

UNCLASSIFIED

Defense Technical Information Center
Compilation Part Notice

ADP021694

TITLE: Distributed Tracking in Wireless Ad Hoc Sensor Networks

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Proceedings of the International Conference on International Fusion
[6th]. Held in Cairns, Queensland, Australia on 8-11 July 2003. Volume 1:
FUSION 2003

To order the complete compilation report, use: ADA442007

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP021634 thru ADP021736

UNCLASSIFIED

Distributed Tracking in Wireless Ad Hoc Sensor Networks

Chee-Yee Chong
Booz Allen Hamilton
San Francisco, CA, U.S.A.
chong_chee@bah.com
cychong@ieee.org

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

Shozo Mori
Information Extraction &
Transport (IET)
Arlington, VA, U.S.A.
smori@iet.com
smori@ieee.org

Sri Kumar
Defense Advanced
Research Projects Agency
(DARPA)
Arlington, VA, U.S.A.
skumar@darpa.mil

Abstract – *Target tracking is an important application for wireless ad hoc sensor networks. Because of the energy and communication constraints imposed by the size of the sensors, the processing has to be distributed over the sensor nodes. This paper discusses issues associated with distributed multiple target tracking for ad hoc sensor networks and examines the applicability of tracking algorithms developed for traditional networks of large sensors. When data association is not an issue, the standard predict/update structure in single target tracking can be used to assign individual tracks to the sensor nodes based on their locations. Track ownership will have to be carefully migrated, using for example information driven sensor tasking, to minimize the need for communication when targets move. When data association is needed in tracking multiple interacting targets, clusters of tracks should be assigned to groups of collaborating nodes. Some recent examples of this type of distributed processing are given.*

Keywords: Wireless ad hoc sensor networks, multiple target tracking, distributed tracking

1 Introduction

Recent advances in micro electromechanical systems (MEMS) and wireless technologies have resulted in inexpensive micro-sensors with embedded processing and communication capabilities [1]. A sensor network consisting of many micro-sensors communicating with each other over wireless links can be deployed rapidly in an area of interest and used in applications ranging from environmental monitoring to battlefield surveillance. In this paper we focus on target tracking with such ad hoc sensor networks.

The availability of inexpensive sensors means that they can be deployed much closer to the targets of interest for detection and tracking. Collaboration among the sensors can result in accuracy not achievable with individual sensors. At the same time, energy constraints due to the size of the sensors impose special processing requirements that are different from tracking with other traditional types of sensors such as ground-based or airborne radars. In particular, communication requires much more energy than computation and thus the processing has to be distributed,

putation and thus the processing has to be distributed, with nodes communicating processed data instead of sensor data.

Distributed tracking for sensor networks was first investigated in the early 1980's to understand how sensors with distributed but overlapping coverage can be used to detect and track targets. Since then there has been much research on distributed tracking [2,3]. However, energy constrained processing and communication has not been a main concern and the research emphasis has been on association and fusion of tracks from multiple processing nodes.

Distributed tracking in sensor networks is a multidisciplinary area that involves hardware, signal processing, estimation and inference algorithms, and computer science. Researchers in computer science and signal processing have undertaken most of the recent work [4,5]. While feasibility has been demonstrated for simple scenarios in simulated and real test beds, the development of algorithms for more difficult and realistic tracking scenarios is just beginning. This paper starts with the standard tracking algorithms and investigates how processing can be mapped to a sensor network with energy and communication constraints. We hope to demonstrate that high performance tracking algorithms can be implemented in such ad hoc sensor network.

The rest of this paper is structured as follows. Section 2 provides a historical perspective on tracking for large sensor networks, discusses the challenges faced by tracking in small sensor networks and the state-of-the-art. Section 3 presents algorithms for single target tracking and how the processing can be distributed over a sensor network. Information driven sensor query is given as an example. In Section 4 we discuss multiple target tracking algorithms and the implication of data association on distributed processing. An example of group management for distributed track initiation and maintenance is given. Section 5 presents some issues for future research and concluding remarks.

2 Tracking in Sensor Networks

The objective of single target tracking is to generate an accurate estimate of the target position. When multiple targets are present, tracking determines the number of targets and estimates the position and velocity of each target. In some applications, it is also important to maintain the continuity of each track, e.g., to determine the identity of the target and its origin.

Multiple sensors are needed when a single sensor does not provide enough spatial coverage. Furthermore, multiple sensors exploiting different phenomenology can provide complementary information to improve tracking accuracy. Thus there is significant advantage in using a distributed network of sensors for target tracking.

