

Resource-Aware Multi-Target Tracking in Distributed Sensor Networks *

Juan Liu, Maurice Chu, and James E. Reich
Palo Alto Research Center
Email: { Juan.Liu, mchu, jreich}@parc.com

1 Introduction

The ability to track targets is essential in many applications. Well-established military applications include missile defense and battlefield situational awareness. Civilian applications are ever-growing, ranging from traditional applications such as air traffic control and building surveillance to emerging applications like supply chain management and wildlife tracking. In all of these applications, target tracking addresses the problem of combining sensed data and target history to provide accurate and timely knowledge of the location of one or more moving objects.

Current technology has enabled the development of *sensor networks*, distributed ad-hoc networks of hundreds or thousands of nodes, each capable of sensing, processing and communication. Much of the theory of tracking was developed for centralized processing of data from a relatively small number of radars or similar large devices endowed with plenty of power and high-bandwidth communications. Sensor networks demand a somewhat different approach, focused on scalable performance and the management of limited resources.

Tracking in distributed sensor networks has gained popularity for several major reasons: (1) As the cost of sensors and devices rapidly decrease, they can be deployed in large numbers to achieve

*The authors would like to thank Feng Zhao, Jie Liu, Leonidas Guibas, Jaewon Shin, Patrick Cheung, and Dan Larnier, for their collaborations that contributed to this work. This work was partially supported by the DARPA Sensor Information Technology program under Contract F30602-00-C-0139.

wide area coverage and their increased density allows sensors to reside far closer to the objects being sensed, improving sensing quality and discrimination. (2) Dense sensors enable overlapping coverage, which may result in increased robustness and improved accuracy. (3) Diverse sensing modalities provide complementary information. For example, certain types of sensors (e.g., laser range-finders) provide good ranging data, while others (e.g., microphone arrays) provide good directional data, and yet others (e.g., cameras) are ideal for object classification. This diversity in sensing modalities can be exploited to provide accurate and rich information about the target. (4) Spatial sensing diversity greatly mitigates the effects of obstructions on line-of-sight sensors.

In this paper, we provide a survey of techniques for tracking multiple targets in distributed sensor networks and introduce some recent developments. In the traditional centralized setting, multi-target tracking (MTT) is difficult. There is a combinatorial explosion in the space of possible multiple target trajectories due to the uncertainty in the association of observed measurements with known targets at each timestep. This *data association* problem has been the primary focus of the MTT literature. Tracking is also complicated by the fact that, for many sensing modalities, targets in close proximity tend to interfere with sensing one another. Compensating for this problem often requires sensing in a higher-dimensional joint space, again increasing computational complexity. Due to the above challenges, MTT is still an open problem even in centralized systems.

In distributed sensor networks, we have the additional challenge of mapping an MTT solution onto a sensor network platform with diverse resource limitations, including power, sensing, communication, and computation. Because data collection, processing, and dissemination all come at the cost of resource expenditure, MTT algorithms must make judicious use of resources while simultaneously addressing computational complexity issues. The more recent concepts introduced later in this paper are techniques for addressing these problems by appropriately partitioning the problem into local tasks tracking single targets, which may periodically be combined into small sets of interfering targets. This is combined with other techniques which maintain long-term identity information, explicitly tracking any unresolved confusion between targets and other approaches to resource management based on metrics of the expected usefulness of sensor data for each task.

We begin by reviewing single target tracking in distributed sensor networks in Sec. 2. We discuss these tracking algorithms and resource management issues because they prepare the foundation for MTT. In Section 3, we review the MTT problem and some typical approaches. These approaches

were originally developed for centralized systems, but in Sec. 4 we describe a method for distributing MTT in a sensor network by maintaining localized, compact target representations, at the cost of mixing target identities. This method provides near-optimal tracking performance with low communication and computation overheads. We illustrate this using our prior work [1] on MTT in an acoustic sensor network. Sec. 5 extends the MTT problem to scenarios where sensor resources are scarce relative to the number of targets or to the desired area of coverage. In this situation, sensing resources need to be multiplexed intelligently to maximize the overall performance. In Sec. 6, we discuss the most important remaining problems and suggest future directions.

2 Target Tracking in Sensor Networks

2.1 Estimation algorithms for target tracking

Tracking can be formulated as obtaining an estimate of target state x^t from a measurement history $\overline{z^t}$. Here $\overline{z^t}$ denotes the collection of measurements from initial time to the time t , i.e., $\overline{z^t} = \{z^{(0)}, z^{(1)}, \dots, z^{(t)}\}$. Without loss of generality, we assume tracking is in a 2-D plane, i.e., $x^t \in \mathcal{X}$, and $\mathcal{X} = \mathbb{R}^2$ or $\mathcal{X} = \mathbb{R}^4$ if velocity is included. Extending the tracking techniques presented in this paper to 3-D space or more general state spaces is straightforward.

For simplicity of illustration, we adopt the common assumption that the target's dynamics are characterized by a stationary Markov model $p(x^t|x^{t-1})$. Each sensor measurement z^t is related to the target state x^t via a given observation model $p(z^t|x^t)$ and are conditionally independent given the state. Under these assumptions, tracking can be performed by sequential Bayesian filtering:

$$p(x^t|\overline{z^t}) \propto p(z^t|x^t) \cdot \int_{\mathcal{X}} p(x^t|x^{t-1}) \cdot p(x^{t-1}|\overline{z^{t-1}}) dx^{t-1}. \quad (1)$$

The integral performs a prediction step, computing the distribution of likely states at time t from the target belief at $t-1$. Then, the multiplication by the likelihood incorporates the contribution of observation z^t . The filter equation (1) is recursive in the sense that the current filter distribution $p(x^t|\overline{z^t})$ is computed from the previous filter distribution $p(x^{t-1}|\overline{z^{t-1}})$ and the new observation z^t .

