Collaborative Signal and Information Processing: An Information Directed Approach

Feng Zhao, Jie Liu, Juan Liu, Leonidas Guibas, and James Reich

Abstract— **This article describes information-based approaches to processing and organizing spatially distributed, multi-modal sensor data in a sensor network. Energy constrained networked sensing systems must rely on collaborative signal and information processing (CSIP) to dynamically allocate resources, maintain multiple sensing foci, and attend to new stimuli of interest, all based on task requirements and resource constraints. Target tracking is an essential capability for sensor networks and is used as a canonical problem for studying information organization problems in CSIP. After formulating a CSIP tracking problem in a distributed constrained optimization framework, the paper describes IDSQ and other techniques for tracking individual targets as well as combinatorial tracking problems such as counting targets. Results from simulations and experimental implementations have demonstrated that these information based approaches are scalable and make efficient use of scarce sensing and communication resources.**

Keywords— **Sensor networks, distributed sensing, collaborative signal and information processing, information utility, constrained optimization, target tracking.**

Category: Collaborative Signal Information processing, Target Classification and Tracking

I. Sensor Network Applications, Constraints, and Challenges

Networked sensing offers unique advantages over traditional centralized approaches. Dense networks of distributed networked sensors can improve perceived signal-to-noise ratio (SNR) by decreasing average distances from sensor to target. Increased energy efficiency in communications is enabled by the multi-hop topology of the network [22]. Moreover, additional relevant information from other sensors can be aggregated during this multi-hop transmission through in-network processing [13]. But perhaps the greatest advantages of networked sensing are in improved robustness and scalability. A decentralized sensing system is inherently more robust against individual sensor node or link failures, because of redundancy in the network. Decentralized algorithms are also far more scalable in practical deployment, and may be the only way to achieve the large scales needed for some applications.

A sensor network is designed to perform a set of high-level information processing tasks such as detection, tracking, or classification. Measures of performance for these tasks are well defined, including detection, false alarms or misses, classification errors, and track quality. Commercial and military applications include environmental monitoring (e.g., traffic, habitat, security), industrial sensing and diagnostics (e.g., factory, appliances), infrastructure protection (e.g., power grid, water distributions), and battlefield awareness (e.g., multi-target tracking).

Unlike a centralized system, however, a sensor network is subject to a unique set of resource constraints such as limited on-board battery power and limited network communication bandwidth. In a typical sensor network, each sensor node operates untethered and has a microprocessor and limited amount of memory for signal processing and task scheduling. Each node also is equipped with one or more of acoustic microphone arrays, video or still cameras, IR, seismic, or magnetic sensing devices. Each sensor node communicates wirelessly with a small number of local nodes within the radio communication range.

The current generation of wireless sensor hardware ranges from shoe-box sized Sensoria WINS NG sensors [20] with an SH-4 microprocessor to matchbox sized Berkeley motes with an 8-bit microcontroller [12]. It is well-known that communicating one bit over the wireless medium consumes far more energy than processing the bit. For the Sensoria sensors and Berkeley motes, the ratio of energy consumption for communication and computation is in the range of 1,000–10,000. Despite the advances in silicon fabrication technologies, wireless communication will continue to dominate the energy consumption of embedded networked systems for the foreseeable future [8]. Thus, minimizing the amount and range of communication as much as possible, for example, through local collaboration, data compression, or invoking only the nodes that are relevant to a given task, can significantly prolong the lifetime of a sensor network and leave nodes free to support multi-user operations.

Traditional signal processing approaches have focused on optimizing estimation quality for a fixed set of available resources. However, for power-limited and multi-user decentralized systems, it becomes critical to carefully select the embedded sensor nodes that participate in the sensor collaboration, balancing the information contribution of each against its resource consumption or potential utility for other users. This approach is especially important in dense networks, where many measurements may be highly redundant, and communication throughput severely limited. We use the term "collaborative signal and information processing" (CSIP) to refer to signal and information processing problems dominated by this issue of selecting embedded

This work is supported in part by the Defense Advanced Research Projects Agency (DARPA) under contract number F30602-00-C-0139 through the Sensor Information Technology Program. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

Feng Zhao, Jie Liu, Juan Liu, and James Reich are with Palo Alto Research Center (PARC), 3333 Coyote Hill Road, Palo Alto, CA 94304, USA (email: {zhao,jieliu,jjliu,jreich}@parc.com). Leonidas Guibas is with Computer Science Department, Stanford University, Stanford, CA 94305, USA (email: guibas@cs.stanford.edu).

sensors to participate in estimation.

This paper uses tracking as a representative problem to expose the key issues for CSIP — how to dynamically determine what needs to be sensed, who should sense, how often the information must be communicated, and to whom. The rest of the paper is organized as follows. Section II will introduce the tracking problem and present a set of design considerations for CSIP applications. Sections III and IV will analyze a range of tracking problems that differ in the nature of the information being extracted, and describe and compare several recent contributions that adopted information based approaches. Section V will discuss future directions for CSIP research.

II. Tracking as a Canonical Problem for CSIP

Tracking is an essential capability in many sensor network applications, and is an excellent vehicle to study information organization problems in CSIP. It is especially useful for illustrating a central problem of CSIP: dynamically defining and forming sensor groups based on task requirements and resource availability.

From a sensing and information processing point of view, we define a sensor network as a tuple, $Sn = \langle V, E, P_V, P_E \rangle$. V and E specify a network graph, with its nodes V, and link connectivity $E \subseteq V \times V$. P_V is a set of functions which characterizes the properties of each node in V, including its location, computational capability, sensing modality, sensor output type, energy reserve, and so on. Possible sensing modalities includes acoustic, seismic, magnetic, IR, temperature, or light. Possible output types include information about signal amplitude, source direction-of-arrival (DOA), target range, or target classification label. Similarly, P_E specifies properties for each link such as link capacity and quality.

