Distributed Multiple Target Tracking and Data Association in Ad Hoc Sensor Networks

Maurice Chu
Palo Alto Research Center
Palo Alto, CA, U.S.A.
mchu@parc.com

Sanjoy Mitter
EECS Department
M.I.T.
Cambridge, MA, U.S.A.
mitter@mit.edu

Feng Zhao
Palo Alto Research Center
Palo Alto, CA, U.S.A.
zhao@parc.com

Abstract – *We introduce an efficient distributed algorithm for tracking multiple targets in an ad hoc sensor network. The key idea is to explicitly model an intermediate representation, which we refer to as individual track contributions, so that the effect of measurements observed by a node of the sensor network can be incorporated locally. Furthermore, since targets affect the observations of a local neighborhood of sensor nodes, communications are limited to local regions around currently known targets. A particularly important problem in multiple target tracking is the problem of data association. Since a sensor network is capable of measuring heterogeneous data, we divide the possible resolutions of ambiguity in data association into three categories. Discriminatory information is: (1) immediately available to resolve the ambiguity in data association, (2) delayed to resolve the ambiguity, and (3) unavailable. In this paper, we demonstrate the incorporation of target identity information when discriminatory information is available and a factored representation of tracks when discriminatory information is unavailable. We build efficiency into the tracking algorithm by triggering the extraction of discriminatory information and triggering the use of such information during tracking only when it is deemed necessary.*

Keywords: Multiple target tracking, classification, data association, sensor network, distributed algorithm.

1 Introduction

Current technology enables us to engineer sensor rich systems capable of collecting massive amounts of data with the potential to extract information about the system state and their environment to an extent which has been unimaginable or prohibitively expensive in the past. With recent advances in sensing, embedded computation, and wireless communications, it is possible to construct networks consisting of a large number of heterogeneous elements, each with the ability to sense its environment in various modalities, process data and extract information, and communicate the results of these computations with each other. These *sensor networks* offer a new kind of system concept: the coordinated collaboration of a network of distributed autonomous agents working together to extract the desired information from the data collected locally by each of the agents.

Of particular importance among information extraction problems is the ability to track objects, either physical targets moving in the physical world or some abstract entity moving in some abstract state space with data collected by the sensor network. Many information extraction problems can be posed in such a tracking framework, and for sensor networks to truly proliferate in complex and cluttered real environments, multiple target tracking is a problem that must be solved efficiently. Due to constraints on resources, like limited battery life and communication, and the physical distribution of data throughout nodes of the sensor network, a distributed algorithm for tracking multiple targets is necessary since there is not enough bandwidth to transmit all data to some central node and employ a centralized approach.

There exist two main approaches for handling multiple target tracking in the literature, the multiple hypothesis tracker (MHT) [7] and the joint probabilistic data association filter (JPDAF) [1]. These two approaches were originally constructed for a centralized system where observations are possible location estimates extracted from radar measurements. Our multiple target tracker is a distributed algorithm using more general sensing models with consideration for communication costs. Furthermore, we have access to more information about target identity from the sensors so that it is possible to resolve data association ambiguity in certain cases. Previous tracking work on sensor networks include the IDSQ algorithm ([3], [9]) for single target tracking. Our work can be considered an extension of the leader-based tracking idea to the multiple target case. The idea of sensor selection is not addressed here although the presented algorithm can be extended to include a selection aspect.

The construction of computationally efficient, local algorithms implementing global tasks is a main theme in sensor network applications. A general methodology is proposed in [2] for structuring the distributed solution of global inference problems, and the multiple target tracking solution presented in this paper is an instance of applying that methodology. Details of the general methodology are not discussed here although some application specific discussion will be given in Section 5.

In this paper, we present an overview of the basic distributed track update algorithm and the ideas for dealing with the data association problem. Details can be found in [2]. Section 2 describes the multiple target tracking problem, assumptions we make, and what we consider a solution. Section 3 presents the sensing models used to demonstrate the different aspects of the tracking algorithm. Section 4 then describes the basic track update algorithm followed by a description of how and when to collect and incorporate classification information to handle the data association problem in Section 5. We conclude with simulations and discussion in Section 6.