The measurements from a sensor network can be processed by different architectures [6]. In a centralized architecture, all the measurements are sent to a central site for processing. This architecture is theoretically optimal since the central site has access to all information. However, significant bandwidth is needed for communication and the central site is a single point of failure. In a distributed architecture, there are multiple processing sites, each responsible for a number of sensors. These sites process their local measurements and communicate results with other sites. The distributed architecture does not provide optimal performance, but requires less communication and is more robust since there is no single point of failure.

The main issues of distributed tracking are: how the processing should be distributed; how the sensors should be controlled to balance performance versus resource utilization; what should be communicated between the processing nodes; and when communication should take place. After communication, the issue is how to fuse the incoming data with the local data.

2.1 Small Number of Large Sensors

Until recently, most distributed tracking systems consist of large sensor nodes (Fig. 1) such as airborne radars used in military systems. These sensors have fairly large fields of view so that each sensor can detect and observe many targets. They may also have overlapping coverage so that multiple sensors can observe the same targets. In a hierarchical tracking architecture, multiple processing nodes controlling different sensors may form their own tracks. Thus two main processing functions in a distributed tracking system are associating the tracks from different processing nodes and fusing the state estimates of associated tracks [7]. Since these estimates may have common information due to past communication, the fusion algorithm has to recognize this correlation as well as that due to common process noise in the target dynamics.

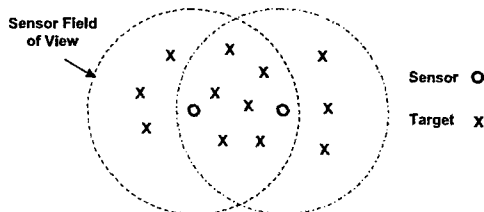


Figure 1: Large Sensor Networks

Research on distributed tracking for such networks began with the DARPA Distributed Sensor Networks (DSN) program in the early 1980's. Although the original vision of

the program was very ambitious, with large numbers of small inexpensive sensors interconnected by wireless communication, the technology at that time was not quite ready. Thus, DSN research was focused more on fixed networks of a small number of large sensors without energy constraints. In particular, the main test bed for distributed tracking consists of acoustic sensor arrays that are 6 meters across and powered by generators. A history of sensor network research evolution and challenges can be found in [8].

A generic distributed tracking architecture was proposed in [2] and each processing node (Fig. 2) consists of the following components:

- *Local processing* that performs multiple hypothesis tracking (MHT) with the local sensor data
- *Information fusion* that associates the tracks in the incoming hypotheses with the tracks in the local hypotheses and updates the state estimate of each track
- *Information distribution* that decides when, what and to whom to communicate by considering the local information content and information needs of the remote nodes

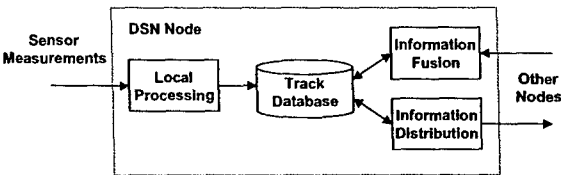


Figure 2: Structure of DSN Node

The use of multiple hypotheses in both local processing and information fusion assumes that the processing nodes are quite powerful and that communication is not an issue. Since communication can be quite complicated in general, the information to be fused may contain common information from previous communication. The concept of information graph was introduced to keep track of the communication history so that common information can be identified and removed.

This distributed tracking concept (without using MHT) was demonstrated in a test bed developed by M.I.T. Lincoln Laboratory [9,10] to track low-flying aircraft using multiple acoustic arrays. Since then much research has been performed on the problem of track association and track fusion when the states estimates of the tracks are not conditionally independent [11]. The emphasis has been more on achieving optimal performance than addressing communication or energy constraints.

2.2 Wireless Ad Hoc Sensor Networks

Advances in sensing, processing and computing hardware over the past few years have made wireless micro-sensor network envisioned in the DSN program a reality. A typical sensor node has a microprocessor and a limited amount of memory for signal processing and task scheduling. Its sensors may include acoustic microphone arrays,

video or still cameras, infrared, seismic, or magnetic sensing devices. Each sensor node communicates wireless with a small number of neighboring nodes within the communication range. Current wireless sensor hardware ranges from the shoebox sized Sensoria WINS sensors [12] to the matchbox sized Berkeley motes [13].