A tracking task can be formulated as a constrained optimization problem $Tr = \langle Sn, Tg, Sm, Q, O, C \rangle$. Sn is the sensor network specified above. Tg is a set of targets, specifying for each target the location, shape (if not a point source), and signal source type. Sm is a signal model for how the target signals propagate and attenuate in the physical medium. For example, a possible power attenuation model for an acoustic signal is the inverse distance squared model. Q is a set of user queries, specifying query instances and query entry points into the network. A sample query is "Count the number of targets in region R". O is an objective function, defined by task requirements. For example, for a target localization task, the objective function could be the localization accuracy, expressed as the trace of the covariance matrix for the position estimate. $C = \{C_1, C_2, ..., \}$ specifies a set of constraints. An example is localizing an object within a certain amount of time and using no more than a certain quantity of energy. The constrained optimization finds a set of feasible sensing and communication solutions for the problem that satisfy the given set of constraints. For example, a solution to the localization problem above could be a set of sensor nodes on a path that gathers and combines data and routes the result back to the querying node.

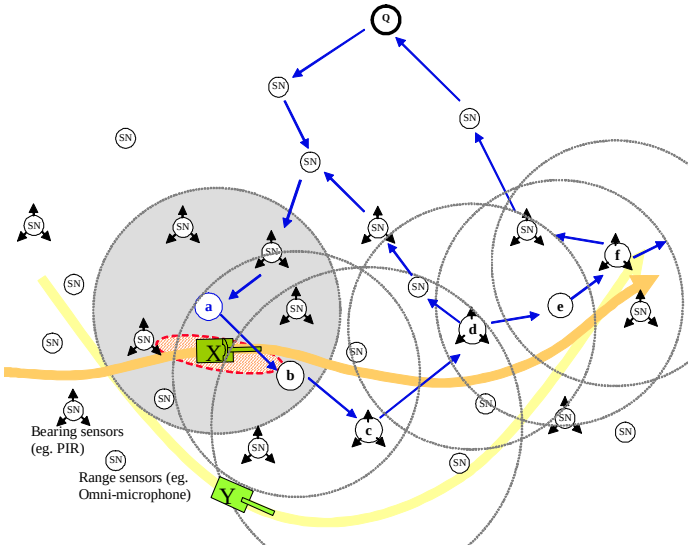

Fig. 1. A tracking scenario, showing two moving targets, X and Y, in a field of sensors. Large circles represent the range of radio communication from each node.

In wireless sensor networks, some of the information defining the objective function and/or constraints is only available at run time. Furthermore, the optimization problem may have to be solved in a decentralized way. In addition, anytime algorithms are desirable because constraints and resource availability may change dynamically.

A. A tracking scenario

We use the following tracking scenario (Figure 1) to bring out key CSIP issues. As a target X moves from left to right, a number of activities are initiated in the network:

1. Discovery: Node a detects X and initializes tracking.
2. Query processing: A user query Q enters the network and is routed towards regions of interest, in this case, the region around node a. It should be noted that other types of queries, such as long-running queries that dwell in a network over a period of time are also possible.
3. Collaborative Processing: Node a estimates the target location, possibly with help from neighboring nodes.
4. Communication: Node a may hand off data to node b, b to c, etc.
5. Reporting: Node d or f summarizes track data and sends it back to the querying node.

Let's now assume another target, Y, enters the region around the same time. The network will have to handle multiple tasks in order to track both targets simultaneously. When the two targets move close to each other, the problem of properly associating a measurement to a target track, the so-called *data association problem*, becomes tricky. In addition, collaborative sensor groups, as defined earlier, must be selected carefully since multiple groups might need to share the same physical hardware [18].

This tracking scenario raises a number of fundamental information processing problems in distributed information discovery, representation, communication, storage,

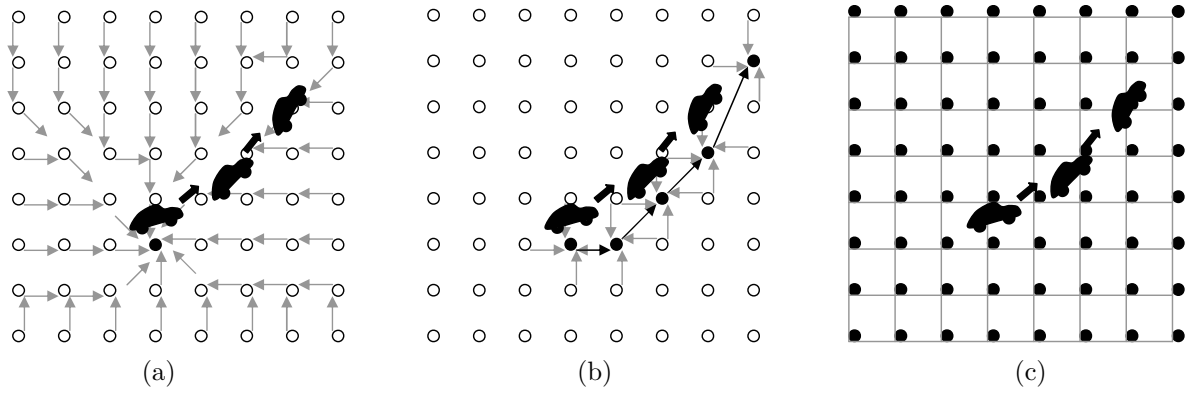

Fig. 2. Storage and communication of target state information in a networked distributed tracker. Circles on the grid represent sensor nodes, and some of the nodes, denoted by solid circles, store target state information. Thin, faded arrows or lines denote communication paths among the neighbor nodes. Thin, dark arrows denote sensor hand-offs. A target moves through the sensor field, indicated by thick arrows. (a) A fixed single leader node has the target state. (b) A succession of leader nodes are selected according to information such as vehicle movement. (c) Every node in the network stores and updates target state information.

and querying; in collaborative processing, the issues of target detection, localization, tracking, and sensor tasking and control; in networking, the issues of data naming, aggregation, and routing; in databases, the issues of data abstraction and query optimization; in human-computer interface, the issues of data browsing, search, and visualization; in software services, the issues of network initialization and discovery, time and location services, fault management, and security. In the rest of the paper, we will focus on the collaborative processing aspects and touch other issues only as necessary.

