filtering and smoothing parts of a tracking system does. Indeed, the BNTD can be thought of as an unrolled filterer/smoother. This network is extended to retain the ambiguity of the considered data associations by representing and retaining the ambiguous connections among the observations of interacting track-like paths through the network. This results in a track picture that is a network or graph of possible paths, rather than collections (hypotheses) of non-interacting paths. This design facilitates the inclusion of late or infrequent data; just find where the newly arrived data fits into the network, connect it to the network, and update as necessary.

The BNTD is designed to bring to bear all the evidence available (all the observations input to the system) to forming the state estimates at any given point. As such, the processing is equivalent to a fixed interval smoother extended to model data association ambiguity. While this full smoothing of the data is useful for a variety of applications (including forensic analysis), it is neither desirable nor necessary to always smooth the entire data set. The BNTD is useful for a variety of applications when only modeling the (forward) filtering of a tracking system. It may do this by performing only a forward pass through the network, which it may do while building the network. When the BNTD is used in this way, the computation time necessary is comparable to that of a typical tracking system, with the addition of time for building the network and the data association ambiguity modeling.

3. A DETAILED ANALYSIS OF A SIMPLE ATTRIBUTE ENTANGLEMENT

We begin by considering how to estimate a simple attribute state along an unambiguous sequence of associated observations, or a path. Suppose there are several observations of an attribute along a sequence of unambiguously associated kinematic observations of a target, i.e., along a track with no data association ambiguity. The unambiguous estimation of the attribute state works as follows.

Suppose we have targets that are either red or not (the attribute to estimate), and we have a sensor that observes whether a target is red or not. Several parameters of the sensor's performance are known. Let $X \in \{r, n\}$ be the target's true red or not state, and let $S \in \{r, n\}$ be the sensor's measurement of the target's state. Then the sensor has the characteristics $p(S = r|X = n) = P_f$, the probability of false alarm; and $p(S = n|X = r) = P_m$, the probability of a missed detection. From these values we can determine $p(S = r|X = r) = 1 - P_m = P_d$, the probability of detection, and $p(S = n|X = n) = 1 - P_f = P_o$, the probability of a true no detection.

Let the notation $[U, V]$ indicate the state of the red or not attribute distribution at some time, where U is the probability of red and V is the probability of not red. Then a sensor report R of red is the distribution $[P_d, P_f]$, and a sensor report N of not red is the distribution $[P_m, P_o]$. Since we keep the probabilities of all possibilities in this notation, it is easily renormalized by dividing by the sum, i.e., $[U, V]$ becomes $[U/(U + V), V/(U + V)]$ after renormalization. This notation

†Such as Numerica's Multiple Frame Assignment (MFA) tracking system.[8,9]

yields a simple update formula: as new sensor attribute reports are received, we multiply the probabilities term by term and renormalize.[‡]

Now consider what happens in the case of ambiguous data association; for example, where some data association choice other than the best rises above some threshold. Figure 4 provides a simple example of data association ambiguity as it would be represented in the BNTD. Two targets are moving from left to right. At the center of the picture, the targets were close enough together that it was no longer clear how to associate the observations; did the targets cross or not? The goal here is to understand how observations of attribute data at various points in the scenario will contribute to the attribute state estimates throughout the network.

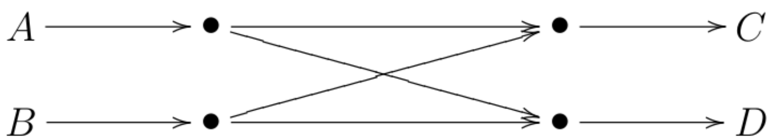

Figure 4. Diagram showing a simple data association ambiguity scenario.

Before diving into the calculation of probabilities, let us examine what is going on in this situation. Even if we know exactly the attribute values at the inputs A and B, the data association ambiguity will cause the attribute estimations to become both ambiguous and dependent on each other. For example, if A is red $([1,0])$ and B is not $([0,1])$ (unambiguously), then the values after the possible swap will be completely ambiguous $([.5,.5])$, if the swap probability was 0.5. Furthermore, an observation on path C of red should draw out the probability of red along the D path; observations along C and D influence each other's estimates. Does this mutual influence ever go away? Perhaps if some level of purity of the attribute estimate is achieved, yielding a nearly unambiguous estimate of the attribute.[§] But in general, if we know nothing more concerning the attribute value along these paths (have no more observations), the attribute value estimates along these paths will remain dependent on each other.

Now let us consider the specifics of how to calculate the probabilities involved in this scenario. The additional ambiguity here is that the tracks either cross or do not cross. The associated tracking system provides the probabilities of these events from its analysis of the data association possibilities: let P_X be the probability of crossing. Also let the labeled paths A, B into and C, D out of the ambiguous region represent the attribute state estimates for those unambiguous paths outside of the ambiguous region, so that $A = [P_A, (1-P_A)] = [p(\text{A path red}), p(\text{A path not red})]$, etc.

The probability distribution estimate $\hat{P}_C$ at point the C can then be computed as a total probability as in equation 1. Here we enumerate the possibilities of A, B, C, D red or not, and of the targets crossing X or not. For example, the first line of equation 1 is the product of the probabilities that A, B, C, D are all red and that there was no crossing. The first sum is the sum of the terms with C red, and the second sum is the sum of the terms with C not red.

$$
\begin{array}{lcccccc}
\hat{P}_C = [& P_A & P_B & (1-P_X) & P_C & P_D & + \\
 & P_A & P_B & P_X & P_C & P_D & + \\
 & P_A & (1-P_B) & (1-P_X) & P_C & (1-P_D) & + \\
 & (1-P_A) & P_B & P_X & P_C & (1-P_D) & , \\
 & (1-P_A) & (1-P_B) & (1-P_X) & (1-P_C) & (1-P_D) & + \\
 & (1-P_A) & (1-P_B) & P_X & (1-P_C) & (1-P_D) & + \\
 & (1-P_A) & P_B & (1-P_X) & (1-P_C) & P_D & + \\
 & P_A & (1-P_B) & P_X & (1-P_C) & P_D &].
\end{array}
\tag{1}
$$

[‡]The attribute estimate is initialized with the prior distribution of the attribute.

