Collaborative In-Network Processing for Target Tracking

Juan Liu*, James Reich and Feng Zhao †
Palo Alto Research Center
3333 Coyote Hill Rd.,
Palo Alto, CA 94304
email: {jjliu, jreich, zhao}@parc.com

Submitted: December 21, 2001
Revised: October 4, 2002

Abstract

This paper presents a class of signal processing techniques for collaborative signal processing in ad hoc sensor networks, focusing on a vehicle tracking application. In particular, we study two types of commonly used sensors — acoustic-amplitude sensors for target distance estimation and direction-of-arrival sensors for bearing estimation — and investigate how networks of such sensors can collaborate to extract useful information with minimal resource usage. The information-driven sensor collaboration has several advantages: the tracking is distributed; the network is energy-efficient, and is activated only on a when-needed basis. We demonstrate the effectiveness of the approach to target tracking using both simulation and field data.

Keywords — sensor network, target tracking, distributed processing, Bayesian filtering, beamforming, mutual information

*Contact author.

†This work is partially supported by the DARPA Sensor Information Technology program under Contract F30602-00-C-0139. We acknowledge the significant contributions of Patrick Cheung, Jaewon Shin, and Dan Larner. We are also indebted to Prof. Kung Yao and Joe Chen of UCLA for their advice on using bearing estimation for collaborative signal processing.

1 Introduction

Sensors of various types have already become ubiquitous in modern life, from infrared motion detectors in our light switches to silicon accelerometers in the bumpers of our cars. As the cost of the sensors comes down rapidly due to advances in MEMS fabrication and these sensors increasingly acquire networking and local processing capabilities, new types of software applications become possible, distributed among these everyday devices and performing functions previously impossible for any of the devices independently. Enabling such functionality without overtaxing the resources of the existing devices, especially when these devices are untethered and running on batteries, may require us to rethink some important aspects of how sensing systems are designed.

1.1 Advantages of distributed sensor networks

There are a number of reasons why networked sensors have a significant edge over existing, more centralized sensing platforms.

An ad hoc sensor network can be flexibly deployed in an area where there is no a priori sensing infrastructure. Coverage of a large area is important for tracking events of a significant spatial extent, as in tracking a large number of events simultaneously, or for tracking dynamic events traversing the sensing ranges of many individual sensors, as in tracking a moving vehicle.

In cases of tracking low-observable phenomena, such as a person walking in an obstacle field in an urban environment, or a stealthy military vehicle, the signal-to-noise ratio (SNR) of data collected from a central location may be unacceptable. As sensor density increases, the mean distance from the nearest sensor to a target decreases and the SNR received at the nearest sensor improves.

A large-area sensor network may also activate different parts of the network to process different queries from users, supporting a multi-mode, multi-user operation.

1.2 Sensor network challenges

The design of signal processing applications for the sensor networks involves a number of significant challenges. The primary concern is the limited energy reserve at each node. Combining information from spatially distributed sensor nodes requires processing and communicating sensor data, hence

consumes energy. Second, the network must be able to scale to large number of nodes and be able to track many events.

To address these challenges, the sensor network must blend sensing application with network routing so that the communication is informed by the application needs. Data source selection is key to conserving network resources, managing network traffic, and achieving scalability.

1.3 Collaborative signal processing

Traditional signal processing approaches have focused on optimizing estimation quality for a 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. The data required to choose the appropriate information sources may be dynamic, and may exist solely on sensors already participating in the collaboration. We use the term collaborative signal processing to refer to signal processing problems dominated by this issue of selecting embedded sensors to participate in estimation.

There already exist a number of approaches to collaborative signal processing. Brooks et al. described a prediction based sensor collaboration that uses estimation of target velocities to activate regions of sensors [1]. Our approach builds on an information-driven approach to tracking that exploits information from both residual uncertainties of the estimation as well as vehicle dynamics (e.g. dynamics as in Brooks et al.'s approach). Estrin et al. developed the directed diffusion approach to move sensor data in a network that seeks to minimize communication distance between data sources and data sinks [2]. Their approach has been successfully demonstrated in experiments. Our algorithms build on the directed diffusion so that the network routing is further informed by application-level knowledge about where to send information and where to get useful information.

1.4 Organization of this paper

In this paper, we describe a particular approach to collaborative signal processing. A class of signal processing algorithms will be presented to support the so-called information-driven sensor

collaboration. The application to tracking a maneuvering vehicle will be used as a primary example throughout the discussion. Finally, experimental results from simulation and field data will be presented to validate the approach.

2 Sensor Network and Target Tracking

The ability to track a target is essential in many commercial and military applications. For example, battlefield situational awareness requires an accurate and timely determination of vehicle locations for targeting purposes. Other applications include facility security and highway traffic monitoring. Networked sensors are often ideally suited for target tracking because of their spatial coverage and multiplicity in sensing aspect and modality. Each sensor acquires local, partial, and relatively crude information from its immediate environment; by exploiting the spatial and sensing diversity of a multitude of sensors, the network can arrive at a global estimation by suitably combining the information from the distributed sources.

In this paper, we follow the information-driven sensor querying (IDSQ) framework in which sensors are selectively activated based on their utility and cost [3]. The application focus will be on tracking a moving vehicle through a two-dimensional sensor field. Because of the constraints on sensing range, computation, communication bandwidth, and energy consumption, we consider a leader-based tracking scheme, where at any time instant there is only one sensor active, namely the leader sensor, while the rest of the network is idle. The leader applies a measurement to its prediction of vehicle position and produces a posterior belief about the target location. The updated belief is then passed on to one of the neighboring sensors. The original leader goes back to sleep, and the sensor which receives the belief becomes the new leader, and the process of sensing, estimation, and leader selection repeats. The leader-based scheme has several advantages. Selective sensor activation and communication make the network energy-efficient, of lower probability of detection, and capable of supporting multi-user operation or multi-target tracking.

We use the following notation throughout this paper:

- Superscript t denotes time. We consider discrete time $t \in \mathbb{Z}^+$.
- Subscript $k \in \{1...K\}$ (where applicable) denotes sensor index. K is the total number of sensors in the network.

- The target state at time t is denoted as $x^{(t)}$. Without loss of generality, we consider the tracking application, where $x^{(t)}$ is the location of the moving object in a two–dimensional plane.
- The sensor measurement of sensor at time t is denoted as $z^{(t)}$.
- The measurement history up to time t is denoted as $\overline{z^{(t)}}$, i.e., $\overline{z^{(t)}} = \{z^{(0)}, z^{(1)}, \cdots, z^{(t)}\}$.
- The collection of all sensor measurements at time t are denoted as $\underline{z}^{(t)}$, i.e., $\underline{z}^{(t)} = \{z_1^{(t)}, z_2^{(t)}, \cdots, z_K^{(t)}\}$. This is used only in Sec. 2.2 when our distributed tracking system is compared to a fully centralized system.