2 Problem Formulation

The problem is to build a distributed multiple target tracker implemented on a sensor network with heterogeneous sensing modalities requiring minimum cross node communications. From the collected sensor data, the multiple target tracker is to compute a set of tracks, which are time series of position estimates, that correspond to the true trajectories of targets moving through the sensor network. Tracking multiple targets, as opposed to tracking a single target, introduces the additional problem of *data association*, which has been studied in the context of radar measurements. Abstractly, radar measurements give estimates about possible positions of true target locations at each time so that the data association problem becomes the combinatorial problem of associating particular position estimates computed from radar measurements with each current track computed by the multiple target tracker.

More generally, the effect of multiple targets on a sensor is that the measurement observed by a sensor is some combination of the individual contributions from each target. Thus, the problem of data association for more general sensing models is to determine what part of a sensor measurement corresponds to a specific track. This more general data association problem can be significantly aided with discriminatory information about the target, and we divide the range of availability of discriminatory information for data association into three categories: (1) discriminatory information is *immediately available* to resolve the ambiguity in data association, (2) discriminatory information is *delayed*, meaning that this information is not immediately available but is available at a later time, and (3) discriminatory information is *unavailable*.

We will refer to the inference of a target's position over time as *tracking* and inference of any discriminatory information about a target other than position as *classification*. In order to resolve data association ambiguity, discriminatory information must exist prior to an ambiguous data association event. Thus, successful multiple target tracking necessitates the classification of targets and the incorporation of this classification information when needed to resolve data association ambiguities.

Let us describe the assumed characteristics and information known by nodes of the sensor network. Nodes of the sensor network are immobile and spatially dense enough that sensor data about targets is highly redundant. Thus, tracking and classification can be performed using data from a small local subset of sensor nodes around each target. We assume that each sensor node is aware of its own location in some suitably chosen global coordinate system (like from GPS) as well as the locations and sensing modalities of all of its one-hop neighbors. Furthermore, all sensor nodes are assumed to be equipped with powerful processors able to implement intensive computations. Our algorithm assumes that wireless communications between one-hop neighbors are reliable with no dropped packets or collisions, and delays in transmission and reception are negligible. Making the algorithm robust in less benign network conditions is a future research direction.

3 Sensing Models

To demonstrate the distributed multiple target tracking algorithm, we present representative sensing models of three broad classes of measurements: signal intensity, binned signal intensity, and discrete classification measurements. In particular, Section 3.1 presents simple single target to single sensor models for sound intensity, frequency bandlimited sound intensity, and magnetometer measurements which are meant to be illustrative rather than realistic. Then, Section 3.2 presents how to combine these individual models to generate a model from multiple targets to multiple sensors.

3.1 Individual Measurement Models

The three types of measurement models used to demonstrate the distributed tracking algorithm are intensity, frequency bandlimited intensity, and magnetometer measurements. All three models below can be encoded by a likelihood function

$$p(z|x,\theta) \tag{1}$$

where z is the sensor measurement, x is the location of a single target, and θ is any other class information about the target.

The first two sensing modalities, intensity and frequency bandlimited intensity, are derived from omnidirectional microphone signals and are an example of features extracted from continuous signals.

We model the intensity measurement extracted from the microphone signal by

$$z_I(t) = Af(||x(t) - x_I||^{\alpha}) + n(t) \tag{2}$$

where A is the known nominal intensity of the sound at the target, $x(t)$ is the location of the target at time t, x_I is the location of the microphone, α is an attentuation coefficient depending on atmospheric conditions, $n(t)$ is a zero-mean, white Gaussian process with variance σ^2, and f is a monotonically nonincreasing function corresponding to the attenuation of the intensity with distance raised to α between target and microphone. This simple model of the intensity allows for negative intensity measurements which is certainly unrealistic since intensity of a signal is a non-negative value, but when the target is close to the microphone, the probability of $z_I(t)$ being negative is small enough that we will not worry about this. The intensity measurement model (Eqn. 2) implicitly defines a likelihood function where z is $z_I(t)$, x is $x(t)$, and θ is A. The other parameters are specific to the particular microphone sensor.