A common task for a sensor network is to gather information from the environment. Doing this under the resource constraints of a sensor network may require data-centric routing and aggregation techniques which differ considerably from TCP/IP end-to-end communication. Consequently, the research community has been searching for the right "sensor net stack" that can provide suitable abstractions over networking and hardware resources. While defining a unifying architecture for sensor networks is still an open problem, we believe a key element of such an architecture is the *principled interaction between the application and networking layers*. For example, Sec. III will describe an approach that expresses application requirements as a set of information and cost constraints so that an *ad hoc* networking layer using, for example, the diffusion routing protocol [13], can effectively support the application.

B. Design desiderata in distributed tracking

In essence, a tracking system attempts to recover the state of a target (or targets) from observations. Informally, we refer to the information about the target state distilled from measurement data as a belief or belief state. An example is the posterior probability distribution of target state, as discussed in Sec. III. As more observation data are available, the belief may be refined and updated.

In sensor networks, the belief state can be stored centrally at a fixed node, at a sequence of nodes through successive hand-offs, or at a set of nodes concurrently. In the first case (Figure 2(a)), a fixed node is designated to receive measurements from other relevant sensors through communication. This simpler tracker design is obtained at the cost of potentially excessive communication and reduced robustness to node failure. It is feasible only for tracking nearly stationary targets, and is in general neither efficient nor scalable.

In the second case (Figure 2(b)), the belief is stored at a node called the leader node, which collects data from nearby, relevant sensors. As the phenomenon of interest moves or environmental conditions vary, the leadership may change hands among sensor nodes. Since the changes in physical conditions are often continuous in nature, these handoffs often occur within a local geographic neighborhood. This moving leader design localizes communication, reducing overall communication and increasing the lifetime of the network. Robustness of this method may suffer from potential leader node attrition, but this can be mitigated by maintaining copies of the belief in nearby nodes and detecting and responding to leader failure. The key research challenge for this design is to define an effective selection criterion for sensor leaders, to be addressed in Sec. III.

Finally, the belief state can be completely distributed across multiple sensor nodes (Figure 2(c)). The inference from observation data is accomplished nodewise, thus localizing the communication. This is attractive from the robustness point of view. The major design challenge is to efficiently infer global properties about targets, some of which may be discrete and abstract, from partial, local information, and to maintain information consistency across multiple nodes. Sec. IV addresses the challenge. Many issues about leaderless distributed trackers are still open and deserve much attention from the research community.

III. IDSQ: a CSIP approach to target tracking

Distributed tracking is a very active field, and it is beyond the scope of this paper to provide a comprehensive survey. Instead, we will focus on the information processing aspect of the tracking problems, answering questions such as what information is collected by the sensors, how that information is aggregated in the network, and

what high-level user queries are answered. This section describes information-driven sensor query (IDSQ), a set of information- based approaches to tracking individual targets, and discusses major issues in designing CSIP solutions. Next, Sec. IV presents approaches to other tracking problems, where the focus is more on uncovering abstract and discrete target properties, such as target density, rather than just their locations.

A. Tracking individual targets

The basic task of tracking a moving target in a sensor field is to determine and report the underlying target state $\mathbf{x^{(t)}}$, such as its position and velocity, based on the sensor measurements up to time t, denoted as $\overline{\mathbf{z}^{(t)}} = \{\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(t)}\}$. Many approaches have been developed over the last half century, including Kalman filters, which assume a Gaussian observation model and linear state dynamics, and, more generally, sequential Bayesian filtering, which computes the posterior belief at time $t+1$ based on the new measurement $\mathbf{z}^{(t+1)}$ and the belief $p(\mathbf{x}^{(t)}|\overline{\mathbf{z}^{(t)}})$ inherited from time t:

$$p(\mathbf{x}^{(t+1)}|\overline{\mathbf{z}^{(t+1)}}) \propto p(\mathbf{z}^{(t+1)}|\mathbf{x}^{(t+1)}) \cdot \int p(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) \cdot p(\mathbf{x}^{(t)}|\overline{\mathbf{z}^{(t)}})\, d\mathbf{x}^{(t)}.$$

Here $p(\mathbf{z}^{(t+1)}|\mathbf{x}^{(t+1)})$ denotes the observation model, and $p(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$ the state dynamics model. As more data is gathered over time, the belief $p(\mathbf{x}^{(t)}|\overline{\mathbf{z}^{(t)}})$ is successively refined.

Kalman filters and many practical forms of Bayesian filters assume that the measurement noise across multiple sensors is independent, which is not always the case. Algorithms such as Covariance Intersection have been proposed to combine data from sensors with correlated information. While these methods have been successfully implemented in applications, they were primarily designed for centralized platforms. Relatively little consideration was given to the fundamental problems of moving data across sensor nodes in order to combine data and update track information. There was no cost model for communication in the tracker. Furthermore, due to communication delays, sensor data may arrive at a tracking node out of order compared to the original time sequence of the measurements. Kalman or Bayesian filters assume a strict temporal order on the data during the sequential update, and may have to roll back the tracker in order to incorporate "past" measurement, or throw away the data entirely.

For multi-target tracking, methods such as Multiple Hypothesis Tracking (MHT) [23] and Joint Probabilistic Data Association (JPDA) [2] have been proposed. They addressed the key problem of data association, of pairing sensor data with targets, thus creating association hypotheses. MHT forms and maintains multiple association hypotheses. For each hypothesis, it computes the probability that it is correct. On the other hand, JPDA evaluates the association probabilities and combines them to compute the state estimate. Straightforward applications of MHT and JPDA suffer from a combinatorial explosion in data association. Knowledge about targets, environment, and sensors can be exploited to rank and prune hypotheses [7], [21].