2.1 Distributed Bayesian estimation

The goal of tracking is to obtain a good estimate of the target location $x^{(t)}$ from the measurement history $\overline{z^{(t)}}$. For this problem, we use the classic Bayesian approach. We would like our estimate $\hat{x}(\overline{z^{(t)}})$ to be, on average, as close to the true value $x^{(t)}$ as possible according to some measure. That is, the estimate should minimize the average cost

$$\mathcal{E} = E[d(\hat{x}(\overline{z^{(t)}}), x^{(t)})], \tag{1}$$

where $d(\cdot, \cdot)$ is a loss function to measure the estimator performance. For example, $d(\hat{x}, x) = ||\hat{x} - x||^2$ measures the square of l^2 distance between the estimate and its true value. For this loss function, the estimate is

$$\hat{x}_{MMSE}^{(t)} = E[x^{(t)}|\overline{z^{(t)}}] = \int x^{(t)} p(x^{(t)}|\overline{z^{(t)}})\ dx^{(t)}. \tag{2}$$

This estimator is known as the minimum mean-squared error (MMSE) estimator [4]. We informally refer to the current *a posteriori* distribution $p(x^{(t)}|\overline{z^{(t)}})$ as the *belief*. The key issue is how to compute the belief efficiently.

As briefly explained earlier, we use a leader-based tracker to minimize computation and power consumption requirements. At time t, the leader receives a belief state $p(x^{(t)}|\overline{z^{(t)}})$ from the previous leader, and takes a new measurement $z^{(t+1)}$. Assuming the following conditional independence assumptions are satisfied:

- conditioned on $x^{(t+1)}$, the new measurement $z^{(t+1)}$ is independent of the past measurement history $\overline{z^{(t)}}$;

- conditioned on $x^{(t)}$, the new position $x^{(t+1)}$ is independent of $\overline{z^{(t)}}$.

These are standard assumptions in dynamics, and fairly mild in practice. Under these assumptions, based on the new measurement, the leader computes the new belief $p(x^{(t+1)}|\overline{z^{(t+1)}})$ using sequential Bayesian filtering:

$$p(x^{(t+1)}|\overline{z^{(t+1)}}) \propto p(z^{(t+1)}|x^{(t+1)}) \cdot \int p(x^{(t+1)}|x^{(t)}) \cdot p(x^{(t)}|\overline{z^{(t)}})\ dx^{(t)}. \tag{3}$$

Sequential Bayesian filtering includes the traditional Kalman filtering [5] as a special case. While the former is restricted to linear systems and explicitly assumes Gaussian belief states and error models, the latter is suitable for more general discrete time dynamic systems. This is useful in multisensor target tracking, where the sensor models and vehicle dynamics are often non-Gaussian and/or nonlinear, as will discussed in the next section.

In (3), $p(x^{(t)}|\overline{z^{(t)}})$ is the belief inherited from the previous step; $p(z^{(t+1)}|x^{(t+1)})$ is the likelihood of observation given the target location; $p(x^{(t+1)}|x^{(t)})$ is related to vehicle dynamics. For example, if the vehicle is moving at a known velocity v, then $p(x^{(t+1)}|x^{(t)})$ is simply $\delta(x^{(t+1)} - x^{(t)} - v)$. However, in practice, the exact vehicle velocity is rarely known. We assume that the vehicle has a "speed uniformly" (i.e. distance traveled per sample interval) distributed in $[0, v_{max}]$, and the vehicle heading is uniform in $[0, 2\pi)$. Therefore, $p(x^{(t+1)}|x^{(t)})$ is a disk centered at $x^{(t)}$ with radius v_{max}. Under this model, the predicted belief $p(x^{(t+1)}|\overline{z^{(t)}})$ (the integral in (3)) is obtained by convolving the old belief $p(x^{(t)}|\overline{z^{(t)}})$ with the uniform circular disk kernel. The convolution reflects the dilated uncertainty about target location due to motion.

Once the updated belief $p(x^{(t+1)}|\overline{z^{(t+1)}})$ is computed, the current leader hands it off to one of its neighboring sensors and goes back to sleep. An information-driven sensor selection criterion, described in Sec. 4, is used to decide which neighboring sensor to hand the belief to based on the expected contribution from that sensor. The most "informative" sensor is selected and becomes the leader for the the next time step $t+1$. The IDSQ tracking algorithm is summarized in Table 1.

Note that minimal assumptions are made in this formulation. We do not require any knowledge about the road configuration, and do not make the exclusive assumption that the vehicle travels only on a road. In particular, the vehicle dynamics model is rather crude. The vehicle can accelerate, decelerate, turn, or stop. We assume that the vehicle velocity sequence sampled at the tracking interval is statistically independent. Only v_{max} has to be known or assumed. These minimal

$x^{(t)}$: target position at time t;	$z^{(t)}$: sensor measurement at time t;
v_{max}: upper bound on target speed;	$\mathcal{N}$: neighbor list.
step 0. Sleep until receive handoff package $(t, p(x^{(t)}\|\overline{z^{(t)}}))$.	
step 1. Diffuse belief using vehicle dynamics: $p(x^{(t+1)}\|\overline{z^{(t)}}) = \int p(x^{(t+1)}\|x^{(t)}) \cdot p(x^{(t)}\|\overline{z^{(t)}})\ dx^{(t)}$.	
step 2. Do sensing: compute $z^{(t+1)}$; compute $p(z^{(t+1)}\|x^{(t+1)})$.	
step 3. Compute $p(x^{(t+1)}\|\overline{z^{(t+1)}}) \propto p(z^{(t+1)}\|x^{(t+1)})p(x^{(t+1)}\|\overline{z^{(t)}})$.	
step 4. For sensor $k \in \mathcal{N}$, compute information utility $I_k = I(x^{(t+2)}; z^{(t+2)}\|\overline{z^{(t+1)}})$ select $k_{next} = \arg\max I_k$.	
step 5. Handoff $(t+1, p(x^{(t+1)}\|\overline{z^{(t+1)}}))$ to k_{next}. Go back to step 0.	

Table 1: Algorithm for IDSQ tracker at each node.

assumptions allow the algorithm to achieve simplicity, flexibility and wide applicability. On the other hand, more accurate prior knowledge can be incorporated to further improve the performance. For example, adding road constraints could further improve the tracking accuracy and decrease the computational load. We are currently exploring more realistic vehicle dynamics models, which take into account higher-order dynamics and other complex characteristics of vehicle trajectories. However, for the purposes of this paper, we have opted for the lower computational complexity of this simple model.