The frequency bandlimited intensity measurement is a vector of intensity measurements $\underline{z}_F(t)$, where each component corresponds to the intensity of the signal in a chosen frequency band. The model is given by

$$\underline{z}_F(t) = \underline{A}f(||x(t) - x_F||^{\alpha}) + \underline{n}(t) \tag{3}$$

where all vectors are b-dimensional, $\underline{A}$ is the set of nominal intensities of the microphone signal in each frequency band which we call the *frequency signature*, x_F is the location of the microphone, $\underline{n}(t)$ is a multivariate, zero-mean, white Gaussian process with covariance $\frac{\sigma^2}{b}I$ where I is a $b \times b$ identity matrix, and all other parameters are the same as in the intensity measurement model (Eqn. 2). Note that the frequency bandlimited intensity measurement is a refinement of the intensity measurement in that the intensity measurement is the sum of the vector components of the frequency bandlimited intensity measurement. Thus, computing frequency bandlimited measurements gives frequency information about the target as well as position information at the price of more computational effort. We will assume known possible classes of frequency signatures to demonstrate the use of frequency information for resolving data association ambiguity. Note that we assume that the attenuation with distance is the same in all frequency bands which is not true in the physical atmosphere over large distances. However, for small enough distances, this simplifying assumption is not too inaccurate.

The third sensing modality is derived from readings of a magnetometer. This model is an example of a discrete measurement giving classification information and is given by

$$z_M(t) = \begin{cases} 1 - n(t) & , \ ||x(t) - x_M|| < r, \theta_{metal} = 1 \\ n(t) & , \ \text{otherwise} \end{cases} \tag{4}$$

where $n(t)$ is a Bernoulli random variable equalling 1 with small probability p and x_M is the location of the magnetometer. The magnetometer sensor measures 1 with high probability if the target is within r from the location of the magnetometer and the target is metallic as indicated by $\theta_{metal} = 1$. This sensing modality will be used to demonstrate the case when discriminatory information is delayed for resolving data association ambiguity.

3.2 Multiple target to multiple sensors

We now present a model of sensor measurements observed under the presence of multiple targets. We will exploit the framework of graphical models ([6], [5]) because the standard inference algorithm, belief propagation, provides an immediate algorithm for approximately computing posterior marginal distributions given sensor measurements. Furthermore, since belief propagation is a message passing algorithm involving local partial computations, the belief propagation algorithm can be naturally mapped to a distributed algorithm on a sensor network. Let us define some notation. X_t^i is the random variable of the location of target i at time t. θ_t^i is the random variable of other characteristics of target i at time t like nominal intensity, frequency signature, and whether the target is metallic. $Z_t^{i,j}$ is the measurement observed by sensor j from target i at time t which we call the *individual track contribution* of target i to sensor j. Z_t^j is the measurement observed by sensor j at time t, which is some combination of the individual track contributions.

We will demonstrate how to generate a graphical model from multiple targets to multiple sensors through the example of Figure 1. Targets are represented by gray dots labelled by letters, and sensor

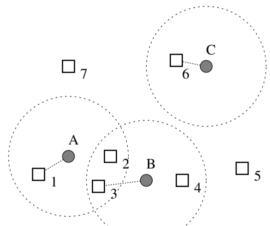

Figure 1: Multiple target/ multiple sensor example

nodes are represented by squares labelled by numbers. We assume that each target contributes to the measurement of sensors that are local to it, which is indicated by the dotted circles around each target location and can be derived from the sensing models.

Figure 2 shows the corresponding graphical model. The transition from $(X_{t-1}^i, \theta_{t-1}^i)$ to (X_t^i, θ_t^i) is given

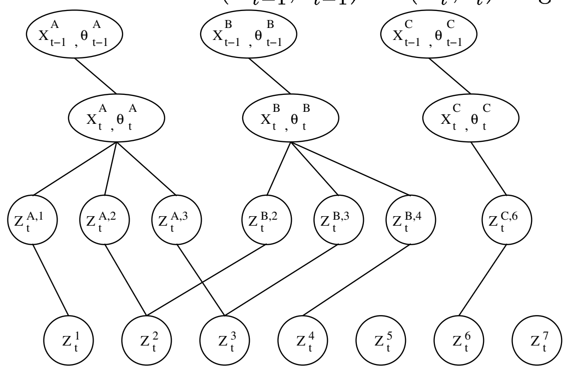

Figure 2: Graphical model

by a motion model of the dynamics of target i where we assume targets move independently. The transition from (X_t^i, θ_t^i) to $\{Z_t^{i,j}\}_j$ is given by the single target to single sensor measurement models as discussed in Section 3.1. The nodes at this level correspond to individual track contributions which serve as intermediate representations which will allow us to distribute the belief propagation algorithm. Finally, individual track contributions to a sensor node j are combined to form the measurement observation Z_t^j. In particular, for the intensity and frequency bandlimited intensity measurements, the combined measurement is the sum of the individual contributions, and for the magnetometer measurements, the combined measurement is the logical OR of the individual contributions. In general, the relationship between individual contributions $\{Z_t^{i,j}\}_i$ and the combined measurement Z_t^j can be specified by a transition density

$$p_{Z_t^j|\{Z_t^{i,j}\}_j}(z_t^j|\{z_t^{i,j}\}_j) \ . \tag{5}$$

Note that those sensor nodes which are not within range of any target play no part in the inference for any target (nodes 5 and 7). Some sensor nodes' measurements (2 and 3) depend on multiple targets, and others (1, 4, and 6) depend on only a single target.