[§]Section 4.6 discusses some approaches to detecting and processing such disentanglements.

Now we would like to transform this formula into an approach to attribute estimation that can be updated in a simple recursive manner, much like the single-path case above. If the computation generally works in time order,[¶] it would be nice to be able to update with a term by term multiplication as above. Inspection of the column with the P_C terms shows that updates from a new observation $Z_C = [P_z, (1 - P_z)]$ along the C path can indeed be multiplied term by term, as this new observation would multiply this way as just a new term along the C path contributing to the original $C = [P_C, (1 - P_C)]$ estimates.

However, this total probability formula (equation 1) for this simple example shows the dependence of the probability at the point C on data along the D path, and inspection of the column with the P_D terms does not admit a simple term by term multiplication with new observation data along the D path. A reorganization of the terms in the total probability formula into more terms will provide a version more suitable for such term by term updates, by explicitly keeping terms for red and not red outputs, for both paths C and D. This is made as follows, where the terms here are [(C red, D red), (C red, D not), (C not, D not), (C not, D red)]:

$$
\begin{array}{rcccccl}
\hat{P}_{C,D} = [& P_A & P_B & (1-P_X) & P_C & P_D & + \\
 & P_A & P_B & P_X & P_C & P_D & , \\
 & P_A & (1-P_B) & (1-P_X) & P_C & (1-P_D) & + \\
 & (1-P_A) & P_B & P_X & P_C & (1-P_D) & , \\
 & (1-P_A) & (1-P_B) & (1-P_X) & (1-P_C) & (1-P_D) & + \\
 & (1-P_A) & (1-P_B) & P_X & (1-P_C) & (1-P_D) & , \\
 & (1-P_A) & P_B & (1-P_X) & (1-P_C) & P_D & + \\
 & P_A & (1-P_B) & P_X & (1-P_C) & P_D &].
\end{array} \tag{2}
$$

This reorganization of the equation 1 is the key idea in managing the calculation of the attribute probability estimates for ambiguous data association. Note that a term is kept for each attribute value (red or not here), for each output path (C and D in this case). It is this bundle of probability estimates that we call the attribute (estimate) *entanglement*. It is organized in such as way as to facilitate the update of attribute estimates as new data are encountered: probabilities from new data update the terms corresponding to the output path on which the new data are found. It is also organized to facilitate the determination of the attribute probability estimates for a given path: the terms corresponding to that path define the marginalization of interest to be computed.

Table 1 shows another way to view the terms of the above equation, and another way to understand the interaction of the pieces of an attribute entanglement. The dimensions of Table 1 (horizontal or vertical) correspond to the paths leading out of the ambiguous region. There are entries in the table for every combination of attribute values on the output paths. Updates and marginalizations take place along the dimension of the path involved. This table may be thought of as a matrix of values; in general for more output paths than two it is a tensor of values. The values here correspond to the sums in equation 2; the first sum is in the top left position (C red, D red); the other sums follow around the table counter-clockwise. We will expand and make all of these notions more explicit in the next section.

		C Output Path	
		C red	C not
D Output Path	D red	C red, D red	C not, D red
	D not	C red, D not	C not, D not

Table 1. Example Attribute Entanglement for Two Tracks. This is the tensor form of the attribute estimate entanglement for the data association ambiguity of Figure 4, going forward.

We note that our prior work concerning ambiguity and uncertainty in attribute estimation[5] considered Uniquely Identifying Attributes (UIAs), attributes whose values must correspond to only one target (such as aircraft IDs). The earlier

[¶]The insertion of late data is generally repropagated through the network in a time-ordered manner.

approach attempted to estimate the attribute probabilities without the notion of an entanglement. This required a post processing step to maintain the uniqueness of the attribute values across the various estimates throughout the system. While this hints at the type of entanglement described here, unlike the entanglement method it did not generalize well to attributes whose values are not unique to only one target (such as target classification, friend/foe).

4. THE ALGORITHM

In this Section we will generalize and make specific the notions of the previous Section. In particular, we will discuss how to recognize when tracks are entangled, how to determine the attribute values involved in an entanglement, how the attribute entanglement estimate data structure is built and updated from attribute observation data, how a particular path's attribute estimate is determined from the attribute estimate entanglement, and how tracks may become disentangled. In all of this discussion we assume that there is only one attribute to be estimated, and that there are only a finite number of values that the attribute in question may have. Also, the analysis here is for estimates moving forward in time, for the forward filtering estimates. The methods for backward filtering and smoothing of entangled attribute estimates are similar.

4.1 Recognizing Entangled Tracks

The tracking system associated with the BNTD delivers its data association problems for further analysis of the data association ambiguity. The k-best solutions to this data association problem are then determined. If there is a non-negligible likelihood of solutions other than the best solution, the data association is considered ambiguous and the tracks involved are considered entangled. If a track extension is part of all the k-best solutions, it is considered to be independent of the entanglement; all the ambiguity is elsewhere. Similarly, all the non-interacting entanglements of the k-best solutions are separated into independent entanglements.

The developing entanglements at this stage consist of the input and output nodes, N_I and N_O (corresponding to observations in the BNTD's network, and labeling the paths in and out of the entanglement) of the data association problem, and the hypotheses $h \in H$ from the k-best solutions connecting the input and output nodes. The likelihoods of the hypotheses $P(h)$ are also available from the k-best solutions. The input nodes are the observations already associated with the rest of the network, the observations already (possibly ambiguously) associated with tracks. The output nodes are the observations to be (ambiguously) associated with the network. The input and output nodes define the input and output paths of the entanglement.