This algorithm is fully distributed. There is no single central node in the network. Note that the sensor nodes do not have a global knowledge about the network such as its topology. Each node is only aware of its immediate neighbors and their specifications. The communication is exclusively neighbor-to-neighbor. Only local computation is involved in computing the measurement $z^{(t)}$ and updating the belief state. As discussed in Sec. 1, such fully decentralized characteristics are often desirable to ensure reliability, survivability, and scalability of the sensor network.

2.2 Comparisons to centralized Bayesian estimation

It is interesting to see how this distributed sensor network compares to a fully centralized one. Consider a centralized sensor network consisting of K sensors. At any time instant t, each sensor k ($k = 1, 2, \cdots, K$) informs the central processing unit about its measurement $z_k^{(t)}$. The central processing unit updates the belief state using the same sequential Bayesian filtering technique as in

	centralized	distributed
Computation	$O(K \cdot \|belief\|)$, if (4) is true $O(K^2 \cdot \|belief\|)$, else	$O(\|\mathcal{N}\| \cdot \|belief\|)$
Bits to be communicated	$O(K \cdot \|belief\|)$	$O(\|belief\|)$
Wireless comm. power	$O(\|belief\| \cdot \sum_k \|\|\zeta_k - \zeta_{center}\|\|^\alpha)$	$O(\|belief\| \cdot \|\|\zeta_{next\ leader} - \zeta_{leader}\|\|^\alpha)$

Table 2: Single-step cost for centralized and distributed Bayesian tracking. In the second row, $\mathcal{N}$ is the leader's neighbor list. The last row is the power needed to communicate reliably through radio. We assume a particular model where the communication power is adjustable, and is proportional to the communication distance raised to the power of RF attenuation exponent α ($\alpha \geq 2$). RF overhead consumption and power consumption of sensing and is neglected.

(3), with the difference that instead of using the single sensor measurement $z^{(t)}$ (which is a scalar), it uses the measurement vector $\underline{z}^{(t)} = \{z_1^{(t)}, z_2^{(t)}, \cdots, z_K^{(t)}\}$. If the sensors measurements are mutually independent when conditioned on the target locations, then

$$p(\underline{z}^{(t)}|x^{(t)}) = \prod_{k=1,\cdots,K} p(z_k^{(t)}|x^{(t)}). \tag{4}$$

Compared to the centralized tracking algorithm, which utilizes all K measurements at every time step, our distributed algorithm incorporates only one out of $|\mathcal{N}|$ measurements each step, where $|\mathcal{N}|$ is the size of the leader node's local neighborhood, and in general, $|\mathcal{N}| \ll K$. Hence the distributed algorithm suffers from decreased tracking accuracy but scales much better in computation and communication as the network grows. Table 2 summarizes the cost for each step of tracking in the centralized and distributed schemes. Unlike the centralized algorithm whose complexities go up linearly or superlinearly with K, the distributed algorithm has complexities independent of K.

2.3 Nonparametric belief representation

As we shall see in Sec 3, the observational model is nonlinear; the likelihood $p(z^{(t)}|x^{(t)})$ is non-Gaussian, as is the posterior belief $p(x^{(t)}|\overline{z^{(t)}})$. In view of these characteristics, we use a nonparametric representation for probability distributions (see Fig. 1). The distributions are represented as discrete grids in the two-dimensional plane. The grey level depicts the probability distribution function (pdf) evaluated at the grid location. The lighter the grid square, the higher the pdf value.

The nonparametric representation of likelihood and posterior belief admits an efficient computation. The MMSE estimate (2) is simply the average of the grid locations in the belief cloud weighted by the belief value. The predicted belief $p(x^{(t+1)}|\overline{z}^{(t)})$ (the integral in (3)) is a weighted sum of the vehicle dynamics pdf conditioned on each grid point in the original belief cloud.

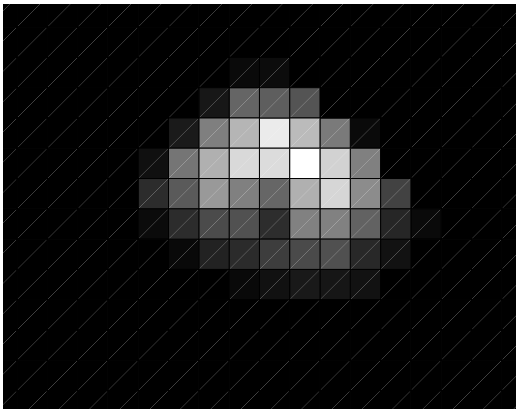


Figure 1: Nonparametric representation of belief state

The resolution of the grid representation is constrained by the computational capacity, storage space, and communication bandwidth of the leader node. For our choices of sensors, as will be detailed in the next section, the likelihood functions are relatively smooth. This smoothness allows low resolution representation without much loss in performance. Furthermore, in our experiment, we store only the grid points which have likelihood value above a fixed threshold. The grid points below this likelihood are neglected.

3 Sensor Models

We use two types of sensors for tracking, acoustic amplitude sensors and direction-of-arrival (DOA) sensors. The acoustic amplitude sensors calculate sound amplitude measured at each microphone, and estimate the distance of the vehicle based on the physics of sound attenuation. The DOA sensors are small microphone arrays. Using beamforming techniques, they determine the direction where the sound comes from, i.e., the bearing of the vehicle.

The nonparametric Bayesian approach we are using poses few restrictions on sensor type, and allows the network to easily fuse data from multiple sensor types. Relatively low cost sensors such as microphones are attractive because of affordability as well as simplicity in computation, compared to imagers. However, there are no barriers to adding other sensor types, including imaging, motion, or magnetic sensors.

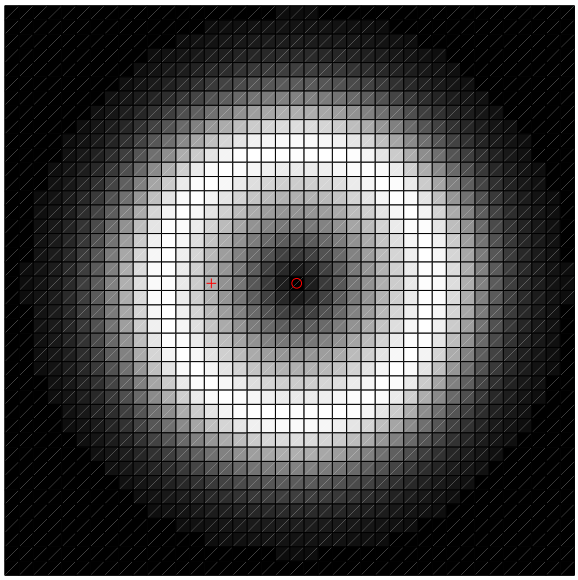


Figure 2: Likelihood function $p(z|x)$ for acoustic amplitude sensors. The circle is the sensor location ζ, and the cross is the true target location.