4 Track Update Algorithm

One iteration of the multiple target tracking algorithm involves computing the filtered distribution of all pairs (X_t^i, θ_t^i) given the previous filtered distribution of $(X_{t-1}^i, \theta_{t-1}^i)$ and the current measurement observations $Z_t^j = z_t^j$. Due to the possible presence of loops in the graphical model shown in Figure 2, belief propagation will result in an approximate solution to the true posterior marginals.

The particular algorithm we use to update tracks is a two-pass algorithm on the graphical model. The first *top-down* pass starts from the prior distributions of the pairs $(X_{t-1}^i, \theta_{t-1}^i)$ and flows downward through the graphical model propagating higher level information down to the observables. The second *bottom-up* pass flows upwards incorporating the effect of observed measurements with higher level information. The result is a set of filtered distributions for (X_t^i, θ_t^i) for each target i which completes one iteration of the track update algorithm.

4.1 Top-down Pass

The top-down pass starts with distributions of higher nodes and computes a marginal distribution of all nodes connected below it. The links between higher nodes and lower nodes are associated with a transition density downwards. For example in Figure 2, the link between nodes $(X_{t-1}^i, \theta_{t-1}^i)$ and (X_t^i, θ_t^i) is associated with the motion model

$$p_{X_t^i,\theta_t^i|X_{t-1}^i,\theta_{t-1}^i}(x_t, \theta_t|x_{t-1}, \theta_{t-1}) \ . \tag{6}$$

The top-down pass computes the following "prior" distributions starting from the posterior distribution

$$\tilde{p}_{X_{t-1}^i,\theta_{t-1}^i}(x, \theta) \tag{7}$$

from the previous time $t-1$.

$$\begin{aligned} \hat{p}_{X_t^i,\theta_t^i}(x,\theta) \quad = & \int_{\mathcal{X}} \int_{\Theta} p_{X_t^i,\theta_t^i|X_{t-1}^i,\theta_{t-1}^i}(x,\theta|x',\theta') \\ & \cdot \tilde{p}_{X_{t-1}^i,\theta_{t-1}^i}(x',\theta') d\theta' dx' \end{aligned} \tag{8}$$

$$\hat{p}_{Z_t^{i,j}}(z) = \int_{\mathcal{X}} \int_{\Theta} p_{Z^j|X,\theta}(z|x,\theta') \hat{p}_{X_t^i,\theta_t^i}(x,\theta') d\theta' dx \tag{9}$$

for all targets i and sensors j where $\mathcal{X}$ is the space of possible target locations and Θ is the space of possible target classes.

4.2 Bottom-up Pass

The bottom-up pass uses the prior densities computed from the top-down pass and incorporates the measurements Z_t^j observed to compute posterior densities. Since there are many links going into each Z_t^j, the transition probability of these links is given in general form by Eqn. 5.

Given that the measurement observed by sensor j is z_t^j, the posterior joint density of $\{Z_t^{i,j}\}_i$ can be computed by Bayes' rule.

$$\tilde{p}_{\{Z_t^{i,j}\}_i}(\{z^{i,j}\}_i) = \alpha \cdot p_{Z^j|\{Z^{i,j}\}_i}(z_t^j|\{z^{i,j}\}_i) \cdot \prod_{i'} \hat{p}_{Z_t^{i',j}}(z^{i',j}) \quad (10)$$

where α is a normalization constant. By marginalizing the above density, we get a posterior marginal density for each $Z_t^{i,j}$

$$\tilde{p}_{Z_t^{i,j}}(z) \ . \quad (11)$$

Continuing up the graphical model (e.g., see Figure 2), we compute the posterior density of each (X_t^i, θ_t^i) approximately by treating the track contributions $\{Z_t^{i,j}\}_j$ as if they were independent.