B. Information-based approaches

The main idea of information-based approaches is to base sensor collaboration decisions on information content as well as constraints on resource consumption, latency and other costs. Using information utility measures, sensors in a network can exploit the information content of data already received to optimize the utility of future sensing actions, thereby efficiently managing scarce communication and processing resources. Distributed information filter as described in [19] is a global method, requiring each sensor node to communicate its measurement to a central node where estimation and tracking are carried out. In this method, sensing is distributed while tracking is centralized. Directed diffusion routes sensor data in a network to minimize communication distance between data sources and data sinks [9], [13]. This is an interesting way of organizing a network to allow publish-and-subscribe to occur at a very fine grain level. A prediction-based tracking algorithm is described in [3] which uses estimates of target velocity to select which sensors to query. IDSQ [26], [5] formulates the tracking problem as a more general distributed constrained optimization that maximizes information gain of sensors while minimizing communication and resource usage. We describe the main elements of IDSQ here.

Given the current belief state, we wish to incrementally update the belief by incorporating the measurements of other nearby sensors. However, not all available sensors in the network provide useful information that improves the estimate. Furthermore, some information may be redundant. The task is to select an optimal subset and an optimal order of incorporating these measurements into our belief update. Note that in order to avoid prohibitive communication costs, this selection must be done without explicit knowledge of measurements residing at other sensors. The decision must be made solely based upon known characteristics of other sensors, such as their position and sensing modality, and predictions of their contributions, given the current belief.

Figure 3 illustrates the basic idea of optimal sensor selection. The illustration is based upon the assumption that estimation uncertainty can be effectively approximated by a Gaussian distribution, illustrated by uncertainty ellipsoids in the state space. In the figure, the solid ellipsoid indicates the belief state at time t, and the dashed ellipsoids are the incrementally updated belief after incorporating an additional measurement from a sensor, S1 or S2, at the next time step. Although in both cases, S1 and S2, the area of high uncertainty is reduced by 50%, the residual uncertainty of the S2 case is not reduced along the long principal axis of the ellipse. If we were to decide between the two sensors, we might favor case S1 over case S2, based upon the underlying measurement task.

In distributed sensor network systems, we must balance the information contribution of individual sensors against the cost of communicating with them. For example, consider the task of selecting among K sensors with measurements $\{\mathbf{z}_i\}_{i=1}^K$. Given the current belief $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i\in U})$, where $U \subset \{1, \ldots, K\}$ is the subset of sensors whose mea-

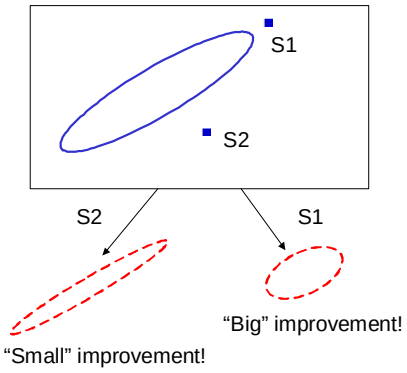

Fig. 3. Sensor selection based on information gain of individual sensor contributions. The information gain is measured by the reduction in the error ellipsoid. In the figure, reduction along the longest axis of the error ellipsoid produces a larger improvement in reducing uncertainty. Sensor placement geometry and sensing modality can be used to compare the possible information gain from each possible sensor selection, S1 or S2.

surement has already been incorporated, the task is to choose which sensor to query among the remaining unincorporated set $A = \{1, \cdots, K\} \setminus U$. For this task, an objective function as a mixture of information and cost is designed in [5]:

$$O(p(\mathbf{x}|\overline{\mathbf{z}_j^{(t)}})) = \alpha\phi(p(\mathbf{x}|\overline{\mathbf{z}_{j-1}^{(t)}}, \mathbf{z}_j^{(t)})) - (1-\alpha)\psi(\mathbf{z}_j^{(t)}). \quad (1)$$

Here ϕ measures the information utility of incorporating the measurement $\mathbf{z}_j^{(t)}$ from sensor j, ψ is the cost of communication and other resources, and α is the relative weighting of the utility and cost. With this objective function, the sensor selection criterion takes the form:

$$\hat{j} = \arg\max_{j \in A} \; O\left(p\left(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\}\right)\right) \; . \quad (2)$$

This strategy selects the best sensor given the current state $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U})$. A less greedy algorithm has been proposed in [17], extending the sensor selection over a finite look-ahead horizon.

Metrics of information utility ϕ and cost ψ may take various forms, depending on the application and assumptions [26]. For example, [5] considers the query routing problem: assuming a query has entered from a fixed node, denoted by "?" in Figure 4, the task is to route the query to the target vicinity, collect information along an optimal path, and report back to the querying node. Assuming the belief state is well- approximated by a Gaussian distribution, the usefulness of the sensor data (in this case, range data), ϕ, is measured by how close the sensor is to the mean of the belief state under a Mahalanobis metric, assuming that close-by sensors provide more discriminating information. The cost ψ is given here by the squared Euclidean distance from the sensor to the current leader, a simplified model of the energy expense of radio transmission for some environments. The optimal path results from the tradeoff between these two terms. Figure 4 plots such a sample path. Note that the belief is updated incrementally along

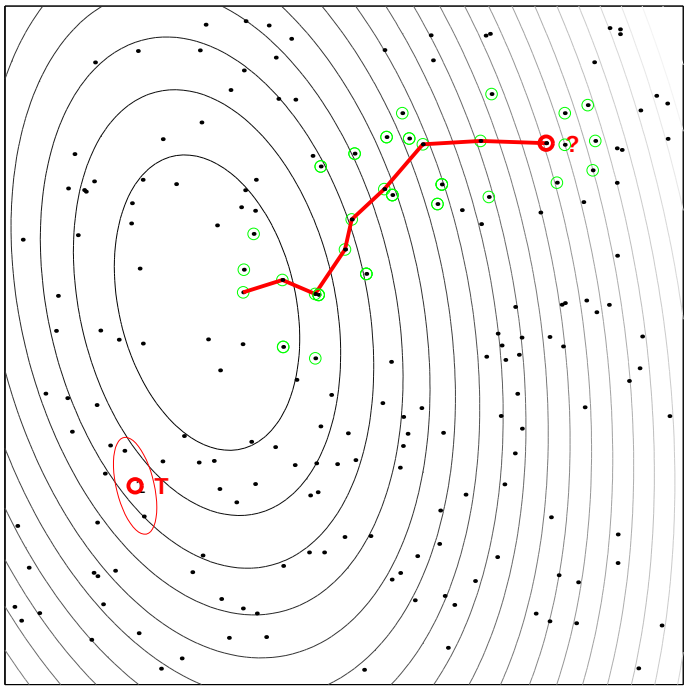

Fig. 4. Sensor querying and data routing by optimizing an objective function of information gain and communication cost, whose isocontours are shown as the set of concentric ellipses. The circled dots are the sensors being queried for data along the querying path. "T" represents the target position, and "?" denotes the position of the query origin.

the information collection path. The ellipses in Figure 4 show a snapshot of the objective function that an active leader node locally evaluates at a given time step.