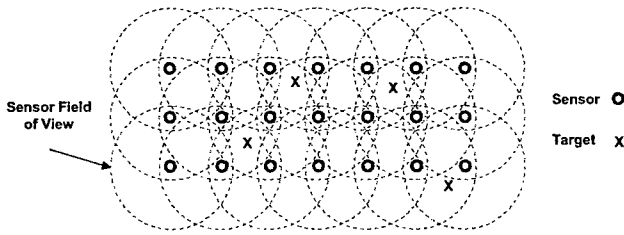


Figure 3: Dense Sensor Networks

Because of the low cost, many of these sensors can be deployed rapidly in an *ad hoc* manner to form a dense network in the area of interest. At the same time, the limited sensing range, battery, processing power, and bandwidth also create many challenging research issues. These include: network discovery so that nodes can communicate with each other; network control and routing, collaborative signal and information processing, tasking and query, and security [8].

The amount of energy used in wireless communication is particularly relevant in designing tracking architectures for wireless sensor networks. Since wireless communication dominates the energy consumption in embedded networked systems, it is important to minimize the amount and range of communication as much as possible through local collaboration and data compression, communicating only when necessary. Further energy savings can be obtained through sensor management. Even though each sensor may contribute additional information to tracking, the information contribution of each measurement should be balanced against its resource utilization. This is especially crucial in dense networks, where measurements are highly redundant and not all measurements are needed when energy is limited.

The recently concluded DARPA Sensor Information Technology (SensIT) program [14] addressed many of these issues. In particular tracking algorithms have been developed under the name of "Collaborative Signal and Information Processing" [15]. These algorithms focus on how to distribute the processing over the sensor nodes to minimize communication and how to select sensors to reduce energy consumption. With proper sensor tasking and control, at any given time, a single sensor node is responsible for tracking a particular target and the responsibility may migrate from sensor to sensor. Thus, a target is tracked by only one sensor, and there is no need for track association or fusion. This is quite different from distributed tracking for networks with a small number of large sensors where a target may be tracked by multiple sensors.

Tracking in wireless ad hoc sensor networks is intertwined with other issues such as routing protocols and signal processing. So far, most of the research has emphasized simple algorithms and feasibility demonstration in uncomplicated scenarios. To address more difficult and realistic scenarios, much additional research is needed. The remaining sections of this paper will show how tracking algorithms developed for more complicated scenarios can be adapted to wireless ad hoc sensor networks.

3 Tracking Single Target

Our approach to distributed tracking is to show that some fairly standard tracking algorithms may be applied to wireless ad hoc sensor networks. This section will focus on single target tracking. Section 4 will address algorithms for tracking multiple targets when data association is needed.

3.1 Model

We assume that either only one target is present in the network or there are multiple widely separated targets so that data association is not needed. Let $x(t)$ be the target state (position, velocity, etc.) at time t . The target state dynamics model is given by the state transition probability $P(x(t + \Delta t) | x(t))$.

Let $z(t)$ be the measurement at time t . The sensor measurement model is also given by $P(z(t) | x(t))$, the target-state-to-measurement transition probability. In general a sensor may not always detect a target, implying a detection probability that is strictly less than 1. A measurement on the target is a triple (z, t_k, s_k) , of measurement z obtained at time t_k by sensor s_k . When a sensor s_k observes at time t_k and does not generate a measurement, that information is represented by a triple (θ, t_k, s_k) , where θ is the symbol for nothing. At any given time, the *cumulative measurement set* Z_k includes all measurements collected up to t_k .

Any measurement generated by a sensor may not originate from the target being tracked. Such a measurement is referred to as a "false alarm". In the current section, we assume that there are no false alarms in the measurements.

3.2 Basic Algorithm

The basic algorithm for single target tracking consists of the following two steps: prediction and update. Assume that a measurement $z(t_k)$ or $(z(t_k), t_k, s_k)$ is generated at time t_k .

Prediction. The previous estimate $P(x(t_{k-1}) | Z_{k-1})$ is predicted to the current time by

$$P(x(t_k) | Z_{k-1}) = \int P(x(t_k) | x(t_{k-1})) P(x(t_{k-1}) | Z_{k-1}) dx(t_{k-1}) \quad (1)$$

Update. The current measurement is used to update the predicted probability by

$$P(x(t_k)|Z_k) = C^{-1}P(z(t_k)|x(t_k))P(x(t_k)|Z_{k-1}) \quad (2)$$

where C is a normalization constant.

When the state transition and measurement equations are linear and Gaussian, these two steps become the standard Kalman filter and the probabilities represented by the means and covariances. For the general nonlinear problem, equations (1) and (2) can be implemented by linearization techniques such as extended Kalman filter or unscented Kalman filter, discretization of state space or Monte Carlo methods such as particle filters. When multiple targets are present, these two steps have to be performed for each target plus data association to be discussed later. In the rest of this paper, we may use target state estimate to mean conditional target state distribution.