$$\tilde{p}_{X_t^i,\theta_t^i}(x,\theta) = \prod_j \left[\int \frac{\tilde{p}_{Z_t^{i,j}}(z)}{\hat{p}_{Z_t^{i,j}}(z)} p_{Z_t^{i,j}|X_t^i,\theta_t^i}(z|x,\theta) dz \right] \cdot \hat{p}_{X_t^i,\theta_t^i}(x,\theta) \quad (12)$$

4.3 Distributed Algorithm

To distribute the top-down and bottom-up passes, the idea is to assign the sensor node which is closest to the position of the target to hold $\tilde{p}_{X_t^i,\theta_t^i}$, which we refer to as the *track holder*. In the presence of multiple targets, there exists a track holder for each target. It is possible for a single sensor node to be the track holder for multiple targets. The track holder initiates each iteration of the top-down and bottom-up pass. The graphical model as shown in Figure 2 must be implicitly constructed in order to perform the two passes.

The distributed algorithm begins with each track holder i knowing the previous track estimate $\tilde{p}_{X_{t-1}^i,\theta_{t-1}^i}$. In the following description, we will refer to the example in Figure 1 where the track holder for tracks A, B, and C are 1, 3, and 6 respectively as indicated by the dotted lines.

1. Each track holder computes $\hat{p}_{X_t^i,\theta_t^i}$ according to Eqn. 8.
2. The track holder determines which sensor nodes j are within range of the target and computes the individual prior track contributions $\{\hat{p}_{Z_t^{i,j}}\}_j$ according to Eqn. 9. Thus, node 1 computes prior contributions from track A to nodes 1, 2, and 3, node 3 from B to 2, 3, and 4, and node 6 from C to 6.
3. The track holder broadcasts these individual prior track contributions to the appropriate sensor nodes. That is, node 1 broadcasts the appropriate prior contributions to nodes 2 and 3, node 3 to 2 and 4, and node 6 does not broadcast any messages.
4. All sensor nodes now have prior track contributions from all tracks which are within range. This allows each sensor node to determine how many targets are contributing to its observed measurement z_t^j. For example, nodes 2 and 3 know that two targets are contributing to their measurement, nodes 1, 4, and 6 know only one target is contributing, and nodes 5 and 7 know that no target is contributing.
5. Each sensor node computes $\tilde{p}_{Z_t^{i,j}}$ for each prior track contribution received according to Eqn. 11 and sends these posterior track contributions back to the appropriate track holders. Nodes 2 and 3 send posterior contributions of A to 1, and nodes 2 and 4 send posteriors of B to 3.
6. All track holders can now compute $\tilde{p}_{X_t^i,\theta_t^i}$ from the received messages according to Eqn. 12.
7. The final step is to make the sensor node closest to the new estimated position of the target be the new track holder. Thus, each track holder determines if any other sensor node is closer than itself to the mean position of $\tilde{p}_{X_t^i,\theta_t^i}$, and if so, alerts the closest sensor node to become the new track holder and passes the posterior distribution to it.

The above distributed algorithm results in sensor nodes handing off track holder responsibility over time and keeps communications needed for updating the multiple tracks localized around target locations.

5 An architecture of distributed information integration

Although we have shown the general update equations for updating the target characteristics θ_t^i in Eqn. 8 and Eqn. 12, we will separate the classification and use of the information in θ_t^i depending on the particular configuration of targets and availability of sensing modalities which refines when certain computations should and should not be performed.

As mentioned in the introduction, we have used the general methodology proposed in [2] for structuring our distributed solution. A main part of the methodology is to organize information processes into a layered information processing architecture which, for multiple target tracking, include localization, tracking, and classification, layered as shown in Figure 3. Successive layers of processes give rise to successive layers of representation as indicated by the large gray rounded boxes. A process layer is decomposed into individual tasks