3.1 Acoustic amplitude sensors

Assuming that the sound source is a point source, and sound propagation is lossless and isotropic, a root-mean-squared (rms) amplitude measurement z is related to the sound source position x as

$$z = \frac{a}{||x - \zeta||} + w, \tag{5}$$

where a is the rms amplitude of the sound source, ζ is the location of the sensor, and w is rms measurement noise [6]. For simplicity, we model w as Gaussian with zero mean and variance σ^2. The sound source amplitude a is also modeled as a random quantity. Assuming a is uniformly distributed in the interval $[a_{lo}, a_{hi}]$, the likelihood has a closed-form expression:

$$\begin{aligned} p(z|x) &= \int_{a_{lo}}^{a_{hi}} p(z|x,a)p(a)da \\ &= \frac{1}{\Delta_a}\int_{a_{lo}}^{a_{hi}} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\frac{a}{r})^2}{2\sigma^2}} da \\ &= \frac{r}{\Delta_a}\left[\Phi\left(\frac{a_{hi} - rz}{r\sigma}\right) - \Phi\left(\frac{a_{lo} - rz}{r\sigma}\right)\right], \end{aligned} \tag{6}$$

where $\Delta_a = a_{hi} - a_{lo}$, $r = ||x - \zeta||$ is the distance between the sound source and the sensor, and $\Phi(\cdot)$ is the standard error function. The details of the derivation is referred to [3].

Fig. 2 shows an example of the likelihood function $p(z|x)$, a crater-shaped function centered at the sensor location. The thickness of the crater (outer radius minus inner radius) is determined by

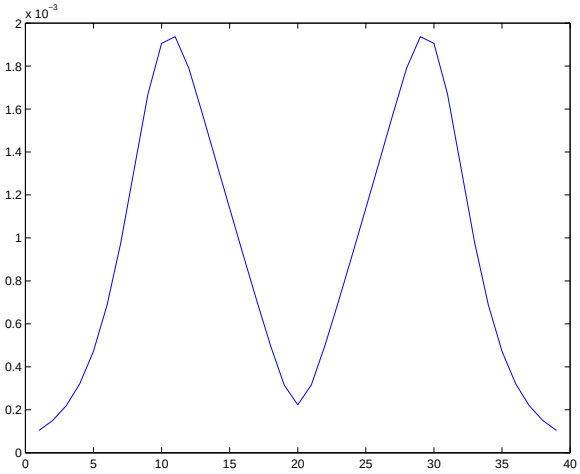


Figure 3: The cross-section of the likelihood (plotted in Fig. 2) along the horizontal line past the sensor location.

a_{lo}, a_{hi}, and σ^2. Fixing the first two, the thickness increases with σ^2, as the target location is more uncertain. Fixing σ^2, the thickness increases as a_{lo} decreases or as a_{hi} increases. In cartesian space, it is clear that this likelihood is non-gaussian, and difficult to approximate as the sum of gaussians. The cross-section of the likelihood function along the radial direction is plotted in Fig. 3, and it is quite smooth and amenable to approximation by sampling.

The uniform, stationary, assumption of source amplitude is computationally lightweight. To accommodate quiet vehicles and vehicles in idle state, a_{lo} is set to zero in our experiments, and a_{hi} is set via calibration. In practice, the uniform assumption is simplistic, and as part of our on-going work, we are developing new models of source amplitude to better model vehicle engine sound characteristics, exploiting the correlation of sound energy over time.

3.2 DOA Sensors

Amplitude sensing provides a range estimate. This estimate is often not very compact and is limited in accuracy due to the crude uniform sound source amplitude model. These limitations makes the addition of a target bearing estimator very attractive.

For estimating the bearing of the sound source, we use the maximum likelihood (ML) DOA estimation algorithm proposed by Chen *et al.* [7]. Here we only outline the formulation of the estimation problem; interested readers may refer to their paper for more details.

Assume that we have a microphone array composed of M identical omni-directional microphones, and the sound source is sufficiently far away from the microphone array so that the wave

received at the array is a planar wave. In this case, the data collected at the m-th microphone at time n is

$$g_m(n) = s_0(n - t_m) + w_m(n), \tag{7}$$

where s_0 is the source signal, w_m is the noise (assumed white Gaussian), and t_m is the time delay, which is a function of the direction of arrival θ. Now consider the equivalent problem in the Fourier frequency domain (by DFT of length L, $L \gg M$). We have

$$\mathbf{G}(l) = \mathbf{D}(l)S_0(l) + \mathbf{W}(l), \tag{8}$$

for $l = 0, 1, \cdots, L-1$, where

- $\mathbf{G}(l) = [G_1(l), G_2(l), \cdots, G_M(l)]^T$ is the frequency component of the received signal at frequency l.
- $S_0(l)$ is the signal component.
- $\mathbf{W}(l) = [W_1(l), W_2(l), \cdots, W_M(l)]^T$ is the noise component.
- The steering matrix takes the form $\mathbf{D}(l) = [D_1(l), D_2(l), \cdots, D_M(l)]^T$, and $D_m(l) = e^{-j2\pi l t_m / L}$.

The ML estimator seeks an estimate

$$(\hat{\theta}, \hat{S}_0) = \arg\min_{\theta, S_0} \sum_{l=0}^{L-1} ||\mathbf{G}(l) - \mathbf{D}(l)S_0(l)||^2. \tag{9}$$

Given the arriving angle θ, the signal spectrum estimate is

$$\hat{S}_0(l) = \mathbf{D}^\dagger(l)\mathbf{G}(l), \tag{10}$$

where $\mathbf{D}^\dagger(l)$ is the pseudo-inverse of the steering matrix $\mathbf{D}(l)$. Plugging-in (10), (9) boils down to a one-dimensional optimization over $\theta \in [0, 2\pi]$, which can be solved using simple searching techniques. This DOA algorithm works well for wideband acoustic signals, and does not require the microphone array to be linear or uniformly spaced.

In our experiments, we use a microphone array with four microphones, as shown in Fig. 4. The centroid's location is defined as $\zeta_0 = (0, 0)$; the arriving angle θ is defined as the angle with respect to the vertical grid axis. This convention is used throughout the paper.