The well-known Kalman filter is a special case of sequential Bayesian filtering under the assumption that the object dynamics and the observation model are both linear in x^t and the uncertainty

in both models are Gaussian. Under these two assumptions, the posterior belief $p(x^t|\overline{z^t})$ is also Gaussian. Because Gaussians are completely characterized by their mean and covariance, the Kalman filter equations update the mean $\hat{x} \triangleq E[x^t|\overline{z^t}]$ and covariance $P \triangleq E[(x^t - \hat{x})(x^t - \hat{x})^T]$ recursively as measurements are observed. The Kalman filter is computationally efficient, but its performance is limited by its modeling assumptions. Variations such as the extended Kalman filter (EKF) and the unscented Kalman filter (UKF)[2] have been proposed to push this beyond the linear Gaussian assumptions. Another alternative gaining popularity recently is the particle filter [3], a nonparametric Monte-Carlo sampling-based method, representing a probability distribution as a set of weighted point samples, $\{x_i, w_i\}_{i=1}^n$, referred to as a particle set. The particle filter algorithm updates the sample points $\{x_i\}$ and their weights $\{w_i\}$ based on the target dynamics $p(x^t|x^{t-1})$ and the observation likelihood model $p(z^t|x^t)$. This representation has the flexibility to accommodate nonlinear dynamics and multi-modal observation models but at the cost of more computation and storage requirements. We refer the reader to [3] for more details.

2.2 Managing limited sensor network resources

We categorize sensor network resources into four broad categories: power, sensing, communication, and computation. For example, imagine a typical sensor network consisting of a large number of battery operated tiny sensor nodes, each with a wireless antenna and inexpensive CPU, mixed with a smaller number of high-end sensors. Prolonging the period of time these sensors can operate is desirable so that power is likely to be a key constraint. Sensors (especially the high-end ones) may need to be shared among multiple coexisting applications. The wireless communication medium has limited throughput capacity [4] so that applications must limit their communication requirements to avoid overloading the network. Finally, tiny inexpensive sensor nodes are often limited in computational capability so that developers may need to implement computationally lightweight algorithms that sacrifice sensing quality but take advantage of the distributed computation resources of the sensor network. In this section, we give a high-level sampling of the quickly accumulating sensor network literature and describe a few examples of target tracking under power, sensing, communication, and computation constraints.

2.2.1 Power conservation

Sleep scheduling has been a major topic for power conservation. The basic idea is that sensors can be selectively ordered to sleep or wake up. One idea is to develop a special low power wakeup channel to wake up a sleeping node, but Fuemmeler and Veeravalli [5] have made the argument that these wakeup channel ideas are impractical given the current state of technology. They propose an alternative strategy where the sensor network plans its sleep schedule based on the available information about target locations and trajectories. In their approach, sleep scheduling is formulated as a POMDP (partially observable Markov decision process) problem and solved via dynamic programming.

Example 2.A. Power conservation for surveillance and tracking. In [6], two operation modes are defined for target tracking: (1) a surveillance mode when there is no target present, and (2) a tracking mode when a target emerges. In the surveillance mode, a set of novel metrics for optimality is proposed, such as the quality of surveillance (QoS_v), defined as the inverse of the expected length that a target can travel without being detected. Optimal sleep schedules have been derived to minimize power usage while maintaining a level of QoS_v. This is similar to the concept of maintaining “peripheral awareness” in the MTT example to be shown in Sec. 5. Similar tradeoffs have been proposed in [7]. In the tracking mode, the sensor network has more detailed information about where the target is and can infer where the target is going to be. Hence, nodes can schedule their sleep with better temporal and spatial precision. This idea is common and found in various tracking schemes such as in [8, 9]. □

2.2.2 Sensor tasking

The idea of sensor tasking is to activate the minimum number of sensors while maintaining an acceptable level of sensing quality [10, 11, 12]. We give an example of sequential sensor selection which tasks a single sensor at a time. Extensions of this idea include tasking a cluster of sensors within a local scope.

Example 2.B. Information-driven sensor querying (IDSQ). IDSQ [10] is a sequential tracking scheme where, at any given point of time t , there is only one sensor active. All the other nodes remain in power-conserving sleep states. The active sensor takes a measurement and updates

the belief $p(x^t|\bar{z}^t)$. It then decides which sensor in its neighborhood is the the most “informative”, hands the belief off to that sensor, and returns to the sleep state. The sensor receiving the handoff becomes active, and this operation repeats. Intuitively, by selecting the most informative neighbor, the active sensor is seeking good quality data. In [10], the sensor selection criterion is described as

$$k_{IDSQ} = \arg \max_{k \in \mathcal{N}} I(X^{(t+1)}; Z_k^{(t+1)} | \bar{Z}^t = \bar{z}^t), \quad (2)$$

where \mathcal{N} is the neighborhood, and $I(\cdot)$ measures the mutual information between a sensor’s measurement and the underlying target state. This criterion seeks the best complementary data: i.e., the sensor whose measurement $z_k^{(t+1)}$, combined with the current measurement history \bar{z}^t , provides the most information about the target location $x^{(t+1)}$. In this way, target tracking takes advantage of sensing modality and spatial diversity while keeping sensor usage to a minimum. \square

2.2.3 Efficient communication

Efficient communication has always been a major focus of networking research. Recent advances in ad-hoc wireless networking have generated a large body of literature which is also applicable to sensor networks. In this paper, we will not focus on such pure communication problems, but rather focus on efficient communication specifically in support of tracking applications. In sensor networks, communication is often not the end goal, but rather a tool serving certain applications. Hence, communication needs to be optimized not just with respect to its own metrics, but also with respect to the application performance. For example, in the survey paper summarizing the interplay between signal processing and networking [13], a few techniques to optimize communication to support detection and parameter estimation are presented (see the references therein). Representative work that optimize communication efficiency in target tracking include [14, 11, 7, 15].

2.2.4 Distributed computation

Sequential Bayesian filtering (1) is recursive and can be implemented in a distributed fashion. One consequence of distributed tracking is that one target can be tracked by multiple sensors (or multiple clusters of sensors) independently. For example, two target beliefs are derived, say, $p(x^t|\bar{z}_{S_1}^t)$ and $p(x^t|\bar{z}_{S_2}^t)$, each from a sensor set (S_1 and S_2 respectively). Suppose we know that the two beliefs