For multi-modal non-Gaussian distributions, a mutual information-based sensor selection criterion has been developed and successfully tested on real data [16]. The problem is as follows: assuming that a leader node holds the current belief $p(x^{(t)}|\overline{\mathbf{z}^{(t)}})$, and the cost to query any sensor in its neighborhood $\mathcal{N}$ is identical (e.g. over a wired network or using a fixed power level radio), the leader selects from $\mathcal{N}$ the most informative sensor to track the moving target. In this scenario, the selection criterion (2) takes the form:

$$\hat{j}_{IDSQ} = \arg\max_{j \in \mathcal{N}} I(X^{(t+1)}; Z_j^{(t+1)} | \overline{Z^{(t)}} = \overline{\mathbf{z}^{(t)}}), \quad (3)$$

where $I(\cdot;\cdot)$ measures the mutual information in bits between two random variables. Essentially, this criterion selects a sensor whose measurement $\mathbf{z}_j^{(t+1)}$, combined with the current measurement history $\overline{\mathbf{z}^{(t)}}$, would provide the greatest amount of information about the target location $\mathbf{x}^{(t+1)}$. The mutual information can be interpreted as Kullback-Leibler divergence between the belief after and before applying the new measurement $\mathbf{z}_j^{(t+1)}$. Therefore, this criterion favors the sensor which on average gives the greatest change to the current belief.

To analyze the performance of the IDSQ tracker, we measure how the tracking error varies with sensor density through simulation. Figure 5 shows that as the sensor den-

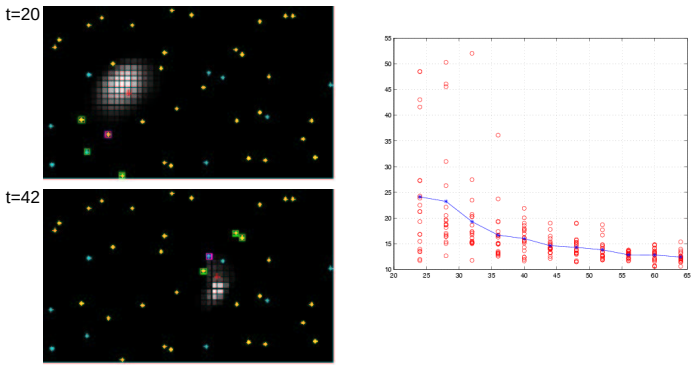

Fig. 5. Experimental results (right figure) show how the tracking error (vertical axis), defined as the mean error of estimated target positions, varies with the sensor density (horizontal axis), defined as the number of sensors in the sensor field. The left figure shows snapshots of a belief "cloud" — the probability density function of the location estimate — for different local sensor densities.

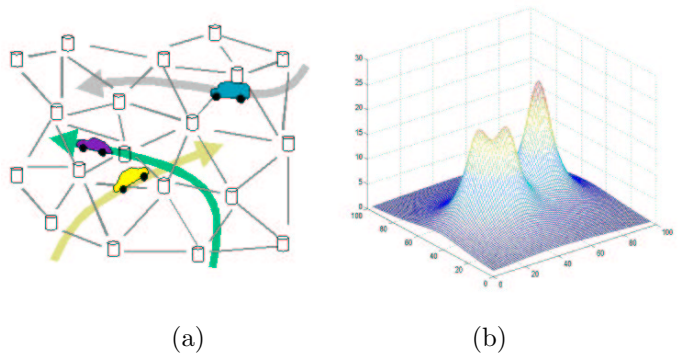

Fig. 6. Target counting scenario, showing three targets in a sensor field (a). The goal is to count and report the number of distinct targets. With the signal field plotted in (b), the target counting becomes a peak counting problem.

sity increases, tracking error, expressed as the mean error of the location estimate, decreases, as one would expect, and tends to a floor dominated by sensor noise. This indicates that there is a maximum density beyond which using more sensors gains very little in tracking accuracy.

The IDSQ tracker has been successfully tested in a DARPA tracking experiment at 29 Palms, November 2001. In the experiment, 21 Sensoria WINS NG wireless sensors were used to collect acoustic data from moving vehicles. Details of the results can be found in [16].

IV. Combinatorial Tracking Problems

The discussion of tracking so far has focused on localizing targets over time. In many applications, however, the phenomenon of interest may not be the exact locations of individual objects, but global properties regarding a collection of objects, for example, the number of targets, their regions of influence, or their boundaries. The information to be extracted in this case is more discrete and abstract, and may be used to answer high-level queries about the world-state or to make strategic decisions about actions to take.

An expensive way to compute such global class properties of objects is to locate and identify each object in the collection, determine its individual properties, and combine the individual information to form the global answer, such as the total number of objects in the collection. However, in many cases, these class properties can be inferred without accurate localization or identification of all the objects in question. For example, it may be possible to focus on attributes or relations that can be directly sensed by the sensors. This may both make the tracking results more robust to noise and may simplify the algorithms to the point where they can be implemented on less powerful sensor nodes. We call these approaches *combinatorial tracking*.