Due to the presence of noise, the DOA estimate of angles is often imperfect. The measurement z (i.e., $\hat{\theta}$) is close to the true underline angle θ with some perturbation. To characterize the likelihood

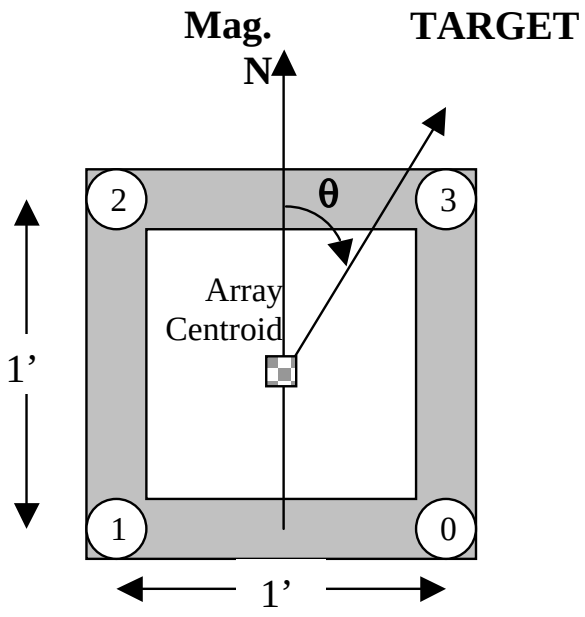


Figure 4: DOA sensor arrangement and angle convention

function $p(z|x)$, we tested the DOA algorithm using recorded vehicle sounds from an AAV, running the DOA algorithm on an actual sensor node (see Sec. 5.2 for information on the node and vehicle). The test took place under reasonable noise conditions including air handling units at a nearby building and some street and aircraft traffic. We performed tests at $r = 50$, 150, 200, and 500 feet, and $\theta = 0, \frac{\pi}{8}, \frac{\pi}{4}$ and $\frac{3\pi}{8}$.

Since the microphone array is symmetric in four quadrants, we only have to examine the first quadrant. At each combination, we ran the DOA algorithm 100 times and computed the histogram of the DOA estimates $\{z_{exp1}, z_{exp2}, \cdots, z_{exp100}\}$. Fig. 5 shows the histograms for $r = 50$ feet. The histograms suggest that the distribution of z is unimodal and approximately centered at θ. Hence for the angle measurement z, a Gaussian model with zero mean is appropriate. The likelihood takes the form

$$p(z|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-\theta)^2}{2\sigma^2}}, \tag{11}$$

where θ is calculated from the geometry of the sound source position x and the sensor position ζ. This model ignores the periodicity of angles around 2π, and is accurate when the variance σ^2 is small.

In our experiments, we have observed that the DOA estimates are reliable in some middle distance range and is less reliable when the sound source is too near or too far away from the

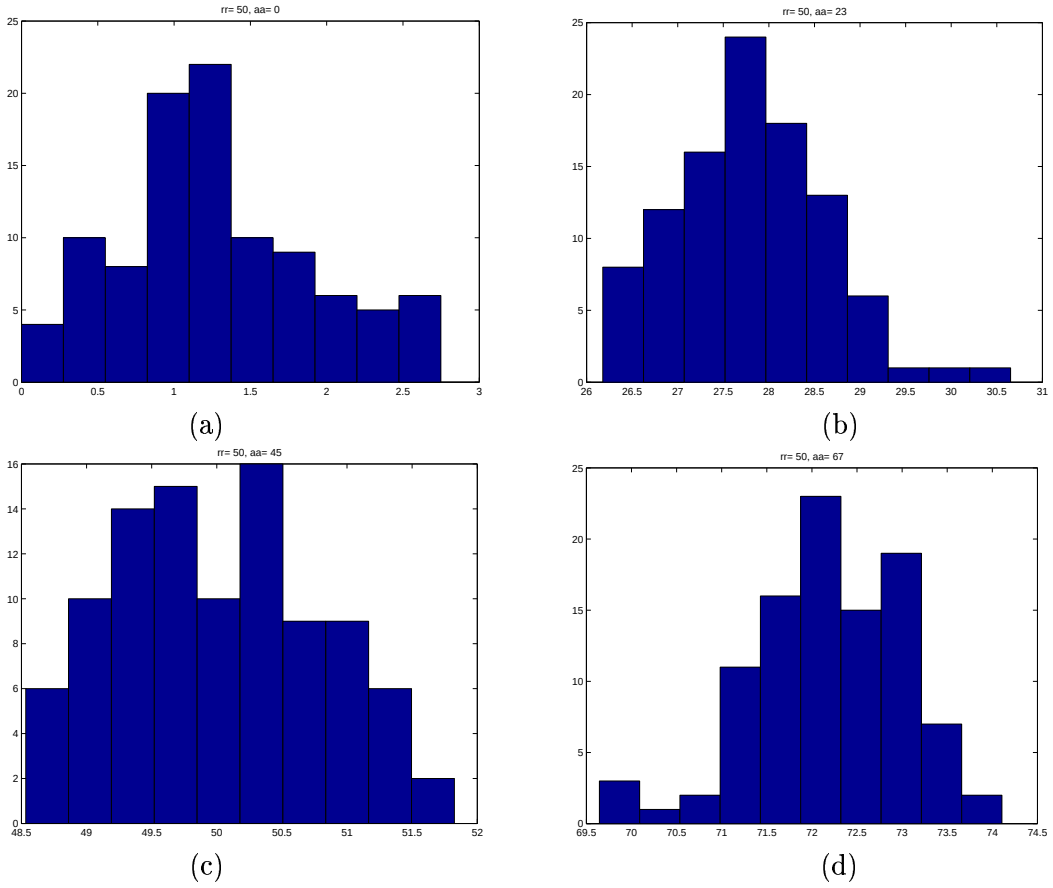


Figure 5: Histograms of the DOA estimate z at $r = 50$ feet and $\theta = 0, \frac{\pi}{8}, \frac{\pi}{4},$ and $\frac{3\pi}{8}$ in (a)-(d) respectively.