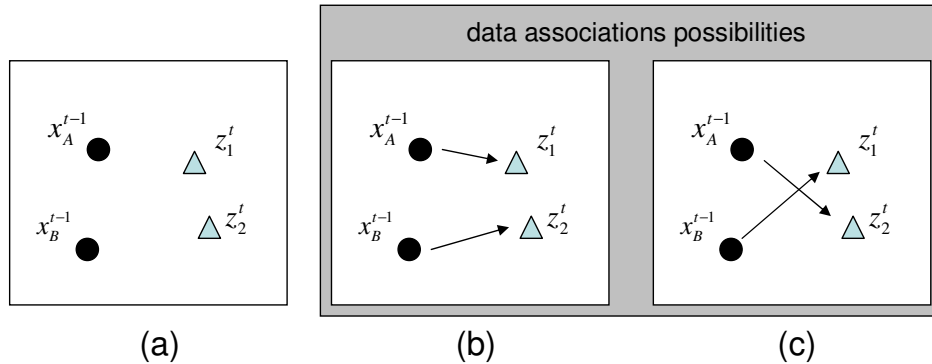


Figure 1: Data Association Example. (a) shows the two target (circles) example with two measurements (triangles) to associate. (b) and (c) are the two possible data associations when each target generates exactly one measurement and there are no false alarms.

correspond to the same target. Now, how should one consolidate the two beliefs into one? — This is known as the distributed fusion problem and has been addressed in a number of publications. The basic idea is to discount the contribution from overlapped sensors [16] if the overlap between S_1 and S_2 is known.

3 Tracking Multiple Targets

MTT is not a trivial extension of single target tracking but rather a challenging topic of research. The foremost difficulty is the problem known as the *data association* problem. To elaborate, consider the simple case of tracking two targets, illustrated in Fig. 1a. At time $t - 1$, say that target A is believed to be located at point x_A^{t-1} , and that target B is believed to be located at point x_B^{t-1} as shown in Fig. 1a. At time t , the system observes two measurements z_1^t and z_2^t . The extra ambiguity in multiple target tracking is the question of which measurement was generated by target A and which was generated by target B . Assuming that each target will generate exactly one measurement and there are no false alarms, there are two possible associations between tracks and measurements: z_1 corresponding to target A and z_2 corresponding to B shown in Fig. 1b; or vice versa as in Fig. 1c.

If we generalize to the case of N targets generating exactly N measurements with no false alarms and missed detections, the number of possible associations is combinatoric, $N!$, and becomes com-

putationally unwieldy for large N . Furthermore, if we consider the number of possible associations over a window of T scans, the number of possible associations is exponential in the number of scans, $(N!)^T$. The computational complexity is even worse when we relax the assumption to allow false alarms, missed measurements, and multiple measurements to be generated from each target.

3.1 General Bayesian formulation of MTT

MTT is an estimation problem with data association ambiguity. We can formulate MTT rigorously as a sequential Bayesian filtering problem of a Markov process with noisy measurements, just as in the single target case in Sec. 2. The main difference is that the state space and observation space are more complex. This general formulation provides a theoretical foundation to understand how the various techniques found in the literature are approximate solutions.

The analog of the single target state $x \in \mathcal{X}$ in Sec. 2 is the “multi-target state” of N targets, which can be represented by an N -tuple $(x_1, \dots, x_N) \in \mathcal{X}^N$. Since we do not know the number of targets, the state space is given by

$$\mathcal{S} = \emptyset \cup \mathcal{X} \cup \mathcal{X}^2 \cup \mathcal{X}^3 \cup \dots ,$$

which is the union of the possibility that there are no targets (\emptyset), that there is 1 target (\mathcal{X}), that there are 2 targets (\mathcal{X}^2), and so on for any finite N number of targets (\mathcal{X}^N).

The transition model from a multi-target state s^{t-1} at time $t-1$ to s^t at time t can be modeled by a transition probability $p(s^t | s^{t-1})$, which is analogous to the single target dynamics model with extra modeling of how targets enter and disappear. To illustrate, one of the simplest examples of specifying this transition probability is given in [17], which is derived from the assumptions that (1) all targets follow the same motion model $p(x^t | x^{t-1})$, (2) the probability that a new target enters the scan area is given by $p_n \in [0, 1]$, (3) the distribution of a new target’s location is given by $p_{new}(x)$, and (4) the probability that a target disappears is given by $p_d \in [0, 1]$. Application-specific information like how targets enter and exit an area can be encoded into $p(s^t | s^{t-1})$, which makes this formulation applicable to a wide range of scenarios.

In the multi-target case, measurements are a collection of observations generated by the multiple targets and false alarms. Thus, if \mathcal{Z} is the measurement space of a single target, the space of

measurements in the multi-target case is the collection of all subsets of \mathcal{Z} , which we will denote by \mathcal{M} . Then, the measurement model is given by a likelihood function $p(m^t|s^t)$, where m^t is the observation collection taking values in \mathcal{M} . In the two target example of Fig. 1, the multi-target measurement is the set $m^t = \{z_1^t, z_2^t\}$. The data association problem arises because the space of measurements is unordered subsets of points which do not reveal the association between target and measurement.

Theoretically, the standard recursive Bayesian filtering techniques can be applied directly to the above general Bayesian formulation for multi-target tracking by computing the filtered distribution $p(s^t|m^0, \dots, m^t)$. However, computing the filtered distribution over the multi-target state space \mathcal{S} and dealing with the combinatorial explosion of possible states due to the data association ambiguity is difficult in practice. Therefore, the main challenge of realizing an MTT system is to manage the computational complexity of the problem while still providing reasonable tracking performance.

3.2 Overview of the traditional MTT approaches

No discussion on multiple target tracking would be complete without mentioning the following two predominant approaches.

MHT which stands for multiple hypothesis tracking was proposed by Reid [18]. The idea is to exhaustively enumerate recursively the set of all associations, called *hypotheses*, of measurements to existing tracks, new tracks, and false alarms while respecting the mutual exclusion association constraint. An advantage of this approach is that the number of tracks need not be known a priori because track initiations and terminations are explicitly hypothesized. Furthermore, data association decisions are effectively delayed until more data is received since multiple hypotheses are kept. Thus, MHT can address low detection probability, high false alarm rates, initiation and termination of tracks, and delayed measurements. However, this approach suffers from large storage space requirements and exponentially increasing processing, so that a key part of making this approach practical is to prune bad hypotheses or combine similar hypotheses as in [19].

JPDAF which stands for the joint probabilistic data association filter [20] was proposed by Fortmann, Bar-Shalom, and Scheffe. The approach is to update each individual track state with weighted combinations of all measurements. Thus, the key part of this approach is computing

the probability that measurements can be associated with tracks so that the mutual exclusion constraint is respected. A disadvantage of this approach is that the number of targets needs to be known a priori.