A. Counting the number of targets

Target counting is an attempt to keep track of the number of distinct targets in a sensor field, even as they move, cross-over, merge, or split. It is representative of a class of applications that need to monitor intensity of activities in an area. To describe the problem, let us consider counting multiple targets in a two-dimensional sensor field, as shown in Figure 6. We assume that targets are point source acoustic signals and can be stationary or moving at any time, independent of the state of other targets. Sensors measure acoustic power and are time synchronized to a global clock. We assume that signals from two targets simply add at a receiving sensor, which is reasonable for non-coherent interference between acoustic sources.

The task here is to determine the number of targets in the region. One way to solve the problem is to compute an initial count and then update the count as targets move, enter, or leave the region. Here, we describe a leader-based counting approach, where a sensor leader is elected for each distinct target. A leader is initialized when a target moves into the field. As the target moves, the leadership may switch between sensor nodes to reflect the state change. When a target moves out of the region, the corresponding leader node is deactivated. Note here the leader election does not rely on accurate target localization, as will be discussed later. The target count is obtained by noting the number of active leader nodes in the network (and the number of targets each is responsible for). Here we will focus on the leader election process, omitting details of signal and query processing.

Since the sensors in the network sense only signal energy, we need to examine the spatial characteristics of target signals when multiple targets are in close proximity to each other. In Figure 6(b), the 3-D surface shown represents total target signal energy. Three targets are plotted, with two targets near each other and one target well separated from the rest of the group.

There are several interesting observations to make here:

- Call the set of sensors who can "hear" a target the target influence area. When targets' influence areas are well-separated, target counting can be considered as a clustering and a cluster leader election problem. Otherwise, it becomes a peak counting problem.

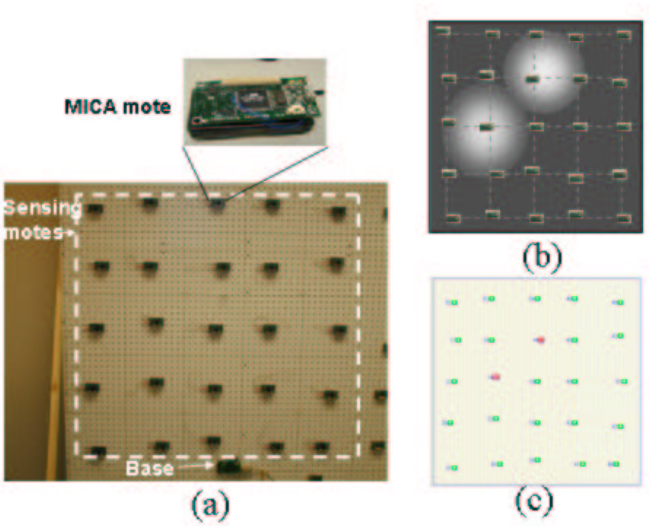

Fig. 7. Target counting application implemented on Berkeley motes. (a) 25 MICA motes with light sensors are placed on a perturbed grid in a dark room. (b) Two light blobs emulating $1/r^2$ signal attenuation are projected onto the mote board. (c) The leader of each collaboration group sends its location back to a base station GUI.

- The target signal propagation model has a large impact on target "resolution". The faster the signal attenuates with distance from the source, the easier targets it is to discern targets from neighboring targets based on the energy of signals they emit.
- Sensor spacing is also critical in obtaining correct target count. Sensor density has to be sufficient to capture the peaks and valleys of the underlying energy field, yet very densely-packed sensors are often redundant, wasting resources.

A decentralized algorithm was introduced for the target counting task [10]. This algorithm forms equivalence classes among sensors and elects a leader node for each class based on the relative power detected at each sensor, and counts the number of such leaders. The algorithm comprises a decision predicate P which, for each node i, tests if it should participate in an equivalence class and a message exchange schema M about how the predicate P is applied to nodes. A node determines whether it belongs to an equivalence class based on the result of applying the predicate to the data of the node as well as information from other nearby nodes. Equivalence classes are formed when the process converges. This protocol finds equivalence classes even when multiple targets interfere.

This leader election protocol is very powerful yet lightweight enough to be implemented on sensor nodes such as the Berkeley motes. Figure 7 shows an experiment consists of 25 MICA motes with light sensors. The entire application, including code for collaborative leader election and multi-hop communication to send the leader information back to the base station, takes about 10K bytes memory space on a mote.

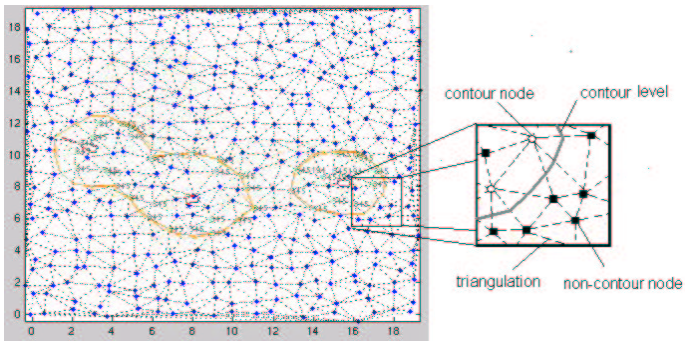

Fig. 8. Simulation result showing contours for three point targets in a sensor field. The contours are constructed using a distributed marching squares like algorithm, and is updated as targets move.

B. Contour tracking

Contour tracking is another example of finding the influence regions of targets without locating them. For a given signal strength, the tracking results are a set of contours, each of which contains one or more targets.

As in the target counting scenario, let us consider a 2-D sensor field and point source targets. One way of determining the contours is by building a mesh over distributed sensor nodes via a Delaunay triangulation or a similar algorithm. The triangulation can be computed offline when setting up the network. Nodes that are connected by an edge of a triangle are called direct neighbors. Given a measurement threshold σ, which defines a σ-contour, a node is called a contour node if it has a sensor reading above σ and at least one of its direct neighbors has a sensor reading below σ. For a sufficiently smooth contour and dense sensor network, a contour can be assumed to intersect an edge only once, and an triangle at exactly two edges, as shown in Figure 8. By following this observation, we can traverse the contour by "walking" along the contour nodes. Again, purely local algorithms exist to maintain these contours as the targets move.