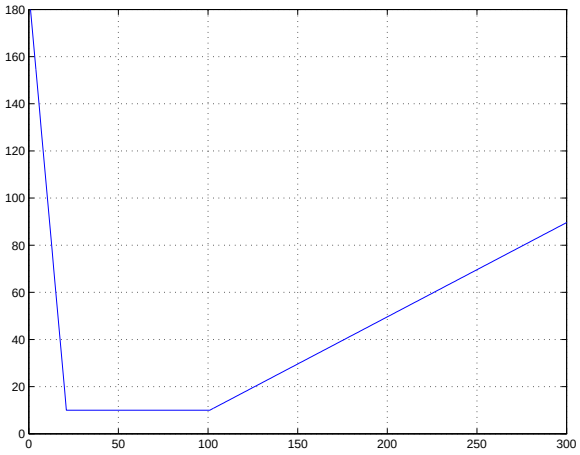


Figure 6: Standard deviation σ in bearing estimation vs. range. Here $r_{near} = 20$ meters, $r_{far} = 100$ meters, and $\sigma = 10^o$ in $[r_{near}, r_{far}]$.

microphone array. This is to be expected. In the near field, the planar wave assumption is violated. In the far field, the SNR is low, and the the DOA algorithm may be strongly influenced by noise and fail to obtain the correct angle. To account for these factors, we developed a simplified likelihood model which is qualitatively reasonable and empirically behaves well. The model varies the standard deviation σ according to distance, as illustrated in Fig. 6. We first specify the range $[r_{near}, r_{far}]$ in which the DOA algorithm performs reliably. In this range, the DOA estimate has a fixed standard deviation. For the nearfield range $[0, r_{near})$, as distance decreases, the standard deviation increases linearly to account for the increasing uncertainty of DOA readings. Likewise, in the far field range $r > r_{far}$, the standard deviation increases with r.

Under this model, the likelihood function $p(z|x)$ is plotted in Fig. 7. It has a cone shape in the working range and fans out in the near and far range. If the target is located in either end of the range, the DOA estimates are unreliable, thus the angle measurement does not provide much evidence about the target's location. Note that the likelihood in the two-dimensional Cartesian plane is not compact. The sequential Bayesian filtering approach (as described in Sec. 2.1) has the flexibility to accommodate such non-compactness, while a standard Kalman filtering approach may have difficulty here.

4 Information-Driven Sensor Selection

Sensor selection is essential for the correct operation of the IDSQ tracking algorithm. The selection criterion is based on information content, to maximize the *predicted* information that a sensor's

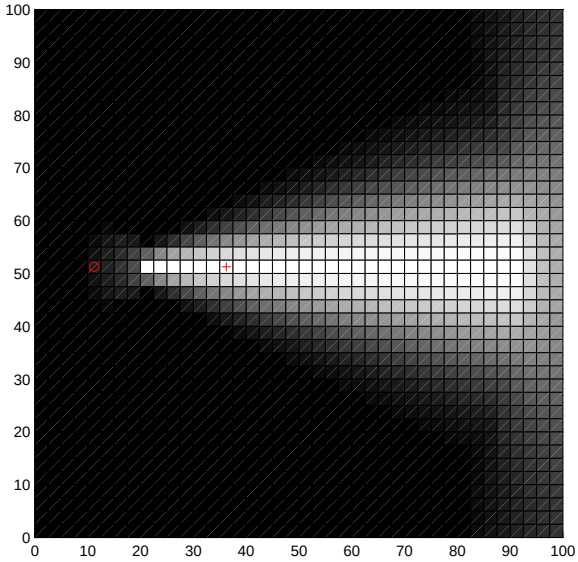


Figure 7: Likelihood function $p(z|x)$. The circle is the sensor location, and the cross is the target location. The standard deviation of the Gaussian distribution is as in Fig. 6.

measurement will bring. This selection is performed based on currently available information alone: the current leader's belief state and its knowledge about the neighboring sensor locations and their sensing capabilities. No querying of neighboring sensors is necessary.

To measure the information content, we consider mutual information, a measure commonly used for characterizing the performance of data compression, classification, and estimation algorithms and with a root in information theory.

4.1 Mutual information

Let $U \in \mathcal{U}$ and $V \in \mathcal{V}$ be two random variables (or vectors) having a joint pdf $p(u,v)$.[1] The mutual information between U and V is defined as

$$\begin{aligned} I(U;V) &\triangleq E_{p(u,v)}\left[\log \frac{p(u,v)}{p(u)p(v)}\right] && (12)\\ &= \int_{\mathcal{U}}\int_{\mathcal{V}} p(u,v)\log\frac{p(u,v)}{p(u)p(v)}\,dudv && (13)\\ &= D(p(u,v)||p(u)p(v)), && (14)\end{aligned}$$

where $D(\cdot||\cdot)$ is the relative entropy between two distributions, also known as the Kullback-Leibler divergence [8]. Similarly the mutual information conditioned on a random variable $W = w$ is

[1] In this section, we use the standard notation, with upper case symbols denoting random variables and lower case symbols denoting a particular realization.

defined as $I(U;V|W=w) \stackrel{\triangle}{=} E_{p(u,v|w)}\left[\log \frac{p(u,v|w)}{p(u|w)p(v|w)}\right]$. We use logarithm of base 2, hence $I(U;V)$ is measured in bits. The mutual information is symmetric in U and V, nonnegative, and is equal to zero if and only if U and V are independent.

The mutual information $I(U;V)$ indicates how much information V conveys about U. From a data compression perspective, it measures the savings in bits of encoding U if V is already known. In classification and estimation problems, mutual information can be used to establish performance bounds. The higher $I(U;V)$ is, the easier it is to estimate (or classify) U given V, or vice-versa [9, 10].

4.2 Sensor selection criterion

In our target tracking problem, we formulate the sensor selection criterion as follows. The leader, with a belief state $p(x^{(t)}|\overline{z^{(t)}})$, must decide which sensor in its neighborhood to hand the belief to. The IDSQ suggests selecting the sensor

$$k_{IDSQ} = \arg\max_{k\in\mathcal{N}} I(X^{(t+1)}; Z_k^{(t+1)}|\overline{Z^{(t)}} = \overline{z^{(t)}}), \tag{15}$$

where $\mathcal{N}$ is the collection of sensors which the current leader can talk to, namely the leader's neighborhood. Essentially, this criterion seeks the sensor whose measurement $z_k^{(t+1)}$, combined with the current measurement history $\overline{z^{(t)}}$, would provide the greatest amount of information about the target location $x^{(t+1)}$. Intuitively, k_{IDSQ} is the most "informative" sensor among the neighborhood $\mathcal{N}$.

From the definition of mutual information (12), the information content of sensor k is

$$I(X^{(t+1)}; Z_k^{(t+1)}|\overline{Z^{(t)}} = \overline{z^{(t)}}) = E_{p(x^{(t+1)}, z_k^{(t+1)}|\overline{z^{(t)}})}\left[\log \frac{p(x^{(t+1)}, z_k^{(t+1)}|\overline{z^{(t)}})}{p(x^{(t+1)}|\overline{z^{(t)}})p(z_k^{(t+1)}|\overline{z^{(t)}})}\right]. \tag{16}$$

The computation of mutual information is illustrated in Table 3.