Both approaches are approximations of the true filtered distribution $p(s^t|m^0, \dots, m^t)$. MHT is a brute force approach which can only approximate the true filtered distribution due to the need for pruning and/or combining hypotheses to limit the combinatorial explosion. On the other hand, JPDAF makes soft data association decisions by incorporating a weighted effect of all measurements to each track, which avoids the combinatorial explosion of MHT but suffers in track quality.

The relationship between JPDAF and MHT has been underemphasized in the literature although there was brief mention of this relationship as early as Reid's original paper [18]. JPDAF is a particular way of combining the multiple hypotheses generated by MHT into a single hypothesis at each time step and, therefore, can be viewed as an instance of MHT. We will elaborate on this relationship now because all approaches to data association can be viewed as instances of MHT, and the idea of combining hypotheses is the conceptual foundation behind new resource-aware representations to be discussed in Sec. 4.

Example 3.A. Relationship between MHT and JPDAF. Consider the two target case where track A and B are independently distributed according to $p_A^{t-1}(x)$ and $p_B^{t-1}(x)$ respectively. There are two measurements observed at time t given by z_1^t and z_2^t . Assuming that there are no false alarms or missed measurements for the sake of simplifying the discussion, there are two hypotheses generated by MHT.

- H_0 : track A associates with z_1^t and track B associates with z_2^t
- H_1 : track A associates with z_2^t and track B associates with z_1^t

To compute the association probabilities, we must first predict each track's belief forward to the current time t

$$\hat{p}_j^t(x) = \int_{\mathcal{X}} p(x^t|x^{t-1})p_j^{t-1}(x^{t-1})dx^{t-1}$$

for $j \in \{A, B\}$. Then, we can compute the probability that track A and B generate z_1^t and z_2^t for each hypothesis.

$$\gamma_0 = P(A \text{ generates } z_1^t \text{ and } B \text{ generates } z_2^t) \tag{3}$$

$$= \int_{\mathcal{X}^2} P(z_1^t|x_A) \cdot P(z_2^t|x_B) \cdot \hat{p}_A^t(x_A) \cdot \hat{p}_B^t(x_B) dx_A dx_B \quad (4)$$

$$\gamma_1 = P(A \text{ generates } z_2^t \text{ and } B \text{ generates } z_1^t) \quad (5)$$

$$= \int_{\mathcal{X}^2} P(z_2^t|x_A) \cdot P(z_1^t|x_B) \cdot \hat{p}_A^t(x_A) \cdot \hat{p}_B^t(x_B) dx_A dx_B \quad (6)$$

Since we are given that the observed measurement is the set $\{z_1^t, z_2^t\}$, the association probabilities are given by the following.

$$P(H_0) = \frac{\gamma_0}{\gamma_0 + \gamma_1}, \quad P(H_1) = \frac{\gamma_1}{\gamma_0 + \gamma_1}$$

Note that these association probabilities are computed based solely on the the measurement model $P(Z|X)$ and the *predicted* distributions of the tracks $\hat{p}_A^t(x)$ and $\hat{p}_B^t(x)$. Thus, the target dynamics play a key role in determining the association probabilities.

Under each hypothesis, the track states of A and B can be updated using Bayes' Rule with the measurement that is associated with it. In particular, the belief of track A under hypothesis H_0 and H_1 is given by

$$p_A^t(x|H_0) = \alpha_0 \cdot P(z_1^t|x) \cdot \hat{p}_A^t(x), \quad p_A^t(x|H_1) = \alpha_1 \cdot P(z_2^t|x) \cdot \hat{p}_A^t(x)$$

where α_0, α_1 are the usual normalization constants. Note that the track state is updated with different measurements under different hypotheses. Thus, MHT maintains a separate track state under each hypothesis.

JPDAF combines these multiple hypotheses into a single one by mixing the beliefs of the same track over all hypotheses.

$$p_{JPDAF,j}^t(x) = p_j^t(x|H_0) \cdot P(H_0) + p_j^t(x|H_1) \cdot P(H_1)$$

for each track $j \in \{A, B\}$. This ‘‘marginalization by track’’ is the soft data association approach of JPDAF. In Sec. 4, this idea of combining hypotheses is expanded to new techniques of ‘‘generalized marginalization’’ for the purposes of minimizing resource usage. \square

3.3 New generation of MTT approaches

Monte Carlo-based sampling methods and graphical models have spawned new techniques to deal with the computational complexity of the data association problem. Rather than generating hypotheses, these approaches search the space of hypotheses. We describe a Monte Carlo-based

method called Markov chain Monte Carlo data association (MCMCDA) here and refer the reader to [21] for a graphical model approach due to space limitations.

Example 3.B. MCMCDA. Markov chain Monte Carlo (MCMC) methods are a class of algorithms that sample from complex probability distributions by constructing a Markov chain so that the desired distribution is its stationary distribution. Applying an MCMC-based approach to data association was first proposed in [22]. By considering the space of association hypotheses from a window of scans, the constructed Markov chain sets up five types of transitions between hypotheses that correspond semantically to (1) birth/death, (2) split/merge, (3) extension/reduction, (4) track update, and (5) track switch moves. The transition probabilities are chosen in a way so that the stationary distribution of this Markov chain is the true association probabilities. Samples are drawn from this distribution, and the sample with the highest probability is considered the best association hypothesis. This approach can be considered a kind of simulated annealing at a constant temperature and has only probabilistic guarantees of finding the best hypothesis. The advantage of this approach is that there is no longer the need to have a large memory store since hypotheses are never explicitly enumerated although there must be enough memory to store all measurements from the window of scans.

4 Managing resources: Switching between single target tracking and MTT

MTT is generally an expensive task in terms of sensing, computation and communication. One natural idea to reduce resource expenditure is to reduce to single target tracking when targets are far apart and switch to MTT only when data association becomes ambiguous. Fig. 2 illustrates the idea. Initially, targets T_1 and T_2 are well-separated. Tracking them separately provides nearly optimal performance. All the resource-aware techniques described in Sec. 2.2 can be applied. As T_1 and T_2 approach each other, tracking should switch to the MTT mode. Resource expenditure is higher but can be confined to the local vicinity around the targets. As the targets separate, tracking returns to single target tracking mode. This common idea has been implemented in distributed sensor networks as in [1, 23].