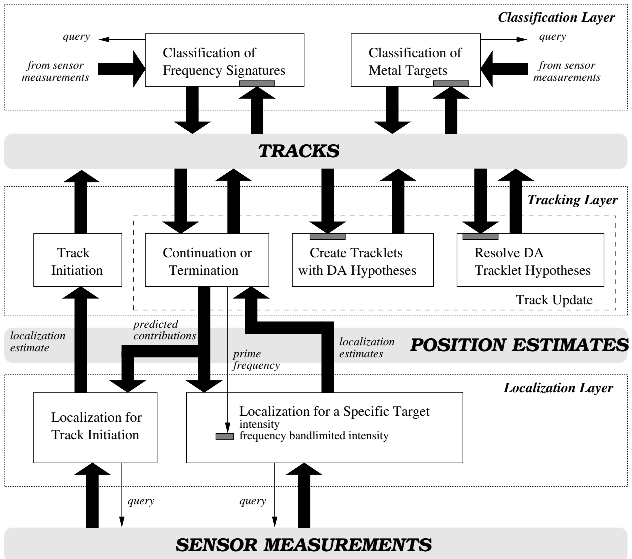

Figure 3: Layered hierarchical view of processes

which transform representations of one layer to another or modify representations in a single layer. These tasks can be labelled top-down or bottom-up depending on the direction of information flow as indicated by the black block arrows. From such a holistic view of the tasks and information flow, conditions for activating tasks (indicated by small gray boxes) and control interactions with data (thin black arrows) can be determined. By decomposing a global inference problem into layers of individual tasks, we gain modularity so that the construction of a distributed algorithm for the system amounts to distributing the implementation of each individual task.

5.1 Updating classification information

When the frequency signature $\underline{A}$ of a target is unspecified, track position estimates are good, and no other targets are nearby, we can compute the frequency signature of the track if there are enough microphones within range of the track. Assuming the frequency signature of any target is one of a set $\mathcal{A}$ of possible frequency signatures, the maximum likelihood estimate $\hat{\underline{A}}$ computed from n frequency bandlimited intensity measurements $\{\underline{z}_{F_i}\}_{i=1}^{n}$ is given by

$$\hat{\underline{A}} = \arg_{\underline{A}} \min \prod_{i=1}^{n} (\underline{z}_{F_i} - \underline{A} f(||\mu - x_{F_i}||^{\alpha}))^2 \qquad (13)$$

where x_{F_i} is the position of i^{th} microphone, μ is the mean of the current density of the target position, and all other parameters are as described in Eqn. 3.

Computing frequency signature information distributedly can be implemented by piggybacking a request for frequency data when the track holder broadcasts individual prior track contributions. If the receiving acoustic sensor node determines that only a single target is within range, then the node computes the frequency bandlimited intensity measurement and sends this information back to the track holder. The track holder can then estimate the frequency signature of the target if enough acoustic sensor nodes send frequency bandlimited intensity measurements.

Another classification task is computing whether a target is metallic when in range of a magnetometer sensor. This operation will be performed only when a single target is within range of a magnetometer for simplicity. The update of whether the target is metallic $\tilde{p}_{\theta_m}(\theta)$ given the previous distribution $\hat{p}_{\theta_m}(\theta)$ and a new measurement z_M is given by

$$\tilde{p}_{\theta_m}(\theta) = \begin{cases} \alpha(z_M(1-p) + (1-z_M)p)\hat{p}_{\theta_m}(1) & , \theta = 1 \\ \alpha((1-z_M)(1-p) + z_M p)\hat{p}_{\theta_m}(0) & , \theta = 0 \end{cases} \qquad (14)$$

where α is a normalization constant to make $\sum_{\theta=0}^{1} \tilde{p}_{\theta_m}(\theta) = 1$. Implementing the classification for whether the target is metallic can be accomplished by the track holder requesting data from any magnetometers within range. If the magnetometer determines that only a single target is within range, it responds to the track holder with its observed measurement. The track holder can then update the distribution of whether the target is metallic.

5.2 Virtual single target tracking

Virtual single target tracking is the case when a track is far from other tracks so that the update of this track is virtually the same as single target tracking. The graphical model corresponding to this scenario is given in Figure 4. Assuming that all targets have a known

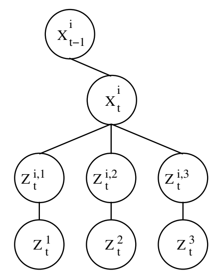

Figure 4: Model for virtual single target

intensity, classification information, in particular frequency signature information, is unneeded for updating track positions so that only intensity measurements are used to update tracks. The graphical model above indicates this by omitting the θ_t^i terms.