We can also take a different view of mutual information, interpreting it as a measure of the difference between two densities. From (16), one can easily show that

$$\begin{aligned} I(X^{(t+1)}; Z_k^{(t+1)}|\overline{Z^{(t)}}) &= E_{p(z_k^{(t+1)}|\overline{z^{(t)}})}\left[E_{p(x^{(t+1)}|\overline{z_k^{(t+1)}})} \log \frac{p(x^{(t+1)}|\overline{z_k^{(t+1)}})}{p(x^{(t)}|\overline{z^{(t)}})}\right] & (17)\\ &= E_{p(z_k^{(t+1)}|\overline{z^{(t)}})} D\left(p(x^{(t+1)}|\overline{z_k^{(t+1)}})||p(x^{(t+1)}|\overline{z^{(t)}})\right). & (18) \end{aligned}$$

Initialization: $p(x^{(t)}|\overline{z^{(t)}})$ is known.

step 1. compute $p(x^{(t+1)}|\overline{z^{(t)}})$ by diffusing $p(x^{(t)}|\overline{z^{(t)}})$ with vehicle dynamics (see Sec. 2.1).

step 2. set $\mathcal{X}$ as the nontrivial grids in $p(x^{(t+1)}|\overline{z^{(t)}})$;
set $\mathcal{Z} \subset \mathbb{R}$ as the grid points for $z_k^{(t+1)}$.

step 3. for $z_k^{(t+1)} \in \mathcal{Z}$ and $x^{(t+1)} \in \mathcal{X}$,
evaluate $p(x^{(t+1)}, z_k^{(t+1)}|\overline{z^{(t)}}) = p(z_k^{(t+1)}|x^{(t+1)}) \cdot p(x^{(t+1)}|\overline{z^{(t)}})$.

step 4. for $z_k^{(t+1)} \in \mathcal{Z}$, compute $p(z_k^{(t+1)}|\overline{z^{(t)}}) = \sum_{\mathcal{X}} p(x^{(t+1)}, z_k^{(t+1)}|\overline{z^{(t)}})$.

step 5. for $x^{(t+1)} \in \mathcal{X}$ and $z_k^{(t+1)} \in \mathcal{Z}$,

$$D_{xz} = \log\left[\frac{p(x^{(t+1)}, z_k^{(t+1)}|\overline{z^{(t)}})}{p(x^{(t+1)}|\overline{z^{(t)}})p(z_k^{(t+1)}|\overline{z^{(t)}})}\right]$$

$$I_k = \sum_{\mathcal{X},\mathcal{Z}} D_{xz} \cdot p(x^{(t+1)}, z_k^{(t+1)}|\overline{z^{(t)}})$$

Return I_k.

Table 3: Computation of mutual information $I(X^{(t+1)}; Z_k^{(t+1)}|\overline{Z^{(t)}})$.

The Kullback-Leibler divergence term measures how different the updated belief after incorporating the new measurement $z_k^{(t+1)}$ would be from the current belief. Therefore, the IDSQ criterion favors the sensor which would on average give the greatest change to the current belief.

4.3 Reduction in dimensionality

Using the discrete representation of the belief, the complexity of computing mutual information grows exponentially in the dimension of the joint pdf. The random variable $X^{(t+1)}$ is a two-dimensional vector for the target tracking problem over a two-dimensional plane. Thus, we need to compute mutual information from the three-dimensional joint density $p(x^{(t+1)}, z_k^{(t+1)}|\overline{z^{(t)}})$. This may be computationally intensive given the limited ability of the sensor nodes.

For acoustic amplitude sensors, we note that observation model (5) indicates that at any given time instant, the observation Z_k is related to the position X only through $R_k = ||X - \zeta_k||$, the distance from the target to the sensor positioned at ζ_k. Equivalently we have $p(z_k|r_k, x) = p(z_k|r_k)$. In this case, R_k is known as the sufficient statistics of X. From the definition of mutual information,

it is easy to show

$$I(X; Z_k) = I(R_k; Z_k). \tag{19}$$

This implies that instead of computing the mutual information from the three-dimensional density $p(x, z_k)$, one can compute it from the two-dimensional density $p(r_k, z_k)$.

Likewise, for DOA sensors, from the observational model (11) we see that Z_k is related to X only through the angle $\theta(X, \zeta_k)$. Hence

$$I(X; Z_k) = I(\theta, Z_k).$$

Again, the computation of mutual information can be reduced to a two-dimensional computation.

5 Experimental Results

To validate and characterize the performance of the tracking algorithm, we carried out both simulations as well as experiments using real data collected from the field.

5.1 Simulation

In the simulation, the vehicle produces stationary sound with a constant rms amplitude $A = 40$ and is traveling at a constant speed $v = 7$ meters/second along a straight line (south to north) in the middle of a field. The field is 150×250 meter^2 and covered by K randomly placed sensors ($K = 24, 28, \cdots, 60$). The sensor positions are simulated as follows:

- first place the sensors in a uniform rectangular grid with $K/4$ rows and four columns, evenly covering the field;
- then add Gaussian noise with distribution $N(0, 25)$ to the horizontal and vertical coordinates.

A realization of this sensor position simulation is pictured in Fig. 8. Acoustic sensor measurements are simulated as $\frac{A}{||x-\zeta_k||} + N(0, 0.05^2)$. DOA sensor measurements are Gaussian random variables centered at the line through the sensor and the target with $\sigma = 3^o$.

Without precise knowledge about the target vehicle, the tracking algorithm allows maximum speed of 15 meters/second. The acoustic amplitude sensors assume $a_{lo} = 0$, $a_{hi} = 80$, and $\sigma_k =$