Due to the uncertainty in data association, track beliefs of each physical target (T_1 or T_2) may

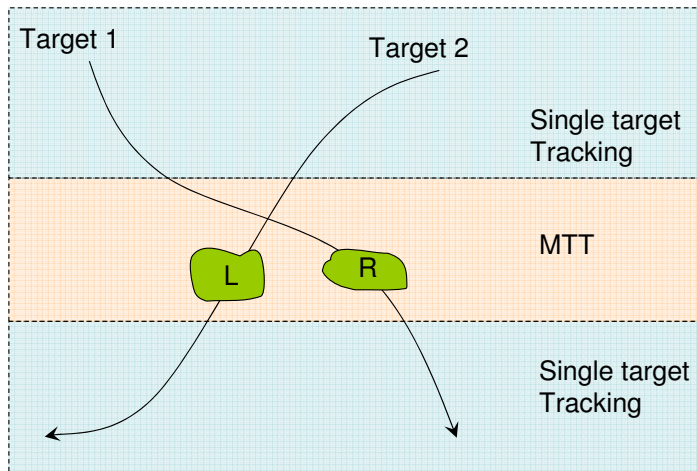


Figure 2: Decomposition of multi-target tracking problem in close range and long range.

be non-compact. For example in Fig. 2, after the two targets cross over, by marginalizing data association hypotheses for track T_1 (See Example 3.A), the belief may be a bimodal distribution like the two blobs in the figure. L and R in the figure stand for “left” and “right” respectively. As T_1 and T_2 move away from each other, the two blobs corresponding to target T_1 also move away from each other since there is no information to distinguish the “correct” association. Representing the track belief in this bi-modal form makes sense in centralized systems since we retain all likely locations of target T_1 . However, this representation is very expensive in distributed sensor networks because sensors around both blobs must be tasked to obtain measurements, and the measurements need to be communicated to the node updating the track belief. As targets move farther and farther apart, this communication can be very long range. Thus, it is desirable to design a representation where track beliefs are compact around a local region of sensors.

4.1 Local compact representation about logical targets

One way to maintain compact track belief representations is to move away from representing the the physical targets T_1 and T_2 to a representation of the “logical targets” [1, 24, 25]. For example, we could represent the track belief of the left target L or the right target R in Fig. 2. The effect is that the target state representation is compact but with the consequence that the target identities are now mixed. That is, logical target L can be target T_1 with some probability or target T_2 otherwise.

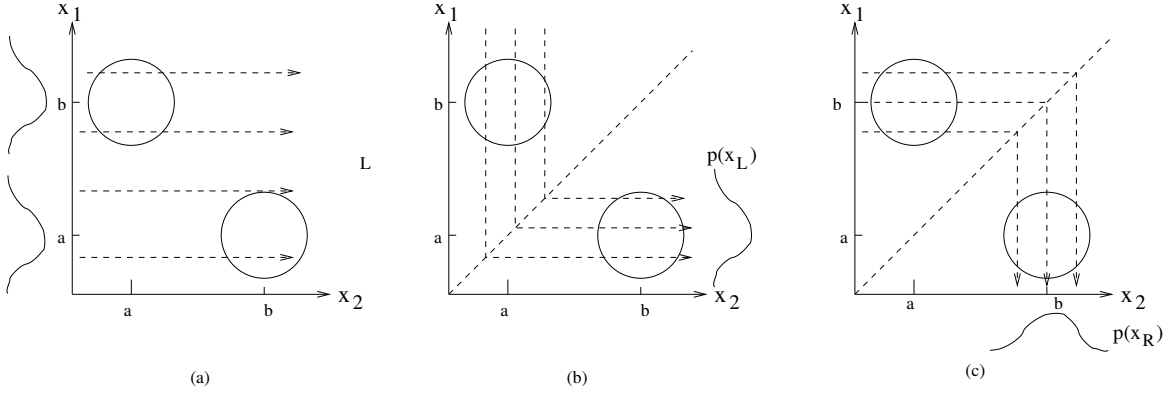


Figure 3: State space representation: (a) for physical target T_1 , (b) for logical left target, and (c) for logical right target. Dashed lines with arrow show marginalization path.

For ease in illustration and visualization, we use a simplified example of tracking two targets along a 1-D line. Target T_1 could be located at location a or b ($a < b$), and so is T_2 . In the joint space $\mathcal{X}_1 \times \mathcal{X}_2$, there are two blobs, around (a, b) and (b, a) . The marginal belief for target T_1 is bi-modal (Fig. 3a). Instead, we can consider the distribution for the logical left target $x_L = \min(x_1, x_2)$ and the logical right target $x_R = \max(x_1, x_2)$. This “marginalization” is given by

$$p(x_L) = \int_{x_1 \geq x_L} p(x_1, x_L) dx_1 + \int_{x_2 \geq x_L} p(x_L, x_2) dx_2, \quad (7)$$

$$p(x_R) = \int_{x_1 < x_R} p(x_1, x_R) dx_1 + \int_{x_2 < x_R} p(x_R, x_2) dx_2. \quad (8)$$

Fig. 3b shows the marginalization slices for x_L . It is an integral over an L-shaped line. Since by definition, $x_L = \min(x_1, x_2)$, the integral is over a vertical line x_1 when $x_1 \geq x_2$, and over horizontal line x_2 when $x_2 \geq x_1$. Similarly Fig. 3c shows the marginalization slices for the right target $x_R = \max(x_1, x_2)$. This nonlinear operation produces compact representations for the logical target locations. For tracking in 2-D, one needs to define logical targets properly. In our work [1], we define a separation plane by collecting particles from both targets into one collection and computing its minor axis. The logical targets are then defined as the target on each side of the separation plane. This works well for two-target crossing. When more than two targets cross each other, one may use other techniques such as k -means clustering. This is the “generalized marginalization” we alluded to in Example 3.A which is conceptually a novel way of combining the tracks of multiple data association hypotheses.

As for when one should switch between single target tracking and MTT, several factors need to be considered. From the pure estimation performance point of view, performing MTT in the joint target state space more accurately estimates track states because it can explicitly take into account interactions between targets. On the other hand, tracking a single target is an estimation problem in a lower dimensional state space \mathcal{X} , while MTT in the joint space is an estimation problem in \mathcal{X}^N . This higher dimensionality means (1) more data samples are needed to support the estimation, and (2) if a particle filter (or similar non-parametric methods) is used for tracking, more particles will have to be used. This has implications on computational complexity and communication cost.