5.3 Multiple targets with immediate discriminatory information

Multiple target with immediate discriminatory information available is the scenario when several targets are close spatially and they each have a distinct frequency signature. The graphical model for two targets with distinct frequency signatures is shown in Figure 5. During the top-down pass, the frequency infor-

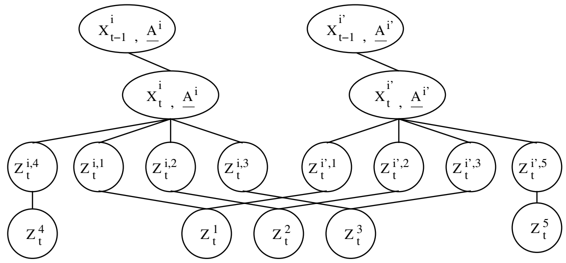

Figure 5: Model for multiple targets (immediate)

mation about the targets is used to compute the prior track contributions to frequency bandlimited intensity measurements of all acoustic sensor nodes within range. The bottom-up pass incorporates frequency bandlimited intensity measurements. The final update of Eqn. 12 updates only the position of the target while keeping the frequency signatures fixed since we have chosen to update classification information only in the scenario described in Section 5.1.

5.4 Multiple targets with no discriminatory information

When there are multiple targets with no known discriminatory information which are spatially close, the best we can do is to represent the ambiguity in some compact representation while continuing to track the two targets. After n targets converge, cross, and diverge again as in Figure 6, there exist n^2 possible track histories, since the track history after crossing can be associated with any of the n track histories before the crossing. Rather than store these n^2 possible

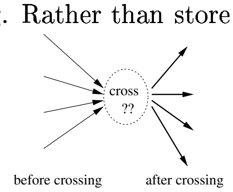

Figure 6: Representing unresolvable ambiguity

tracks, we will consider a factored representation which amounts to representing the track histories before and after crossing into separate tracklets. Then, there exists $2n$ total tracklets to represent. Furthermore, we must encode which tracklets can possibly correspond with the previous history of the track. Thus, each of the n tracklets after crossing stores that any of the n tracklets before crossing can be the previous history of the track.

To implement this distributedly in the sensor network, track holders must determine if there exist other tracks nearby and determine whether there exists any discriminatory information between them. If there is no discriminatory information, then each track holder terminates its own track, creates a new track in its place, and attaches information to the new track that its previous history can be one of the set of tracklets which have been terminated.

5.5 Multiple targets with delayed discriminatory information

The case when there exists discriminatory information between crossing targets but that information cannot be exploited until after targets have diverged again can be dealt with initially in the same manner as the case when there exists no discriminatory information. When targets cross, the old tracks are terminated and new tracklets are instantiated attached with information about the possible previous track histories, including classification information about each of the possibilities. The new tracklets are instantiated with no classification information. After the targets have diverged again, the new tracklets can begin to collect classification information (as described in Section 5.1), and when there is enough information to warrant a decision about the type of target, the ambiguity of associating the tracklets with the old track histories can be resolved.

6 Simulations and Discussion

We present simulations demonstrating the distributed track update algorithm for multiple target tracking under various cases of availability of discriminatory information. Particle filters [4] are employed to represent, propagate, and update the various probability densities in the algorithm presented in Section 4. Details of the particle filter and implementation are omitted in this paper but can be found in [2].

In the figures, squares represent microphone sensors, and triangles represent magnetometer sensors. The circles around each triangle is the effective range of the magnetometers. The spatial distribution of tracks are represented by clouds of particles, and the track holder is indicated by a line from the mean of the cloud of particles to the sensor node. The circles around track contributions indicate the range of sensor node positions that receive a substantial track contribution.

Figure 7 shows snapshots of two simulations where two targets with distinct frequency signatures either cross or converge and diverge. For both simulations, Figure 7a shows the two tracks collecting classification information indicated by a diamond on the track history. In Figure 7b, the tracks are detected to be close enough for the possibility of data association ambiguity so that frequency signature information is used to update the tracks, which is indicated by circles around the track history. For the simulation when targets cross, Figure 7c shows the resulting track history of this successful multiple target tracking run. Figure 7d shows

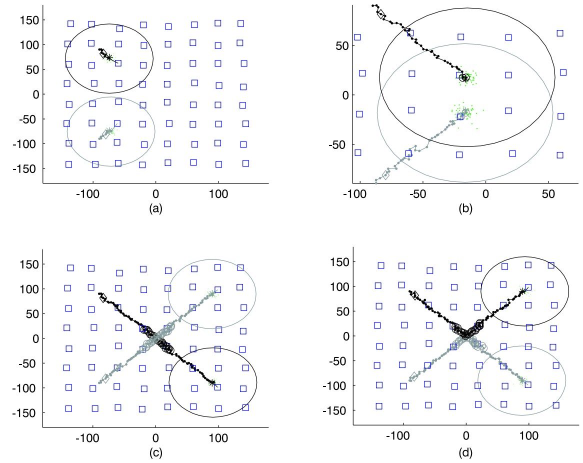

Figure 7: Data association (immediate)

the same situation as Figure 7c except that the latter half of each target trajectory is switched. The resulting tracks show the targets converging and diverging.