C. Shadow edge tracking

Contour tracking can be viewed as a way to determine the boundary of a group of targets. In an extreme case, the group of targets can be a continuum over space, where no single sensor alone can determine the global information from its local measurement. An example of this is to determine and track the boundary of a large object moving in a sensor field, where each sensor only "sees" a portion of the object. One such application is tracking a moving chemical plume over an extended area using airborne and ground chemical sensors.

We assume the boundary of the object is a polygon made of line segments. Our approach is to convert the problem of estimating and tracking a nonlocal (possibly very long) line segment into a local problem using a dual-space transformation [15]. Just as a Fourier transform maps a global property of a signal, such as periodicity in the time domain, to a local feature in the frequency domain, the dual-space transform maps a line in the primal space into a point in

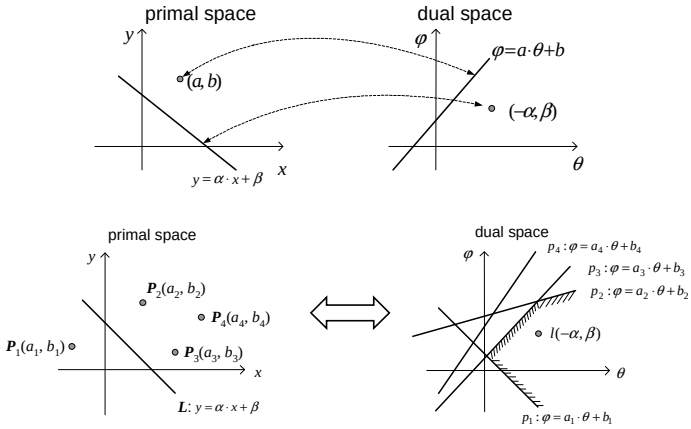

Fig. 9. Primal-dual transformation, a one-one mapping where a point maps to a line and a line maps to point (upper figure). The image of a half-place shadow edge in the dual space is a point located in a cell formed by the duals of the sensor nodes (lower figure).

the dual space, and vice versa (Figure 9). Using a primal-dual transformation, each edge of a polygonal object can be tracked as a point in the dual space. A tracking algorithm has been developed based on the dual-space analysis and implemented on the Berkeley motes [15]. A key feature of this algorithm is that it allows us to put to sleep all sensor nodes except those in the vicinity of the object boundary, yielding significant energy savings.

Tracking relations among a set of objects is another form of global, discrete analysis of a collection of objects, as described in [11]. An example determining whether a friendly vehicle is surrounded by a number of enemy tanks. Just as in the target counting problem, the "am I surrounded" relation can be resolved without having to solve the local problems of localizing all individual objects first.

V. Discussions

We have used the tracking problem as a vehicle to discuss sensor network CSIP design. We have focused on the estimation and tracking aspects and skipped over other important details, such as target detection and classification, for space reasons.

Detection is an important capability for a sensor network, as a tracker must rely on detection to initialize itself as new events emerge [25], [14]. Traditional detection methods focused on minimizing false alarms or the miss rate. In a distributed sensor network, the more challenging problem for detection is the proper allocation of sensing and communication resources to multiple competing detection tasks spawned by emerging stimuli. This dynamic allocation and focusing of resources in response to external events is somewhat analogous to attentional mechanisms in human vision systems, and clearly a future research direction. More research should also be directed to the information architecture of distributed detection and tracking, and address the problems of "information double-counting" and data association in a distributed network [18], [24].

Optimizing resources for a given task, as for example in IDSQ, relies on accurate models of information gain and cost. To apply the information-driven approach to tracking problems involving other sensing modalities or to problems other than tracking, we will need to generalize our models for sensing and estimation quality as well as our models of the tradeoff between resource use and quality. For example, what is the expected information gain per unit energy consumption in a network? One must make assumptions about the network, stimuli, and tasks in order to build such models. Another interesting problem for future research is to consider routing and sensing simultaneously and optimize for the overall gain of information.

We have not yet touched upon the programming issues in sensor networks. The complexity of the applications, the collaborative nature of the algorithms, and the plurality and diversity of resource constraints demand novel ways to construct, configure, test, and debug the system, especially the software. This is more challenging than traditional collection-based computation in parallel processing research because sensor group management is typically dynamic and driven by physical events. In addition, the existing development and optimization techniques for embedded software are largely at the assembly level and do not scale to collaborative algorithms for large-scale distributed sensor networks. We need high-level system organizational principles, programming models, data structures, and processing primitives to express and reason about system properties, physical data, and their aggregation and abstraction, without losing relevant physical and resource constraints.

A possible programming methodology for distributed embedded sensing systems is shown in Figure 10. Given a specification at a collaborative behavioral level, software tools automatically generate the interactions of algorithm components, and map them onto the physical hardware of sensor networks.

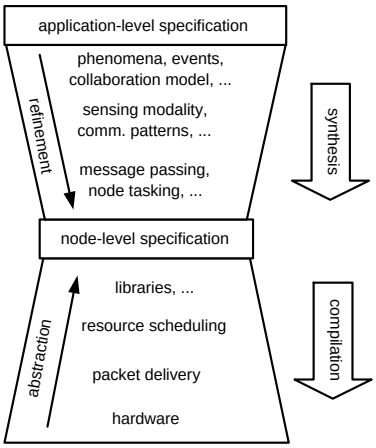

Fig. 10. A programming methodology for deeply embedded systems.

At the top level, the programming model should be expressive enough to describe application level concerns: physical phenomena to be sensed, user interaction, and collaborative processing algorithms, without the need to manage node-level interactions. The programming model may be domain specific. For example, SAL [27] is a language

for expressing and reasoning about geometries of physical data in distributed sensing and control applications; various biologically inspired computational models [1], [6] study how complex collaborative behaviors can be built from simple components. The programming model should be structural enough to allow synthesis algorithms to exploit commonly occurring patterns and generate efficient code. TinyGALS [4] is an example of synthesizable programming models for event-driven embedded software.