3.3 Distributed Single Target Tracking

As discussed before, communication may consume far more energy than computation or sensing. The prediction step in single target tracking is basically a local operation and does not require communication. On the other hand, the update step requires acquiring a sensor measurement, thus involving communication unless processing takes place at the sensor node.

There are two ways of reducing the communication cost in the update step. The first is to reduce the number of sensor measurements. In a dense sensor network, many measurements are redundant and some may not provide useful information. This will also cut down on the energy used in sensing. The second way is to have the track processing follow the motion of the target as given by the state estimate. Then the sensors generating the measurements will not be too far from the processing node, making multi-hop communication unnecessary. Figure 4 shows an approach for energy-efficient distributed single target tracking by moving the track processing with the target position.

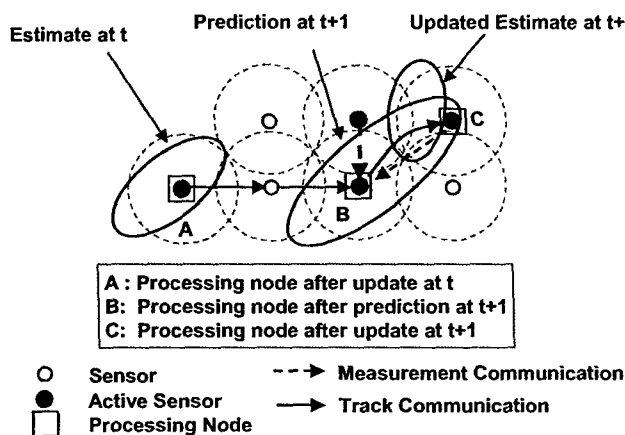


Figure 4: Migrating Track Processing with Target

Initialization. One or more sensors detect the target for the first time when it begins to move (e.g., generating acoustic signal). The first sensor to detect the target becomes the processing node and initializes the track with a state estimate $P(x(t_0)|Z(t_0))$. As the owner of this track, it suppresses all other neighboring sensors that can sense this target from making more measurements. These sensor nodes can then go back to sleep, thus conserving their batteries.

Prediction. The processing node for this track predicts the target state estimate $P(x(t_k)|Z_{k-1})$ at time t_k when the next measurement will be taken using equation (1), the local estimate and the target motion model. The sensor node whose coverage has the most overlap with the predicted position is then chosen to be the processing node and the track state distribution is moved to this node, thus incurring some communication cost.

Sensing. The processing node selects a set of sensors to generate the next measurements and tells the other sensors to stay dormant. Communication is needed to suppress the other nodes. In the extreme case, only the processing node is allowed to sense.

Update. The observed measurements are sent to the processing node and used to update the track state estimate. This track is assigned to the sensor node whose coverage has the most overlap with the updated state estimate and the steps are repeated.

In the above steps, only relevant sensors are active while the rest are suppressed. The active and suppressed nodes are dynamically updated as the target moves in the sensor field. This is to be accomplished via a group management mechanism, to be discussed in Section 4.4.

As a result of the group management, at any particular time the track state is maintained by only one node and thus there is no need to consider track fusion and how to deal with dependent information. Essentially, the processing for each track is centralized but the processing node follows the target to minimize communication. This concept is also discussed in [15, 16].

3.4 Sensor Selection

When prediction indicates that multiple sensors can observe the target, information and energy considerations require that a single sensor be selected. The *Information Driven Sensor Query (IDSQ)* approach [17] selects sensors based on information utility and resource constraints, thereby efficiently managing the scarce communication and processing resources while satisfying the information needs. This information-driven approach to sensor querying and data routing builds upon data centric routing and storage services such as geographic routing or directed diffusion routing [18].

The main idea of IDSQ is to base sensor collaboration decisions on information content as well as constraints on resource consumption, latency and other costs. In Fig. 5, the current leader node can task sensor a or b , given the current

uncertainty about the target position shown as the ellipse at time t . Assuming each sensor provides a distance constraint on the location of the target, then sensor b provides a larger reduction in the uncertainty along the longer axis of the ellipse, and hence is more desirable in terms of information utility. Since communication is the main cost component, sensors in a network must exploit the information content of the data already received to optimize the utilities of future sensing actions such as that by sensor b .

In general, IDSQ formulates the sensor tasking problem as a distributed constrained optimization problem that maximizes the information gain of sensors while minimizing communication and resource usage. The information gain depends on the specific sensor and has to be computed before the measurements are actually taken. For linear Gaussian models, the information gain from using a particular sensor depends on the prediction error covariance, which can be computed from the Kalman filter covariance propagation equation and is independent of the actual measurements. For other nonlinear models, the expected information gain over all possible measurements has to be used in the optimization. Several methods for computing the information gains are discussed in [17], and experimented in simulations and on field data from sensor network test beds.