Figure 8 shows the case with delayed discriminatory information which also demonstrates the case with no discriminatory information. The two targets have the same frequency signature, but the top target is metallic and the bottom one is not. As the two targets pass

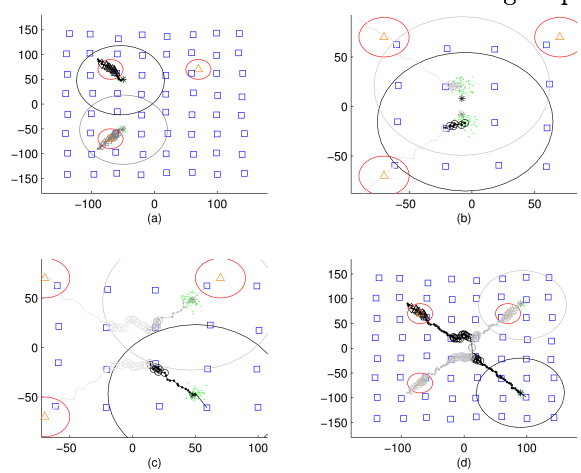

Figure 8: Data association (delayed)

through the magnetometer sensors in Figure 8a, they each collect information about whether their target is metallic or not. When the two targets are close enough for an ambiguity in data association to occur as in Figure 8b, the old tracks are terminated (light gray), and new ones are instantiated with the appropriate possible associations. After diverging, the gray track collects discriminatory information by passing through another magnetometer and repairs itself. Furthermore, the gray track holder broadcasts a message to the black track holder indicating it may no longer be associated with the track which originated from the bottom left. The result is shown in Figure 8d.

Note the error in the track positions when the targets cross each other. This error is caused by the limits of the belief propagation approximation to the true marginals and the fact that only marginal distributions, rather than joint distributions, of track locations are represented. For better tracking accuracy, converting to a joint representation may be necessary.

The presented scheme for repairing tracklets are hard decisions once enough information has been collected. A “softer” version would be to update the probability of associating tracklets before and after crossing as in [8] and would be a generalization of our scheme. The broadcast message to repair the other track could be a flooding protocol in its most naive form since the position of the other track is unknown; however, schemes for maintaining the position of the other track to avoid a flooding protocol is a topic of future research.

References

[1] Y. Bar-Shalom, *Extension of the probabilistic data association filter in multi-target tracking*, Proc. 5th Symp. Nonlinear Estimation, pp. 16–21, Sep 1974.

[2] M. Chu, *A Hierarchical Framework for Constructing Computationally Efficient Algorithms for Distributed Inference Problems*, EECS Dept., Massachusetts Institute of Technology, 2003.

[3] M. Chu, H. Haussecker, and F. Zhao, *Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks*, Int’l Journal for High Performance Computing Applications, Vol. 16, No. 3, Aug 2002.

[4] M. Isard and A. Blake, *CONDENSATION – conditional density propagation for visual tracking*, Int’l Journal for Computer Vision, Vol. 29, No. 1, pp. 5–28, 1998.

[5] F. Kschischang, B. Frey, and H. Loeliger, *Factor graphs and the sum-product algorithm*, IEEE Trans. Information Theory, Vol. 47, pp. 498–519, Feb 2001.

[6] J. Pearl, *Probabilistic reasoning in intelligent systems*, Morgan-Kaufmann, 1988.

[7] D. B. Reid, *An algorithm for tracking multiple targets*, IEEE Trans. on Automatic Control, Vol. 24, No. 6, pp. 843-854, Dec 1979.

[8] J. Shin, L. Guibas, and F. Zhao, *A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks*, The 2nd Int’l Workshop on Information Processing in Sensor Networks, Apr 2003.

[9] F. Zhao, J. Shin, and J. Reich, *Information-driven dynamic sensor collaboration for tracking applications*, IEEE Signal Processing Magazine, Vol. 19, No. 2, pp. 61–72, Mar 2002.