4.2 Determining Joint Probabilities to Estimate

What we are ultimately building in an entanglement is a joint distribution of attribute values and the entangled output paths. For this we need to know the values the attribute A can have. Let these be denoted as the values $v \in A$. Typically this will be available from the attribute values present on the input paths. If there are no attribute estimates on any of the input paths, there will be no attribute entanglement.

Next, we want to assign attribute values to all the output paths to enumerate all possible outcomes. A particular assignment of attribute values to the output paths will be called a *syndrome*. This is just a vector of attribute values, where each output path corresponds to one position in the vector. Given the attribute values and the output paths, all possible output syndromes $s \in S$ can be enumerated.

Thus, the (C red, D red) of Table 1 is one possible output syndrome of the entanglement of Figure 4; the other possible output syndromes are listed in the Table. There the output paths were mentioned explicitly, where from now on they will be determined solely by position within the syndrome vector.

4.3 Building an Attribute Entanglement Estimate

Now that all the pieces are ready we can build the attribute entanglement estimate corresponding to the data association ambiguity in question. Let $P_Q(v), \forall v \in A$ be the distribution provided by an observation Q of an attribute A. Then the entanglement distribution P_E is determined by:

$$\forall s \in S : P_E(s) = \sum_{h \in H} P(h) P_{N_O}(s) P_{N_I}(h^{-1}(s)), \tag{3}$$

$$P_{N_O}(s) = \prod_{q \in N_O} P_q(s(i_q)), \tag{4}$$

$$P_{N_I}(s) = \prod_{q \in N_I} P_q(s(i_q)). \tag{5}$$

Note that $s(i_q)$ is the value of the syndrome at the index for path/node q, which is just an attribute value. Furthermore, $h^{-1}(s)$ is the back trace through the hypothesis h of the syndrome s, that is, the input syndrome producing the output syndrome s. Also, P_q for $q \in N_I$ is the attribute distribution estimate on the input node/paths, which may be a prior, an observation node's distribution, or the attribute estimation on that path from an earlier entanglement. (The entanglement probabilities $P_E(s)$ should be normalized after construction.)

Algorithm 1 describes the steps necessary to build the entanglement data structure.

Algorithm 1 Build the Attribute Entanglement Tensor

```
procedure BuildEnt(S, H)                                  ▷ Build The Entanglement Tensor P_E
    for s_O ∈ S do                                         ▷ Collect likelihoods by out-syndrome
        P_E[s_O] ← 0.0                                     ▷ Initialize P_E for out-syndrome
        for h ∈ H do
            s_I ← h^{-1}(s_O)         ▷ Pass out-syndrome back through the hypothesis to get in-syndrome
            P_{N_O}(s_O) = ∏_{q∈N_O} P_q(s_O(i_q))         ▷ Get likelihood at h output for out-syndrome
            P_{N_I}(s_I) = ∏_{q∈N_I} P_q(s_I(i_q))         ▷ Get likelihood at h input for in-syndrome
            syndromeLike ← P_{N_I}(s_I) * P(h) * P_{N_O}(s_O)   ▷ Likelihood for current s_O, h combination
            P_E[s_O] ← P_E[s_O] + syndromeLike
        end for
    end for
    return P_E                                             ▷ return the Entanglement Tensor P_E
end procedure
```

Since several paths out of an entanglement all share the data in the entanglement estimate, the track state along a path is extended to include a pointer to the (shared) entanglement. The track state must also include the index, or the entanglement tensor dimension, of the output path to which it corresponds.

4.4 Updating an Attribute Entanglement

New observations with attribute data must now be used to update the entangled attribute estimate. If a new observation is part of an ambiguous data association, a new entanglement is generated as before with the new observation on one of the output paths. If the new observation is unambiguously associated with the existing data, but is associated to a path from an earlier entanglement, the entanglement is updated with the new data as described here.

Again let S be the set of syndromes for an entanglement E. Let the observation O be on E's output path p, and let i_p be the index of p in the syndrome vectors. If P_E is the distribution of E, this distribution is updated by the observation attribute distribution P_O as follows:

$$\forall s \in S : P_E(s) \leftarrow P_E(s) P_O(s(i_p)). \tag{6}$$

In this manner all the terms of P_E receive an update from the new observation O. Note that the entanglement probabilities $P_E(s)$ generally require renormalization after this step.

4.5 Retrieving a Path's Attribute Estimate

When considering a track we typically want to know the estimates corresponding to that track as opposed to the details of its entanglements with other tracks. Thus, we need a way to extract the attribute estimation information for a specific track from the entanglement. The tracks here correspond to the output paths of an entanglement. Since the entanglement is setup as a joint distribution for all the participating output paths, to get the values for a specific path is to do a marginalization over all paths except the one of interest. Specifically, continuing with the assumptions as before, if the attribute distribution estimate on path p is P_p (what we want to find, initialized to all 0):

$$\forall v \in A : P_p(v) = \sum_{\substack{s \in S, \\ s(i_p)=v}} P_E(s). \tag{7}$$

4.6 Removing a Track from an Entanglement

Does a track/path, once entangled, ever become disentangled? We would like a way to remove the entanglement processing and data structures when they no longer provide an advantage over the attribute estimation along non-interacting tracks. According to the formulas above, the entanglement will not go away as long as there is uncertainty in the data association or the inputs and outputs to the data association. As long as the input data are uncertain, the entanglement will remain. Nevertheless, we can attempt to detect when the estimates of either the hypothesis or the output values become nearly certain.

Initially the tracks were not entangled if for some hypothesis $h \in H$, the probability $P(h)$ was nearly certain. This is one method to approach disentanglement; whenever the reestimated $P(h)$ becomes 1.0 (within some threshold), we can say that these tracks are disentangled.