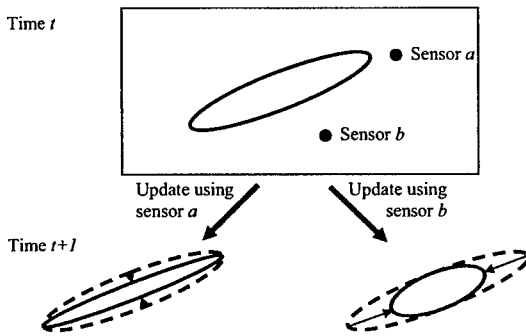


Figure 5: Sensor tasking based on information utility

4 Tracking Multiple Targets

Tracking multiple targets that are far apart can be treated as single target tracking since there is little interaction among the targets. The tracks for different targets are processed in parallel by different sensor nodes. When the targets are close to each other relative to the measurement error, the tracking of one target cannot be treated independently from those of the other targets. In particular, measurements have to be associated to the different targets or false alarms. Allocating the processing to the sensor nodes will be more involved.

4.1 Model

We assume that there are multiple targets $i = 1, \dots, N$. Each target i has a state $x_i(t)$ at time t . The state transition for each target is independent of that of another target and given by the conditional probability $P(x_i(t_{k+1}) | x_i(t_k))$.

The measurement model for each sensor consists of:

- Target wise independent state-dependent detection probability
- Target wise independent conditional probability $P(z_{ij}(t) | x_i(t))$ for the measurement $z_{ij}(t)$ if target i is detected by sensor j
- Density of false alarms as a function of the threshold setting

Each set $z_j(t_k)$ of measurements collected at time t_k by sensor s_j , is called a *measurement set* and contains measurements from targets as well as false alarms. To simplify the notation, we denote a measurement set by z_k . Each measurement in a measurement set z_k is uniquely identified by its index (h, k) , i.e., the h -th measurement in the measurement set z_k . A collection of measurement sets is called an *information set* and represents the information available to the tracker. The multiple target tracking problem is to determine the number of targets from the information set and to estimate the state (position and velocity) of each target track. A more precise formulation of the problem can be found in [19].

4.2 Basic Algorithm

Multiple target tracking is a difficult problem and many algorithms have been developed over the years [20, 21]. An overview of tracking algorithms for ground targets can be found in [22]. The main difficulty of multiple target tracking is data association. A measurement may originate from a false alarm, a target that has been detected before or a new target. Furthermore, a target may not be detected by a sensor at a given time. Thus, data association is a key component in any multi-target tracking algorithms.

Association algorithms may be simple or sophisticated. Simple algorithms (e.g., nearest neighbor) may associate measurements to individual targets independent of the association decisions of other tracks, while sophisticated multiple hypothesis algorithms consider the coordinated association of measurements to tracks over multiple frames of data. We now describe the basics of a multiple hypothesis tracking algorithm [19, 23].

A set of measurement indices (h, k) in an information set is a *track* when it is hypothesized to originate from a single track. A set of tracks is called a *data-to-data association hypothesis*, or simply a *(scene) hypothesis* when it is a consistent set of hypothesized detected targets. The system state of any MHT algorithm is a collection of hypotheses and tracks with the posterior probability for each hypothesis and target state distribution for each track. Moreover, the collection of hypotheses formed and evaluated on any given information set can be decomposed into stochastically independent components, each called a *stochastic cluster* or simply a *cluster*. The measurement-oriented MHT algorithm first developed in [23] and generalized in [19] consists of the following steps shown in Fig. 6:

Time alignment (prediction). The state estimates of the tracks in each cluster are predicted to the time of the measurement set.

Gating. The candidate track and measurement pairs that can possibly be associated are computed using gates that reflect state estimate and measurement uncertainty.

Track-to-measurement likelihood calculation. For each pair of track and measurement that can be associated, the likelihood of association is computed.

Cluster merging. Based on this likelihood calculation, some of the clusters are merged into a larger cluster to account for cross-cluster correlation.

Hypothesis and track evaluation. For each cluster, a set of hypotheses is formed by assigning each measurement either to an old track (targets already detected), a new track (a newly detected target), or a false alarm. The probability of each hypothesis is then evaluated by first multiplying the appropriate track-to-measurement likelihoods followed by normalization. Each track is then evaluated by updating its target state distribution by a newly assigned measurement.