4.2 Simplifying target mixing into belief matrix

Switching the representation to logical targets circumvents the problem of sensing around spatially wide-spread bi-modal target beliefs and eliminates the need for frequent long range communication, but it creates an additional problem: how do we know which physical target corresponds with each logical target? This problem is called the identity management problem.

Traditional MHT-type approaches keeps the entire data association history to perform Bayesian inference on target identity. While this is theoretically optimal, it is too expensive with $O(N!)$ computation complexity, where N is the number of targets. If one wants to store all MHT hypotheses and carry that from node to node, storage and communication may also become a problem. This calls for suitable approximation to reduce resource cost. We refer to the identity management method proposed by Shin et al [24]. Similar approaches include [25, 23]. The idea is to maintain an approximation of identity ambiguity at a low cost and postpone the identity clarification until classification evidence is collected. Target identity is represented using an $N \times N$ identity belief matrix \mathbf{B} , whose elements are defined by

$$\mathbf{B}_{i,j} = Prob(\text{estimate } x_j \text{ comes from target } i), \quad (9)$$

for all $i, j = 1, \dots, N$. This formulation assumes no emerging or disappearing targets, but these assumptions have been loosened in [25]. In this identity belief matrix:

- Each column j represents the possible identity of a track, i.e., it could be target T_1 , T_2 , or T_N respectively with probability $B_{1,j}$, $B_{2,j}$, or $B_{N,j}$.

- Each row i represents where target i can be, i.e., target i can be track 1 with probability $B_{i,1}$, track 2 with probability $B_{i,2}$, and so on.

Each row and column sum up to 1, known as the “doubly stochastic” property. This comes from the fact that there is a one-to-one mutually exclusive association between physical targets and logical targets.

The identity belief matrix summarizes the identity information up to the current time and evolves as targets mix and separate. The mixing of identity is modeled as a mass flow: when two targets cross, mass will be redistributed between the two tracks, causing the effect of “smearing” track identity. Mathematically, it is formulated as

$$\mathbf{B}(t + 1) = \mathbf{B}(t)\mathbf{M}(t + 1), \quad (10)$$

where \mathbf{M} is a “mixing” matrix, which can be derived from the data association history (for example, see [23]).

When targets split apart and classification evidence is collected, one needs to “de-mix” and restore target identity. This is done by redistributing identity mass to restore the doubly stochastic property of \mathbf{B} (see the reference in [24]). Again, this is an approximation but works well in practice, reducing the computational complexity to $O(N^2)$. For comparison, an exact Bayesian inference method needs to keep track of all possible identity assignments, and the worst case complexity is $O(N!)$. The savings are significant.

This identity management scheme is suitable for implementation in a distributed sensor network. The identity belief matrix \mathbf{B} has the structure that each column encapsulates the identity of a track, and can be stored in a single node along that track (the track leader). The god’s eye view of the sensor network is as follows: When targets are far apart, they each have a track leader in their vicinity, carrying information regarding the track belief and the respective identity information, which is simply a column in \mathbf{B} . As targets mix, the track leaders modify their belief and identity. As targets separate, the leaders also separate but maintain a link among themselves [26]. When one leader receives identity evidence, for example, from a sensor with good classification results, it notifies the other leaders, and they adjust their identity information accordingly. The communication between these leaders is long range but infrequent.

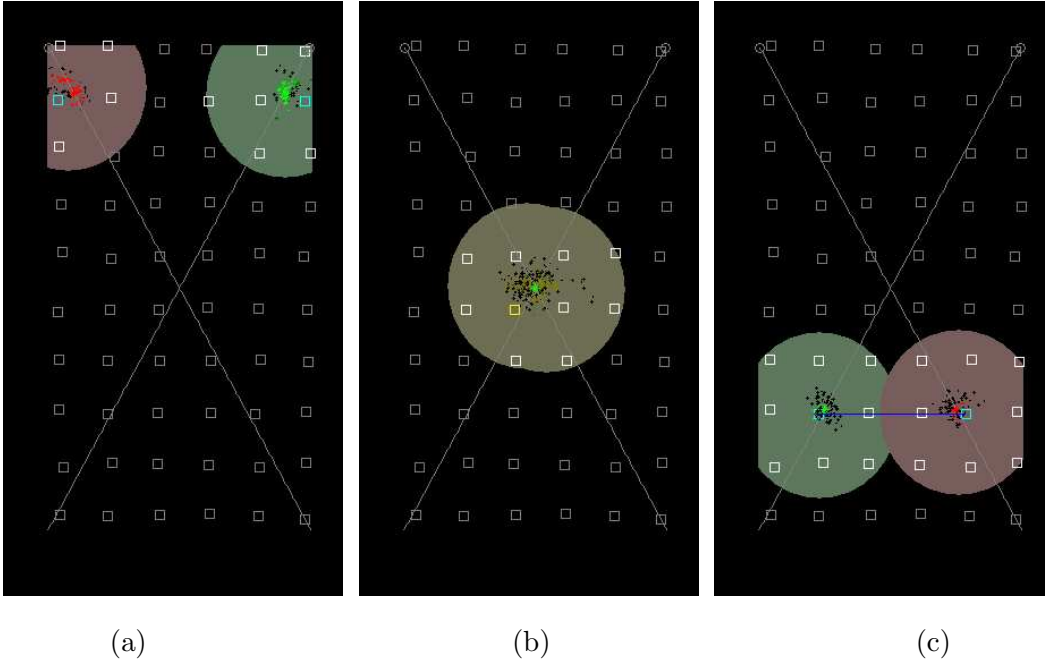


Figure 4: Tracking two crossing targets.

4.3 Example: Tracking MTT in a distributed acoustic sensor network

Fig. 4 demonstrates tracking of two targets crossing in a rectangular sensor field. The targets (red and green) start at the top-left and top-right corners and move down along the diagonals. The trajectories are shown in grey lines. The field is covered by acoustic energy sensors (marked with little squares in the figures). Each sensor picks up acoustic energy from any targets within its sensing range ($R_{sense} = 120\text{ft}$), together with some random background noise, modeled as Gaussian noise.