Another approach is to notice when the output syndrome of an entanglement becomes determined, i.e., some attribute value along every path has a probability near 1.0. This is more useful as we are continually reestimating these values as the scenario develops.

This is an area we have not fully explored, with several interesting questions remaining. For example, could one track be extracted from an entanglement, leaving the rest entangled? Could an entanglement be subdivided into several non-interacting sub-entanglements?

5. SOME SIMPLE EXAMPLES

In this Section we present three examples of the entangled attribute estimation in action. These simple scenarios were all constructed to contain challenging ambiguous data association examples, but be small enough to easily understand. Our examples have the targets traveling from left to right. We present two versions of each example, one in diagram form similar to Figure 3, and one from the output of the TAP system. In the diagrams, the attribute will again be shape. In the snapshots of actual output, the example attribute is color, where the attribute values for the targets are typically either red or blue. The diagrams should clarify the snapshots of output as the scenarios become more complex.

The first example is a continuation of the analysis in Section 3 of the scenario of Figure 4. Figure 5 shows the setup, with a circle and a triangle target moving through a region of data association ambiguity. (The four-pointed stars are kinematic observations with no attribute information, and the arrows show the data association possibilities above some threshold.) The probabilities of crossing or not are equal (0.5) in this scenario. After the ambiguity, the attribute estimates on both paths exiting the ambiguity are ambiguous. (The attribute distribution estimates reported in this Figure and the following ones are the distributions on the paths out of the entanglement.) The similar actual output snapshot is given in Figure 6.

Note that even at this point the entanglement has provided something useful: the information that the attribute estimates are ambiguous. If the attribute estimates were based on a prematurely chosen data association, the reported attribute values could lead to an incorrect response to the track picture, the consequences of which could lead to engaging the wrong target. This is not to say that the ambiguous estimate is the perfect situation; if there are both friend and foe targets, we need to know who's who. Nevertheless, a true estimate of the uncertainty of the track picture provides the operator with the most correct information available for any upcoming engagement decisions.

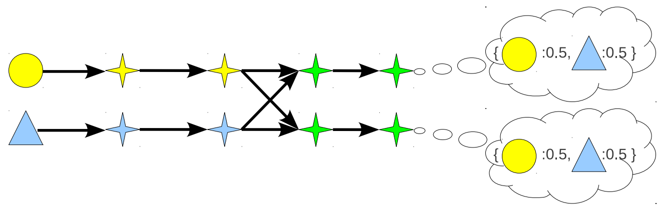

Figure 5. Simple crossing scenario setup. Two targets, traveling left to right, one with an attribute of circle, the other with an attribute of triangle, cross with a probability of 0.5. The attribute values after the ambiguous crossing are a 50-50 mix of circle and triangle. No attribute observations have been made after the ambiguous crossing.

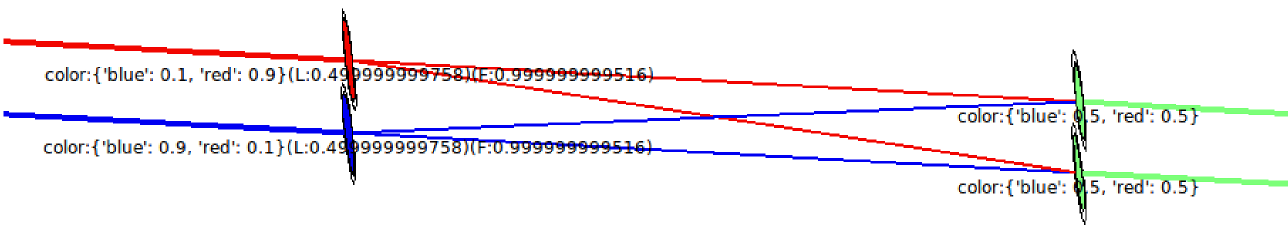

Figure 6. Simple crossing scenario snapshot of actual output, using red and blue attributes, corresponding to Figure 5.

In Figure 7 a new attribute observation has arrived and the track picture has been updated. A triangle observation was reported that was associated with the lower track. Suppose this change was enough to lower the ambiguity probabilities to where we can claim disentanglement. This effectively resolves the attribute ambiguity of the lower track; it is a triangle. This also resolves the attribute ambiguity of the upper track; it is a circle. Furthermore, this new observation has essentially resolved the ambiguity of the prior data association; there was no track swap. Alternatively, if the new observation did not disentangle the tracks, then the entanglement would be updated and propagated forward. (Figure 8 shows the forward propagation of the entanglement estimates for actual output.)

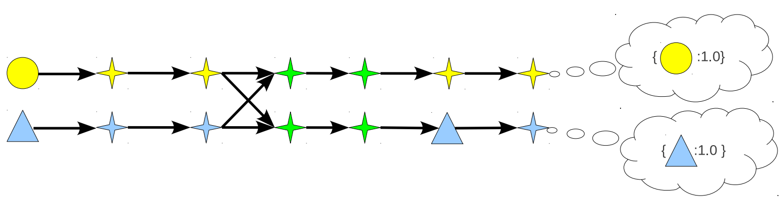

Figure 7. Simple crossing scenario after new attribute observation. An attribute observation of triangle on the lower track changes the attribute estimation of the lower track to triangle *and* the upper track to circle.

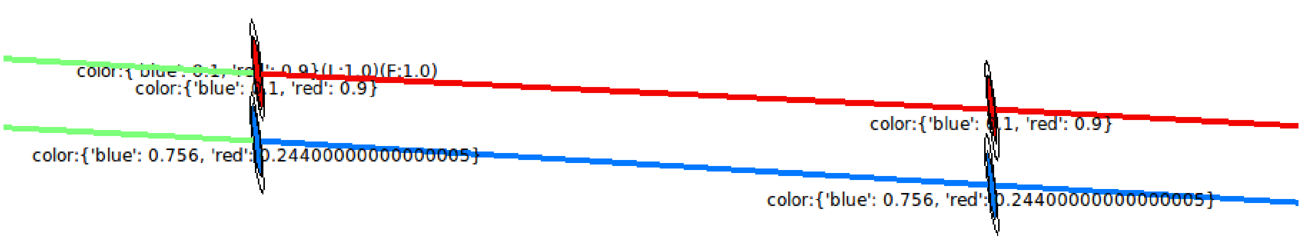

Figure 8. Simple crossing scenario snapshot of actual output, using red and blue attributes, corresponding to Figure 7. Here the new attribute observation was red on the upper track.