Hypothesis pruning and combining. The size of each updated cluster is reduced by pruning weak hypotheses and combining similar hypotheses

Cluster splitting. Whenever possible, a cluster will be split into sub clusters, corresponding to separating targets.

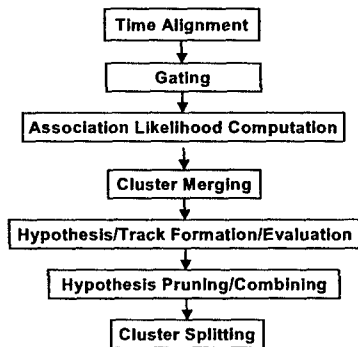


Figure 6: Structure of Multiple Hypothesis Tracking

4.3 Distributed Multiple Target Tracking

We assume that computation is less an issue than communication, and the main objective is minimizing the communication among the nodes. As before, predicting the future state of a track is a local operation and does not require any communication. However, association and update require communicating the sensor measurements to the processing nodes.

As in single target tracking, the track state estimate and track processing should follow the movement of the target. However, since multiple track association hypotheses may be associated with a single target, they cannot be processed independently. In addition, tracks corresponding to targets that come close to each other are dependent and should be processed together. Thus a natural unit for distributed processing is the *cluster*.

The cluster, as defined here, refers to the logical data structure of the tracks and hypotheses. However, in sensor networks, a cluster is often meant to be a group of nodes. The hard problem for distributed tracking is to map the logical clusters of tracks to the physical clusters of sensor nodes, as we will discuss below and in Sec. 4.4.

The components in a cluster are distributed to the sensor nodes in a two level structure:

- Individual tracks are assigned to sensor nodes based on the closeness of the sensor nodes to the center of the track state distributions. Some nodes may be responsible for multiple tracks.
- The sensor nodes within a cluster elect a node to be responsible for the tracks and hypotheses in the cluster. Some load balancing may be needed to make sure that the “leader” node for the cluster or cluster processing node is not overloaded.

The following operations are performed at each sensor observation time t_k .

For each track, the operations are similar to those in single target tracking.

Prediction. The processing node for the track predicts the track state estimate, and assigns the track to the sensor closest to the center of the target state distribution.

Gating. Based on the current state estimate and sensor measurement uncertainty, the track processor or owner determines the sensors that should collect measurements.

Sensing. The active sensors return with one or more measurements to the owner of the track.

Track measurement association likelihood computation. The track owner computes the track/measurement association likelihood(s).

The cluster processing node determines whether it should merge its cluster with another cluster by examining if these two clusters have tracks that can be associated with the same measurement. When two clusters merge, one of the cluster processors takes over the responsibility of the merged cluster and the tracks may be reassigned to the track processing nodes for load balancing. Within this (merged) cluster, the cluster processor performs association using an assignment algorithm (single frame association) or multiple hypothesis processing (multiple frame association).

Each track then updates its state estimate using the associated measurement. The cluster is split into multiple clusters and assigned to multiple nodes if possible. The process then repeats at another sensor observation time. Figure 7 illustrates how clusters merge and split for two targets merging and separating.

This processing architecture is centralized at the cluster level but the group of nodes responsible for processing each cluster migrates with the cluster. Multiple clusters can be processed independently except when they merge. Within each cluster, the tracks are assigned to nodes that move with the tracks.

Multiple hypothesis processing requires significant computing power at each processing node. When this is not

possible, a single hypothesis approach can be used. In some highly confusing situation, e.g., when two targets are very close to each other, it may make sense to maintain a single group track instead of individual target tracks since there is insufficient information to make good association decisions.

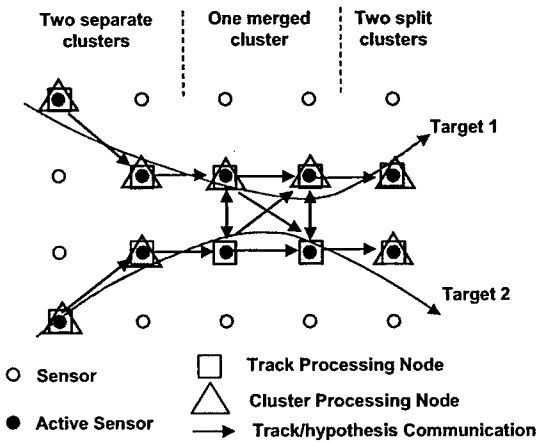


Figure 7: Cluster Migration in Multi-Target Tracking

4.4 Group Management for Track Initiation and Maintenance

The distributed group management approach for track initiation and maintenance presented in [24] is one way to map the logical tracks to physical sensors for both single or multiple target tracking problems. A collaborative group is a set of sensor nodes that are responsible for the initiation and maintenance of a single track. Effectively, these are sensors whose coverage overlaps with the state estimate of the track.