Initially, the targets are well separated at the two corners of the sensor field. Fig. 4a shows a snapshot. The targets are tracked individually, each with a particle filter with 100 particles (visualized in red and green color respectively). Sensors within a sensing range R_{sense} , shown as shaded disks centered at the estimated target locations, are organized into a cluster and elect a leader. The leader is responsible for collecting measurements from the group members, updating the track, and maintaining the cluster structure.

When two targets move closer, their clusters collide. Collision is flagged when a sensor finds itself led by two distinct cluster leaders. At this time, tracking switches to MTT mode, and two

clusters merge into one. A node closest to the centroid of the original two clusters is elected to be the new leader. Fig. 4b shows the snapshot when the two targets are exactly at the same location. Target positions are tracked fairly accurately, but target identities are mixed. In Fig. 4b, the color of the particles are also blended proportionally and result in a range of gray to black. At this point, it is impossible to distinguish one from the other. After the two targets cross over, the estimated locations start to diverge. We set $R_{separate} = 140$ as the switching point to go back to single target tracking mode, slightly larger than the sensing range $R_{sense} = 120$. After the switch, the two tracks are maintained separately. Interested readers may refer to [1] for multi-target tracking in more complicated target crossing scenarios.

5 MTT with scarce sensors

The previous examples showed how resource use may be minimized in sparse target scenarios by limiting activity to the subset of sensors near the targets. Two other classes of sensor network MTT problem are: (1) those with *too many* targets to track simultaneously, and (2) those with *too few* sensors for full coverage. These are “triage” situations, where it is impossible to handle all of the data from all sensors or to fully cover the area using the available sensors.

An example is shown in Fig. 5 in which several moving targets are tracked by a set of pan-tilt cameras. These cameras may select from a subset of fixed fields of view (FOVs) A-F, and are marked with letters indicating their viewable FOVs. Each target is labeled with a value, and may appear or disappear at any time. In the leftmost panel of Fig. 5, no targets are present, and the cameras will switch from FOV to FOV, searching for a new target. In the middle panel, a single target has been detected, and cameras must trade off the surveillance task against tracking the target. Passing the tracking task between sensors, or even ignoring the target momentarily may be optimal. In the rightmost panel, more targets than cameras are present, and the system must pick the most valuable. This requires global cooperation. With a greedy approach, the top cameras might have chosen to view region E but would have missed the targets in regions A and C which only they can view.

In this resource allocation problem, cameras are scarce and must prioritize their tasks according to some *utility function* giving the relative value of viewing a particular FOV. Such a utility function

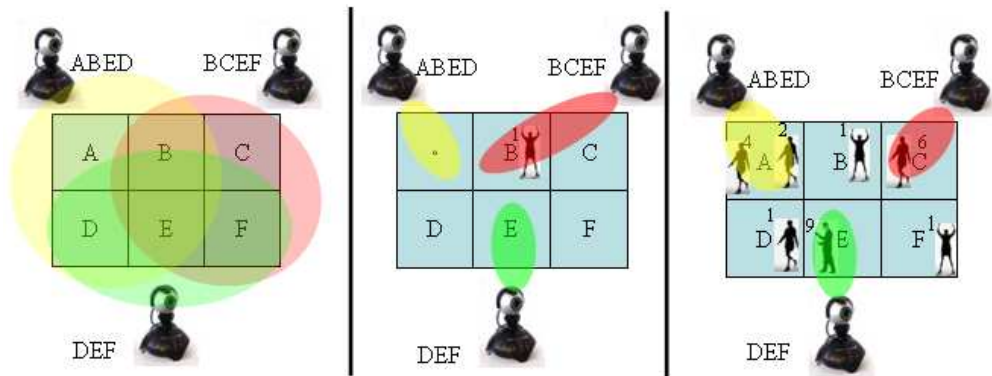


Figure 5: Example MTT Scarce Sensor Management Problem

might be based on the expected information gain of a sensor configuration, as the IDSQ [10] metric, or even based on tactical considerations, as in [27]. Regardless of the utility function, the goal of these systems is to allocate sensing resources so as to maximize the total utility achieved.

In sensor networks, it is often impractical to allocate resources centrally, due to issues of scalability and latency. As a result, we cannot guarantee global optimality. In practice, however, local sensor allocation decisions rarely have effects over long distances, and local approximations can be solved by decentralized methods such as auction methods [28] or the max-sum algorithm [29].

In a real-world experiment [30], we used pan-tilt cameras to track a number of “abnormal” vehicles hiding among robotic vehicles behaving in a orderly fashion. The system contained two classes of tasks: (1) “peripheral awareness” tasks, searching for new targets that may emerge, and (2) a particle-filter-based tracking task once a vehicle is detected. Utility functions are assigned to each class of tasks based on the uncertainty level of detection and tracking. FOVs were then chosen to optimize the total utility across all tasks, according to a variant of max-sum [30] in which a factor graph captured the relationships between each term of the utility function and the relevant sensors. In practice, this system demonstrated many of the properties we desired: when no targets were present, it slightly exceeded the detection performance of a random search pattern. As the number of targets increased, it detected new targets quickly while still keeping a camera on each target a significant fraction of the time, outperforming a random search pattern in nearly every case.

6 Discussion

With the advent of sensor networks, the MTT problem moves from a centralized task performed on a handful of radar tracking stations to a ubiquitous function in networks of thousands of inexpensive sensor nodes. In these systems, resource management is even more critical than in traditional tracking systems due to the limited bandwidth of the shared wireless channel, the limited availability of battery energy or solar power, and the limited computational capabilities of sensor nodes. This paper has presented an end-to-end tutorial of how multiple target tracking can be implemented for a resource-limited distributed sensor network platform. By selecting an appropriate fusion mechanism, sensor utility metric and a sensor tasking approach, one can produce a system which efficiently tracks targets as independent entities while they are widely separated. By adding algorithms for tracking in the joint space for targets in close proximity to each other and an identity management scheme to handle potential confusions between crossing targets, we need consider only a small number of targets at a time, avoiding the exponential complexity when possible.

In adapting the MTT paradigm for sensor networks, the centrality of sensor management has become clear. Researchers need to begin considering the larger problem of *resource management*. Resources include sensor resources, network resources, computational resources and energy/power resources. What is critical to understand here is that these resources can be traded off between categories. For example, one might substitute a bank of fixed cameras for a pan-tilt unit. If tracking is performed locally, then local processing can be allocated to perform tracking for a particular direction— a “virtual pan-tilt.” Similarly, if tracking is performed offboard, a unit of network bandwidth spent in transporting data from an additional camera is equivalent to another virtual pan-tilt. In the past, these tradeoffs have been commonplace in hardware and system design, but in a resource managed system, they may be varied dynamically according to the system’s current capabilities and requirements.