Note that the new attribute observation, updating one track initially, now updates the attribute estimates on all tracks of the entanglement. This is not a function of how close the tracks are together; the same change would have taken place even if the tracks had moved far apart. This change is rather a function of the entanglement of the tracks. Similar to its quantum mechanical namesake, the attribute estimate entanglement produces a kind of peculiar action at a distance within the track picture as the uncertainty is reduced.

The next example involves three targets traveling from left to right. The upper and lower targets appear to go straight while the middle target appears to trace out an "S"-shaped path, first becoming ambiguous with the lower target and then with the upper target. Let the middle target be a circle and the upper and lower targets be triangles at the beginning. This is the situation in Figure 9. First, the lower two target attributes mix (as in Figure 5) and then the upper two target attributes mix. Figure 10 shows a snapshot of the first crossing of a similar actual scenario.

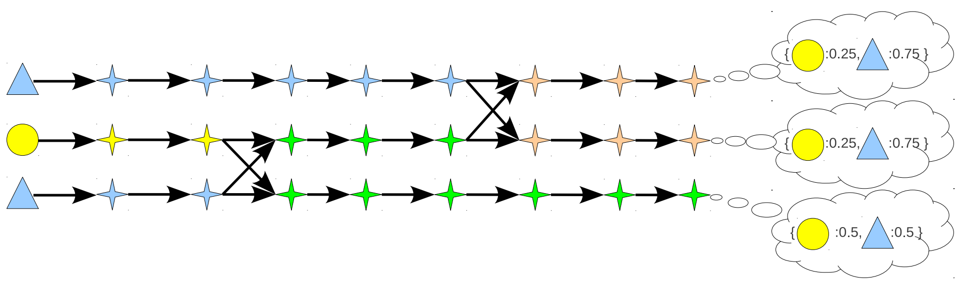

Figure 9. S-scenario setup. Three targets travel left to right. At the beginning, the outer two have an attribute of triangle, and the middle one is circle. The apparently middle target is first ambiguous with the lower target, and then later with the upper target.

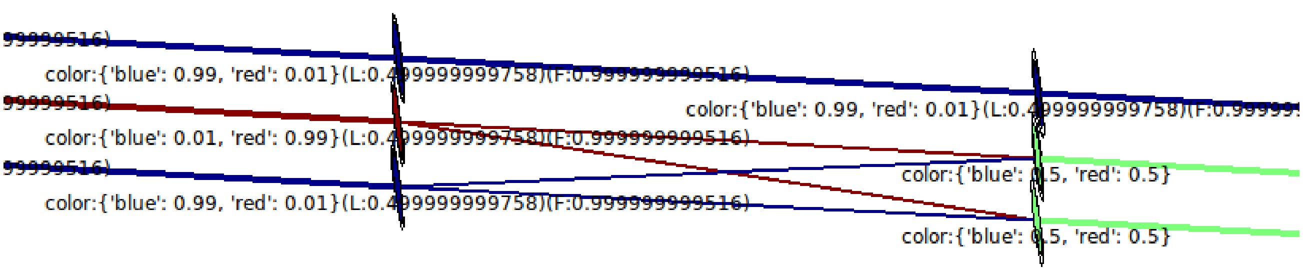

Figure 10. Output snapshot of a scenario similar to Figure 9, using red and blue attributes. (The later ambiguity is not in this Figure, but its output is in the starting (left) portion of Figure 12.)

Some time after the two ambiguous crossings of this scenario, an attribute observation reports that the currently middle track is a circle. This changes the attribute value of the middle target to circle. This also changes the upper and lower targets to have a triangle attribute value. Figure 11 shows the resulting scenario.

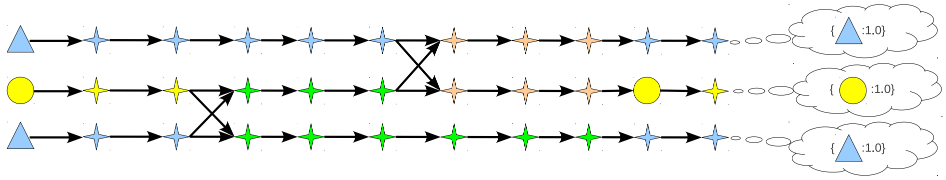

Figure 11. S-scenario after new attribute observation. An attribute observation of circle on the currently middle target after the two ambiguities resolves the attribute ambiguity in the entire scenario. The middle target now has an attribute of circle, *and* the two outer targets now have an attribute value of triangle.

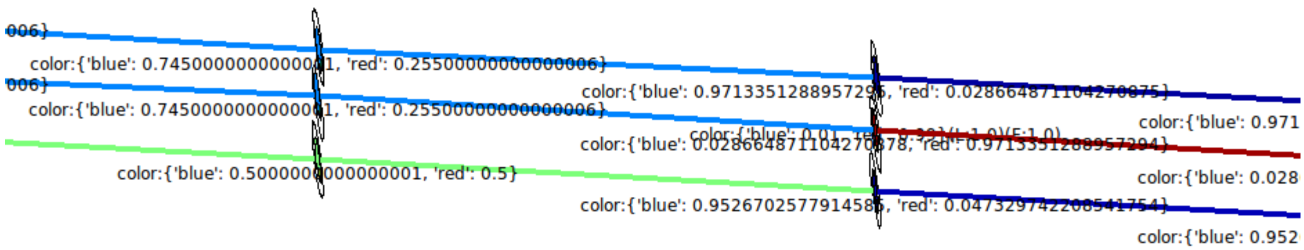

Figure 12. Output snapshot of a scenario similar to Figure 11, using red and blue attributes. The ambiguity is resolved. (The left of the figure shows the output of the second ambiguous crossing of the scenario, between the middle and upper targets.)