Track Initiation.

When the target enters the sensor field or emits signal for the first time, it is detected by a set of sensor nodes.

1. Each individual sensor performs local detection using a likelihood ratio test.
2. Nodes with detections form a collaborative group and select a single leader (e.g., based on time of detection). While we discuss the single leader approach, it is also possible that a small number of nodes are elected to share the leadership.
3. Leader initiates a track and assigns a track identity using detection time. It uses the uncertainty in track position estimate and maximum detection range to calculate a *suppression* region and informs all group members in the *suppression* region to stop detection. This reduces energy consumption of the other nodes and avoids further track initiation. The assumption is that there is only one target in the neighborhood.
4. As the target moves, sensors are selected to make measurements, using for example the IDSQ algorithm. These measurements will be used to update the *suppres-*

sion regions and *unsuppression* regions, and the group membership of nodes will change.

The suppression or collaborative region for a track contains the set of sensors that can potentially update the track. An example of collaborative groups maintained for two targets is shown in Fig. 8. With only a single target present, sensors other than the leader or leaders in the group may not provide much additional information to tracking and can cause confusion by initiating new tracks. This single target assumption may hold if the target density is low or the sensors have short range.

Fig. 8. shows distributed group management for multiple target tracking on a 17-node sensor network test bed at PARC. The lower figure shows the physical sensor lay down and communication topology. The upper figure shows how two distinct tracks are maintained by separate groups of collaborating sensors, as enclosed by the rectangular bounding boxes.

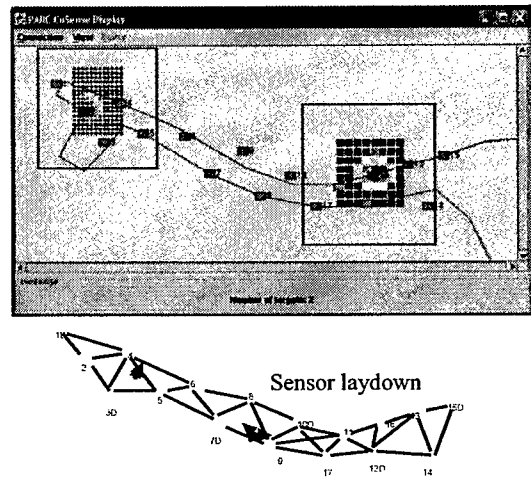


Figure 8: Distributed Track Management: Experiment on PARC Sensor Network Test Bed.

Track Maintenance.

When the targets are far apart, their tracks are handled by multiple collaborative groups working in parallel. When targets cross, the position uncertainty regions for their tracks overlap and the collaborative groups for these tracks are no longer distinct. This can be detected when a leader node receives a suppression message from with different (track) ID from its own. When two groups collide, the sensor measurements in the overlapping region can now be associated to either one of the two tracks.

Data association algorithms such as optimal assignment or multiple hypothesis processing can be used to resolve this ambiguity. In reference [24], a simple track merging approach is proposed to keep the older track and drop the younger track. The two collaborative groups then merge into a single group.

This approach works well if the two tracks were initiated from a single target. When the two tracks result from two targets, the merging operation will temporarily track the two

targets as one. When they separate again, a new track corresponding to one of the targets will be re-initiated. However the identities of the targets will be lost.

Using an identity management algorithm, the ambiguities in the target identities after crossing tracks can be resolved using additional local evidence of the track identity and then propagate the information to other relevant tracks. Details of distributed identity management can be found in [25].

5 Conclusions

Wireless ad hoc sensor networks have great potential in target tracking applications since the sensors can be deployed close to the targets of interest. At the same time, energy and bandwidth constraints require that the processing be distributed over the sensor nodes. We reviewed the evolution of tracking in sensor networks and showed how the structure of standard tracking algorithms can be used to distribute the processing. In particular, processing should follow the movement of the target(s) over the sensor field. This is fairly straightforward in single target tracking but distributing the data association functions in multiple target tracking is more complicated. We gave some examples of recent tracking work to illustrate these concepts.

Research in distributed tracking in wireless ad hoc sensor networks is just beginning. In order to address more complicated scenarios, many research issues have to be addressed. Examples include what kind of prior information is needed for each node and how to provide this information. Another issue is how to handle out-of-sequence measurements due to communication delays. In addition, data association is a challenge and doing it efficiently over a network is even more difficult. When the same target is detected and processed by multiple sensor nodes we also have to associate the tracks and make sure that information is not double-counted.