As sensor networks move away from tracking tanks in the desert and ships at sea, and into crowded human environments, traditional multiple target tracking approaches will hit their limits. These systems will need to know their power, computation and communication limits, focus their sensing resources, and partition their inference tasks appropriately. This is how they can tame a complex world.

References

- [1] J. Liu, M. Chu, J. Liu, J. Reich, and F. Zhao, “Distributed state representation for tracking problems in sensor networks,” in *The 3rd International Symposium on Information Processing in Sensor Networks*, (Berkeley, CA), April 2004.
- [2] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” *Proc. of AeroSense*, 1997.
- [3] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York, 2001.
- [4] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Info. Theory*, vol. 46, pp. 388–404, Mar. 2000.
- [5] J. Fuemmeler and V. V. Veeravalli, “Smart sleeping policies for energy efficient tracking in sensor networks.” submitted to *IEEE Trans. Signal Processing*, October 2006. Also available on <http://www.ifp.uiuc.edu/~vvv/cv/pubs.html>.
- [6] C. Gui and P. Mohapatra, “Power conservation and quality of surveillance in target tracking sensor networks,” in *MobiCom’04*, (Philadelphia, Pennsylvania), 2004.
- [7] T. He, S. Krishnamurthy, J. A. Stankovic, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hu, and B. Krogh, “Energy-efficient surveillance system using wireless sensor networks,” in *Proc. MobiSys’04*, (Boston, MA), 2004.
- [8] D. Friedlander, C. Griffin, N. Jacobson, S. Phoha, and R. R. Brooks, “Dynamic agent classification and tracking using an ad hoc mobile acoustic sensor network,” *EuraSip Journal on Applied Signal Processing*, no. 4, pp. 371–377, 2003.
- [9] W. Zhang and G. Cao, “Optimizing tree reconfiguration for mobile target tracking in sensor networks,” in *Proc. InfoCom’04*, (Hong Kong, China), Mar. 2004.
- [10] J. Liu, J. E. Reich, and F. Zhao, “Collaborative in-network processing for target tracking,” *EURASIP, Journal on Applied Signal Processing*, vol. 2003, pp. 378–391, Mar. 2003.

- [11] W.-P. Chen, J. C. Hou, and L. Sha, “Dynamic clustering for acoustic target tracking in wireless sensor networks,” in *Proc. 11th IEEE International Conference on Network Protocols*, pp. 284–294, 2003.
- [12] Y. He and E. Chong, “Sensor scheduling for target tracking in sensor networks,” in *Proc. 44th Conf. Decision and Control*, 2004.
- [13] Q. Zhao, A. Swami, and L. Tong, “The interplay between signal processing and networking in sensor networks,” *IEEE Signal Processing Magazine*, July 2006.
- [14] R. Brooks, P. Ramanathan, and A. M. Sayeed, “Distributed target classification and tracking in sensor networks,” *Proc. of the IEEE*, vol. 91, no. 8, pp. 1163–1171, 2003.
- [15] J. Liu, F. Zhao, and D. Petrovic, “Information-directed routing in ad hoc sensor networks,” *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Self-Organizing Distributed Collaborative Sensor Networks*, vol. 23, no. 4, April 2005.
- [16] M. E. Liggins, C.-Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, “Distributed fusion architectures and algorithms for target tracking,” *Proc. of the IEEE*, Jan. 1997.
- [17] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. 1988.
- [18] D. B. Reid, “An algorithm for tracking multiple targets,” *IEEE Trans. On Automat. Control*, pp. 843–854, Dec. 1979.
- [19] I. J. Cox and S. L. Hingorani, “An efficient implementation of reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking,” *IEEE Trans. PAMI*, vol. 18, feb 1996.
- [20] T. Fortnam, Y. Bar-Shalom, and M. Scheffe, “Multi-target tracking using joint probabilistic data association,” in *19th IEEE Conf. on Decision and Control*, (Albuquerque, NM), December 1980.
- [21] L. Chen, M. Cetin, and A. S. Willsky, “Distributed data association for multi-target tracking in sensor networks,” in *Proc. International Conference on Information Fusion*, (Philadelphia, Pennsylvania), July 2005.

- [22] S. Oh, S. Russell, and S. Sastry, “Markov chain monte carlo data association for general multiple-target tracking problems,” in *43rd IEEE Conf. on Decision and Control*, (Paradise Island, Bahamas), December 2004.
- [23] S. Oh, I. Hwang, K. Roy, and S. Sastry, “A fully automated distributed multiple-target tracking and identity management algorithm,” in *Proc. AIAA Guidance, Navigation, and Control Conference*, Aug. 2005.
- [24] J. Shin, L. Guibas, and F. Zhao, “A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks,” in *Proc. of 2nd workshop on Information Processing in Sensor Networks (IPSN)*, Apr. 2003.
- [25] I. Hwang, K. Roy, H. Balakrishnan, and C. Tomlin, “A distributed multiple-target identity management algorithm in sensor networks,” in *Proc. 43rd IEEE Conf. on Decision and Control*, (Bahamas), Dec. 2004.
- [26] Q. Fang, J. Liu, and L. Guibas, “Maintain connectivity of roaming agents in sensor networks,” in *the Workshop on Information Processing in Sensor Networks (IPSN’04)*, Berkeley, CA, 2004.
- [27] A. El-Fallah, A. Zatezalo, R. Mahler, R. K. Mehra, and M. Alford, “Advancements in situation assessment sensor management,” in *Signal Processing, Sensor Fusion, and Target Recognition XV* (I. Kadar, ed.), vol. 6235, SPIE, 2006.
- [28] G. Mainland, D. C. Parkes, and M. Welsh, “Decentralized, adaptive resource allocation for sensor networks,” in *2nd USENIX/ACM Symposium on Networked SYstems Design and Implementation (NSDI)*, 2005.
- [29] F. R. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, 2001.
- [30] M. Chu, J. Reich, and F. Zhao, “Distributed attention in large scale video sensor networks,” *IEE Seminar Digests*, vol. 2004, no. 10426, pp. 61–65, 2004.