Note that this latest single observation has cleared up all the ambiguity of the scenario. We now know how the final targets connect to the initial targets of the scenario. We note here that the implementation we have made also includes backward attribute estimation and smoothing, so that the ambiguity of the entire scenario may be resolved, including the region between the two ambiguous crossings (that is, if the probabilities say the tracks can be disentangled).

The next example scenario has three targets traveling from left to right. It is known that the upper target is a circle and the lower two targets are triangles at the start (left) of the scenario. There is a completely ambiguous crossing of all three targets in the middle of the scenario. The estimated attribute values upon exit of the ambiguous crossing is the blend of all the inputs: $\frac{1}{3}$ circle, $\frac{2}{3}$ triangle, on all output paths. Figure 13 shows the scenario at this time. A similar actual output scenario is in Figure 14.

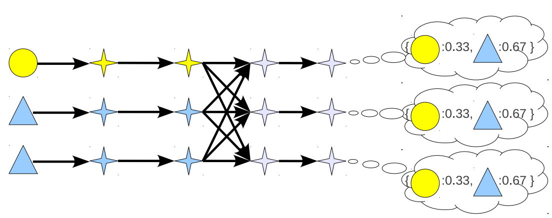

Figure 13. Three target ambiguous data association scenario setup. Three targets, traveling from left to right, ambiguously cross in this scenario. The upper target is a circle, the lower targets are triangles. The attribute estimates after the ambiguous crossing represent a mix of the attribute values prior to the ambiguous crossing.

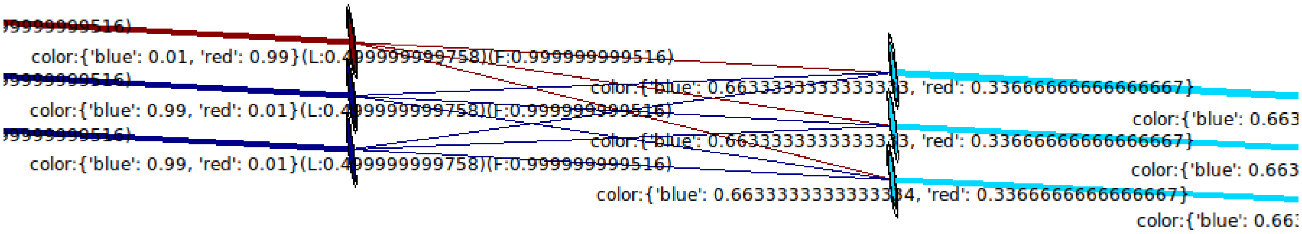

Figure 14. Output snapshot of a scenario similar to Figure 13, using red and blue attributes.

After the ambiguous crossing an attribute observation of triangle is reported on the lower target. This disambiguates the lower target (making it triangle) *and* removes its triangle contribution from the entanglement of the upper two targets. The attribute estimates of the upper two targets now reflect the extraction of the lower triangle target. This is the situation present in Figure 15.

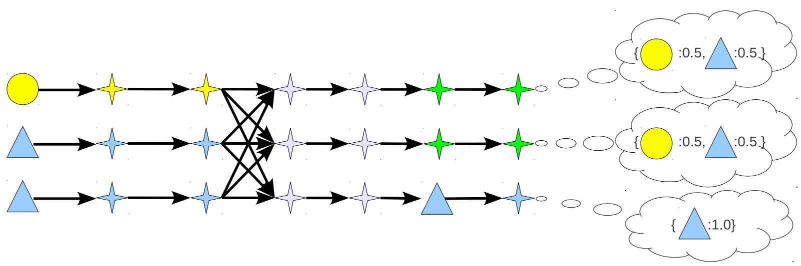

Figure 15. Three target ambiguous data association scenario after new observation. An attribute observation of triangle on the lower target effectively removes that target from the entanglement. The two upper targets remain entangled with each other. The triangle observation has pulled one of the triangle targets out of the entanglement.

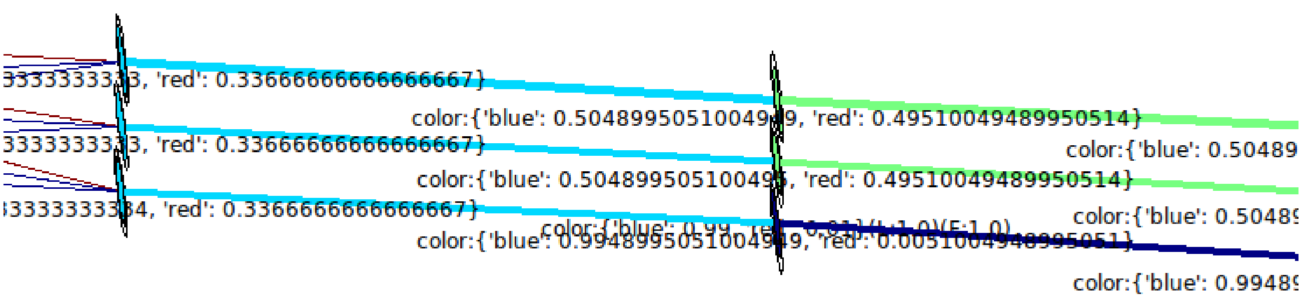

Figure 16. Output snapshot of a scenario similar to Figure 15, using red and blue attributes.