Automated software synthesis is a critical step in achieving the scalability of sensor network programming. Hardware-oriented concerns such as timing and location may be introduced gradually by refinement and configuration processes. The final outputs of software synthesis are operational code for each node, typically in forms of imperative languages, from which the more classical operating system, networking, and compiler technologies can be applied to produce executables. The libraries supporting node-level specifications need to abstract away hardware idiosyncrasy across different platforms, but still expose enough low-level features for applications to take advantage of.

VI. Conclusion

This paper has focused on the collaborative signal and information processing (CSIP) issues in designing and analyzing sensor network applications. In particular, we have used tracking as a canonical problem to expose important constraints in designing, scaling, and deploying these sensor networks, and described approaches to several tracking problems that are at progressively higher levels with respect to the nature of information being extracted.

From the discussions, it is clear that for resource-limited sensor networks one must take a more holistic approach and break the traditional barrier between the application and networking layers. The challenge is to define the constraints from an application in a general way so that the networking layers can exploit, and vice versa. An important contribution of the approaches described in this paper is the formulation of application requirements and network resources as a set of generic constraints so that target tracking and data routing can be jointly optimized.

Acknowledgment

The algorithm and experiment for the targe counting problem were designed and carried out in collaboration with Qing Fang, Judy Liebman, and Elaine Cheong. The contour tracking algorithm and simulation were jointly developed with Krishnan Eswaran.

Patrich Cheung designed, prototyped, and calibrated the PARC sensor network testbeds and supported the laboratory and field experiments for the algorithms and software described in this paper.

References

[1] H. Abelson et al, "Amorphous Computing." *Communications of the ACM* 43(5):74-82, May 2001.

[2] Y. Bar-Shalom and X.R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS Publishing, Storrs, CT, 1995.

[3] R.R. Brooks, C. Griffin, and D.S. Friedlander, "Self-organized distributed sensor network entity tracking." *Int'l J. of High-Performance Computing Applications*, 16(3), 2002.

[4] E. Cheong, J. Liebman, J. Liu, and F. Zhao, "TinyGALS: A Programming Model for Event-Driven Embedded Systems." 18^{th} *ACM Symposium on Applied Computing*, March, 2003, to appear.

[5] M. Chu, H. Haussecker, F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks." *Int'l J. of High-Performance Computing Applications*, 16(3):90-110, 2002.

[6] C. Calude, M.J. Dinneen, and F. Peper (Eds.), *Unconventional Models of Computation.* LNCS 2509, Springer, 2002.

[7] 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. on PAMI*, Vol 18., No. 2, pp. 138-150, Feb. 1996.

[8] L. Doherty, B.A. Warneke, B.E. Boser, K.S.J. Pister, "Energy and Performance Considerations for Smart Dust," *International Journal of Parallel Distributed Systems and Networks*, 4(3), pp. 121-133, 2001.

[9] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next Century Challenges: scalable coordination in sensor networks." In *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99)*, Seattle, Washington, August 1999.

[10] Q. Fang, F. Zhao, and L. Guibas, Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation. ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), 2003.

[11] L. Guibas, "Sensing, tracking, and reasoning with relations." *IEEE Signal Processing Magazine*, 19(2):73-85, March, 2002.

[12] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, Kristofer S. J. Pister. "System Architecture Directions for Networked Sensors." ASPLOS 2000.

[13] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks." *Proc. ACM MobiCOM*, Boston, August 2000.

[14] Dan Li, Kerry Wong, Yu Hen Hu, and Akbar Sayeed. "Detection, Classification and Tracking of Targets in Distributed Sensor Networks", *IEEE Signal Processing Magazine*, 19(2), March, 2002.

[15] J. Liu, P. Cheung, L. Guibas, and F. Zhao, "A dual-space approach to tracking and sensor management in wireless sensor networks." In *Proc. 1st ACM Int'l Workshop on Wireless Sensor Networks and Applications*, pp. 131-139, Atlanta, April 2002.

[16] J.J. Liu, J. Reich, and F. Zhao, "Collaborative In-Network Processing for Target Tracking." *J. of Applied Signal Processing*, April 2003.

[17] J.J. Liu, D. Petrovic, and F. Zhao, "Multi-Step Information-Directed Sensor Querying in Distributed Sensor Networks," *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2003.

[18] J.J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao, "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications." *Proceedings of 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, April, 2003.

[19] J. Manyika and H. Durrant-Whyte, *Data fusion and sensor management: a decentralized information-theoretic approach.* Ellis Horwood, 1994.

[20] W. M. Merrill, K. Sohrabi, L. Girod, J. Elson, F. Newberg, and W. Kaiser, "Open Standard Development Platforms for Distributed Sensor Networks." *Proceedings of SPIE, Unattended Ground Sensor Technologies and Applications IV*, AeroSense 2002, Vol. 4743, Orlando, FL, pp. 327-337, April 2-5, 2002.

[21] A.B. Poore, "Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking." *Computational Optimization and Applications*, 3:27-57, 1994.

[22] G.J. Pottie and W.J. Kaiser, "Wireless integrated network sensors." *Comm. ACM*, 43(5):51-58, May 2000.

[23] D.B. Reid. "An Algorithm for Tracking Multiple Targets." *IEEE Trans. on Automatic Control*, 24:6, 1979.

[24] J. Shin, L. Guibas, F. Zhao, "A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-Hoc Sensor Networks." *Proceedings of 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, April, 2003.

[25] R.R. Tenny and N.R. Sandell, Jr., "Detection with Distribute sensors." *IEEE Trans on Aero and Elec. Sys*, v17, p.501-510, 1981.

[26] F. Zhao, J. Shin, J. Reich, "Information-driven dynamic sensor collaboration." *IEEE Signal Processing Magazine*, 19(2):61-72, March, 2002.

[27] F. Zhao, C. Bailey-Kellogg, M. Fromherz, "Physics-Based Encapsulation in Embedded Software for Distributed Sensing and Control Applications." *Proceedings of the IEEE*, 91(1):40-63, Jan. 2003.