References

- [1] Special Issue on Sensor Networks and Applications, *Proc. IEEE*, Aug. 2003.
- [2] C. Y. Chong, S. Mori, and K. C. Chang, "Distributed Multitarget Multisensor Tracking," *Multitarget Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., pp. 247 - 295, Artech House, 1990.
- [3] M. E. Liggins II, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed Fusion Architectures and Algorithms for Target Tracking," *Proc. IEEE*, vol. 85, no. 1, pp. 95 - 107, Jan., 1997.
- [4] Special Issue on Sensor Networks, *Int. Journal. High Performance Comp. Applications*, vol. 16, no. 3, Fall 2002.
- [5] Special Issue on Sensor Networks, *IEEE Signal Processing Magazine*, vol. 19, no. 3, Mar. 2002.
- [6] C. Y. Chong, "Distributed Architectures for Data Fusion," *Proc. Int. Conf. on Multisource - Multisensor Information Fusion*, Las Vegas, NV, July 1998.
- [7] C. Y. Chong, S. Mori, K. C. Chang, and W. H. Barker, "Architectures and Algorithms for Track Association and Fusion," *Proc. 2nd Int. Conf. on Information Fusion*, Sunnyvale, CA, July 1999.
- [8] C. Y. Chong and S. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proc. IEEE*, Aug. 2003 (to appear).
- [9] R. T. Lacoss, "Distributed Mixed Sensor Aircraft Tracking," *Proc. American Control Conf.*, Minneapolis, MN, Jun., 10-12, 1987.
- [10] "Distributed sensor networks," Final Report, M.I.T. Lincoln Laboratory, Sep. 1986.
- [11] C. Y. Chong and S. Mori, "Convex Combination and Covariance Intersection Algorithms in Distributed Fusion," *Proc. 4th Int. Conf. on Information Fusion*, Montreal, Canada, Aug. 2001.
- [12] W. M. Merrill, K. Sohrabi, L. Girod, J. Elson, F. Newberg, and W. Kaiser, "Open Standard Development Platforms for Distributed Sensor Networks," *Proc. SPIE Unattended Ground Sensor Technologies and Applications, AeroSense 2002*, vol. 4723, Orlando, FL, Apr. 2002.
- [13] J. Hill, R. Szwedczyk, A. Woo, S. Hollar, D. E. Culler, K. S. J. Pister, "System Architecture Directions for Networked Sensors," *Proc. Int. Conf. Architectural Support for Programming Languages and Operating Systems*, 2000.
- [14] S. Kumar and D. Shepherd, "SensIT: Sensor Information Technology for the Warfighter," *Proc. 4th Annual Conf. on Information Fusion*, Montreal, Quebec, Aug. 2002.
- [15] F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, "Collaborative Signal and Information Processing: An Information Directed Approach," *Proc. IEEE*, 2003, to appear.
- [16] R. R. Brooks, C. Griffin, and D. S. Friedlander, "Self-Organized Distributed Sensor Network Entity Tracking," *Int. Journal of High Performance Computing Applications*, vol. 16, no. 3, Fall 2002.
- [17] M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks," *Int. J. High Performance Computing Applications*, vol. 16, no. 3, pp. 90-110, Fall 2002.
- [18] C. Intanagonwivat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *ACM/IEEE Trans. Networking*, vol. 11, no.1, pp. 2-16, Feb. 2002.
- [19] S. Mori, C. Y. Chong, E. Tse, and R. P. Wishner, "Tracking and Classifying Targets Without A Priori Identification," *IEEE Trans. Automatic Contr.*, vol. AC-31, no. 5, pp. 401 - 409, May 1986.
- [20] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.
- [21] Y. Bar-Shalom and X. R. Li, *Multitarget Multisensor Tracking: Principles and Techniques*, YBS Publishing, Storrs, CT, 1995.
- [22] C. Y. Chong, D. Garren, and T. P. Grayson, "Ground Target Tracking - a Historical Perspective," *Proc. 2000 IEEE Aerospace Conf.*, vol. 3, pp. 433 - 448, Big Sky, MT, Mar. 2000.
- [23] D. B. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automatic Control*, vol. AC-24, vol. 6, pp. 843-854, Dec. 1979.
- [24] J.J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications," *Proc. 2nd Int. Workshop on Information Processing in Sensor Networks (IPSN'03)*, April, 2003.
- [25] J. Shin, L. Guibas, and F. Zhao, "A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-Hoc Sensor Networks," *Proc. 2nd Int. Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.