Note from this example that what remains ambiguous after the uncertainty is reduced is just as important as reducing or resolving the ambiguity. The updated entanglement allows us to capture the remaining ambiguity in a manner that can be updated correctly with any subsequent attribute observations.

6. A COMPLEX SCENARIO

In this Section we show the results of entangled attribute processing on a more complex scenario, using data simulated with a realistic benchmark. This scenario contains multiple networked sensors and multiple interacting targets. This example shows that the entanglement methods are suitable for working with tasks of greater complexity than the simple toy problems of Section 5.

An overview of the scenario is shown in Figure 17. This shows the complex interactions of the multiple targets. There are about 30 targets in this scenario. The duration of the scenario is about 800 seconds. There are six sensors in this scenario of three types: fixed phased array, rotating phased array, and airborne phased array. The sensors are arranged in a realistic defensive placement.

One target of the scenario was designed to explicitly add data association ambiguity. This target follows the path roughly in the middle of Figure 17, moving from center-lower-right toward left center. It becomes ambiguous with two of the turning targets, first with the turning target near the center of the Figure, then with the turning target off to the left. We use the color attribute in this example. Let the turning targets have attribute values of blue and the leftward moving target have an attribute value of red. These interactions then essentially form the interactions of the S-scenario as in Figures 9 and 10, where here the turning targets correspond to the upper and lower targets of the earlier Figures. Figure 18 shows the ambiguity of the attribute estimates after the times of both regions of explicit ambiguity. The first pair of entangled tracks have changed to a green color, representing the roughly equal blend of red and blue. After the second ambiguous region, the exiting tracks are a light blue, representing roughly equal blend of the previous blend with another blue track. At this time, the three tracks we have been talking about are now entangled.

Figure 17. A view of the full duration of the complex scenario is shown. No attribute estimation is represented in this view.

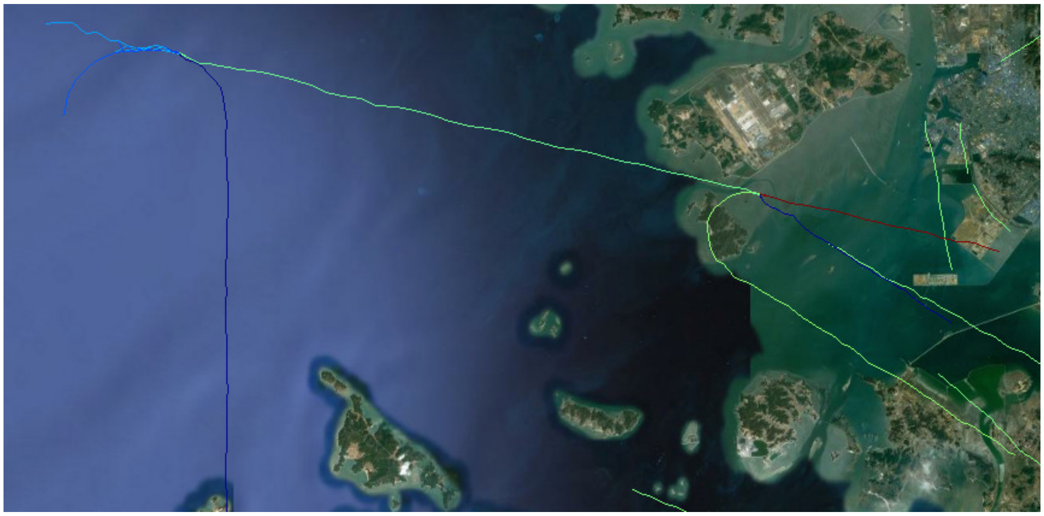

Figure 18. Here the attribute estimate ambiguity resulting from the ambiguous data association regions is shown, with most of the tracks having no attribute information, two having an initial (dark) blue attribute and one having a red attribute.

Next, a red attribute observation arrives and is put into the track moving to the left and slightly upward, after the ambiguous regions. Entanglement processing coveys this updated information to all the tracks that are members of the entanglement, so as this track becomes red, the others become (darker) blue. Figure 19 shows the changes taking place on all three tracks involved. The left portion shows the changes near where the new observation was added. The lower right of this Figure shows the long distance ambiguity resolution resulting from the disentanglement.

Figure 19. A new red observation was added to the leftward moving, non-turning target as shown in upper left, where the track turns red. The resulting disentanglement between this track and the turning track nearby is seen in the lower left, where the downward moving track returns to a darker blue. The resulting disentanglement between these tracks and the earlier turning one is seen in the lower right, where that track has also returned to the darker blue.

7. DISCUSSION

In this Section we discuss several issues related to the understanding and use of entanglements.

How do entanglement updates report changes to the tracking system? Specifically, are attribute estimates pulled from or pushed to the tracking system? We have tentatively assumed throughout that whenever the tracking system needs to report the attribute estimate for a track, it will query the TAP/BNTD system. This may not always be an appropriate method of operation. It may be more appropriate for the TAP/BNTD to send messages of its updates whenever the attribute estimate of a track changes. This may be necessary, in particular, to communicate that changes have taken place, especially in the case where the change to an attribute entanglement changes the attribute estimate on a track that the new attribute observation is not associated with.

What other estimates, if any, should be entangled? Normally in our experience the observations associated with a track will only update the estimates associated with that track. This experience is primarily from kinematic observations. The idea of entanglements is contradictory to this experience, where attribute observations contribute to the estimates for all tracks with which they are entangled. Are there other estimates for which estimate entanglement is appropriate? Target features, continuous valued characteristics of a target such as signal to noise ratio, would appear to require similar entanglements. However, the dynamic nature of such estimates will make such analysis more difficult.

How do entanglements affect BNTD efficiency? Earlier we claimed that the BNTD is efficient largely due to the fact that it is primarily a dynamic Bayesian network, a Bayesian network composed of dynamic models. Attribute estimation, especially involving estimates on entangled tracks, put the modeling of the BNTD closer to that of a more typical discrete probability Bayesian network. To maintain the efficiency of the current BNTD, it is recommended that the attributes modeled be either: (1) attributes with few values, such as friend or foe; or (2) uniquely identifying attributes, such as aircraft identification number. Also, while the kinematic modeling of the BNTD blends its estimates following ambiguity, thereby eliminating the need for maintaining duplicate hypotheses, attribute propagation through ambiguity must entangle the estimates, leading to more complex computation. Thus the management of data association ambiguity in attribute estimation requires more subsequent processing than kinematic estimation.

With regards to kinematic estimation, several enhancements are being made to the speed of the BNTD's smoother. A variant of a Rauch-Tung-Striebel (RTS) smoother[13] has been developed that is capable of handling ambiguous associations. In comparison to the currently used Fraser-Potter type smoother,[14] this new smoother should substantially reduce computation time, while sacrificing a small amount of accuracy. While this smoother is not presently fully incorporated into the BNTD, prototype code suggests, using the Kalman filter, there is no loss of accuracy with computation time cut approximately in half. Using an IMM filter, there is a $1 - 10\%$ increase in error with computation time roughly a third of the previous approach. These speed enhancements will be documented in a future publication.

How can we quantify performance changes from entanglements? In lieu of a detailed set of performance tests we consider here the basic benefits to be derived from attribute entanglement estimate processing. (Any such tests would be highly dependent on the ambiguity present in the scenarios chosen.) Suppose then that two targets, identified as a friend and a foe, pass close to each other so as to produce a data association ambiguity. Suppose further that the ambiguity has a clear but not overwhelming winner: the targets cross with a probability of .3, and travel without crossing with a probability of .7. The targets then travel far apart with no more attribute (friend or foe) observations. Now suppose the target with the foe attribute makes a threatening move. Should the foe be engaged?

If there is no representation of data association and hence attribute estimate ambiguity, the answer is clearly yes, although there is a .3 chance of being wrong. If the ambiguity is available for inspection prior to a fire order, the operator would probably wait for confirming evidence, since at this point there is an expected fratricide probability of .3.

Now suppose that there is an observation of this attribute, not on the threatening track but on the track with which it is entangled. Let the new observation report a .99 chance that the associated target is a friend. Then, if there is no entanglement processing, there is no change to the situation. Perhaps an astute operator can make the connection, but this is less likely if the targets are now far apart. Entanglement processing, however, updates the attribute estimate of both the associated track and the entangled track (the threatening track), providing the confirmation needed to the operator for engagement.

8. CONCLUSIONS

Attribute estimate entanglements have been developed as a means of representing the joint nature of the developing attribute estimates of the tracks involved in data association ambiguity. The uncertainty present in the ambiguous data association is essentially propagated along with the attribute estimates, so that any new attribute evidence will update the attribute estimation on all the tracks involved. New evidence may also lead to the conclusion that a track has become disentangled, where its attribute estimates are no longer dependent on other tracks.

ACKNOWLEDGMENTS

This work was supported by the US Army Space and Missile Defense Command under contract W9113M-08-C-0083. We would also like to gratefully acknowledge the support of many others at Numerica.

REFERENCES

[1] Drummond, O. E., "Feature, attribute, and classification aided target tracking," in [*Proceedings of SPIE, Conference on Signal and Data Processing of Small Targets*], **4473**, 542–558 (2001).

[2] Bar-Shalom, Y., Kirubarajan, Y., and Gokberk, C., "Tracking with classification-aided multiframe data association," *IEEE Transactions on Aerospace and Electronic Systems* **41**, 868–878 (July 2005).

[3] Chan, S. and Paffenroth, R. C., "Out-of-sequence measurement updates for multi-hypothesis tracking algorithms," in [*Proceedings of the SPIE, Conference on Signal and Data Processing of Small Targets*], Drummond, O. E., ed., **6969** (2008).

[4] Bar-Shalom, Y. and Chen, H., "IMM estimator with out-of-sequence-measurements," *IEEE Trans. on Aerospace and Electronic Systems* **41**, 90–98 (January 2005).

[5] Trawick, D. J., Slobumb, B. J., and Paffenroth, R. C., "A tracker adjunct processing system for reconsideration of firm tracker decisions," *Proceedings of the SPIE, Conference on Signal and Data Processing of Small Targets* **7698-44** (April 2010).

[6] Obermeyer, F. H. and Poore, A. B., "A Bayesian network tracking database," *Proceedings of the SPIE, Conference on Signal and Data Processing of Small Targets* **5428**, 400–418 (August 2004).

[7] Slocumb, B. and Trawick, D., "Bayesian Network Tracking Database: Final Report," Missile Defense Agency SBIR Phase II Final Report Contract N00178-04-C-3007, Numerica Corporation (July 2006).

[8] Poore, A. B., "Multidimensional assignment formulation of data association problems arising from multitarget tracking and multisensor data fusion," *Computational Optimization and Applications* **3**, 27–57 (1994).

[9] Poore, A. B. and Robertson III, A. J., "A new class of Lagrangian relaxation based algorithms for a class of multidimensional assignment problems," *Computational Optimization and Applications* **8**(2), 129–150 (1997).

[10] Pearl, J., [*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*], Morgan Kaufmann (1988).

[11] Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J., [*Probabilistic Networks and Expert Systems*], Springer-Verlag, New York (1999).

[12] Ghahramani, Z., "An Introduction to Hidden Markov Models and Bayesian Networks," *International Journal of Pattern Recognition and Artificial Intelligence* **15**, 9–42 (2001).

[13] Rauch, H. E., Tung, F., and Striebel, C. T., "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal* **3**(8), 1445–1450 (1965).

[14] Fraser, D. C. and Potter, J. E., "The optimal linear smoother as a combination of two optimum linear filters," *IEEE Trans. on Automatic Control* **AC-14**, 387–390 (August 1969).