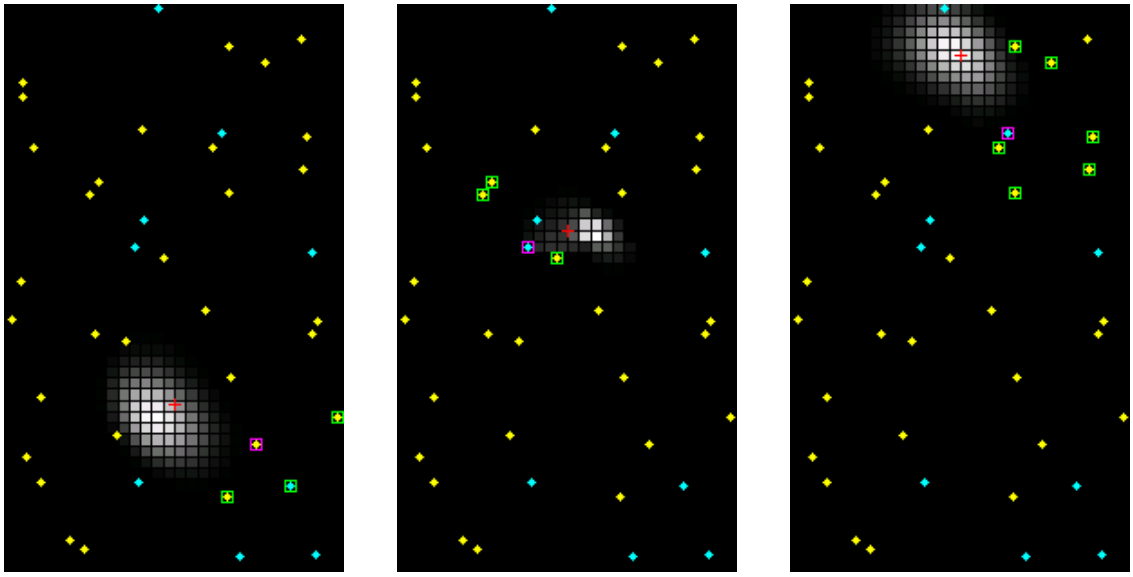
0.1 (twice the actual noise contamination to accommodate outliers). The DOA sensors assumes $r_{near} = 20$ meters and $r_{far} = 100$ meters. The standard deviation of the DOA estimates is as plotted in Fig. 6. The tracker updates the belief every 0.5 second.

The tracking algorithm begins with an initial belief which is uniform over the entire field. The acoustic amplitude sensor with the highest amplitude reading at time $t = 0$ is initialized as the leader. The connectivity between sensors are determined as follows: each sensor can talk to sensors within a 40-meter range; or if there are less than two sensors in range, it can talk to the two nearest sensors. The leader selects the next leader from among these sensors based on their information content according to the IDSQ rule.

To further enforce sensor diversity, we implement a "triplet" rule: the previous leader is not included in the neighborhood of the current leader. That is, if A hands off to B in the previous step, B is prohibited from selecting A as the next leader. This rule prevents the leadership from constantly oscillating between two sensors. This is important in assuring that biases due to modeling error of one particular sensor will not be overweighted in the overall result. Ideally, one would like to task the sensors more or less evenly to exploit the spatial and sensing modality diversity of the sensors, expecting that the modeling error across different sensors will probably balance each other.

Fig. 8 shows three snapshots of a simulation run with 40 sensors, 30% of which are DOA. The belief is shown using the grid-based nonparametric representation described in Sec. 2.3. The grid size is 5 meters in each direction (which is approximately the size of a tank). The belief follows the target fairly closely. In general, the belief cloud is more compact in sensor-dense regions (e.g., in Fig. 8b) than in sensor-sparse regions (e.g., in Fig. 8a and c) . This is as expected, since the SNR is higher in sensor-dense regions, and the leader has more neighbors to choose from. The collaboration between sensors are more effective.

Table 4 summarizes the statistics from simulation results for different values of K. For each K value, twenty runs are simulated. We use x to denote the target's true location, $\hat{x}$ to denote the the blocks in the belief state, and $\hat{x}_{MMSE}$ to denote the MMSE estimate of target location. From (2), we know that $\hat{x}_{MMSE}$ is simply the centroid of the belief state, computed as the weighted average of location among all the blocks in the belief, weighted by the posterior $p(x^{(t)}|\overline{z^{(t)}})$. To analyze tracking performance, we use the distance $||\hat{x}_{MMSE} - x||$ to measure how far away the MMSE estimate is to the true target position, and use the variance $||\hat{x} - \hat{x}_{MMSE}||^2$ to measure the spread (thus the uncertainty) of the belief cloud. Other quantities of interest include the belief



(a) (b) (c)

Figure 8: Snapshots of a simulation run with 40 sensors, 30% DOA. The target is marked with a red "+". The yellow diamonds are acoustic amplitude sensors; the cyan diamonds are DOA sensors. The active leader is the sensor circled with a magenta square. Its neighbors (after applying the triplet rule) are circled with green squares.

K	avg. $\|\|\hat{x}_{MMSE} - x\|\|$	avg. $\|\|\hat{x} - \hat{x}_{MMSE}\|\|^2$	avg. belief size	avg. neigh. size
24	24.13	457.81	188.6	2.0
28	23.25	452.97	182.0	2.2
32	19.29	400.68	155.2	2.4
36	16.68	340.41	141.7	2.9
40	16.00	361.88	145.0	3.0
44	14.62	352.32	129.5	3.5
48	14.29	322.09	134.6	4.0
52	13.82	334.57	123.3	4.6
56	12.85	332.52	120.5	5.1
60	12.81	304.56	116.4	5.5
64	12.42	284.99	109.1	6.0

Table 4: Tracking performance averaged over all tracking steps and twenty runs.

cloud size (directly related to communication throughput) and the neighborhood size. From Table 4 we can see that the tracking performance (the mean error, variance, and belief size) improves as more sensors are deployed. Fig. 9 plots the average error $||\hat{x}_{MMSE} - x||$. Outliers (the tracker losing the target) occur occasionally, especially for small K. The improvement in average tracking performance (the blue curve) with increasing K is quite prominent.

We have also experimented with varying percentage of DOA sensors and summarized the result in Table 5. The improvement from 0 to 10% DOA is significant, thus it is desirable to use a few DOA sensors in the sensor network, though they are computationally more expensive than acoustic amplitude sensors. The all-DOA network gives better results than the all-amplitude sensor network. This may be due to the fact that the acoustic amplitude sensors uses the very crude uniform distribution for modeling sound source amplitude.

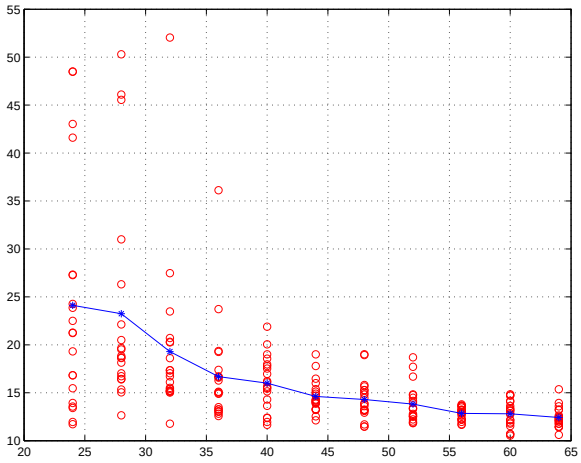


Figure 9: Average error vs. the number of sensors in the field. The points marked with a circle are the error average over the tracking steps. The points marked with "$*$" and linked using a blue line is the average over 20 runs.

Percentage of DOA	avg $\|\|\hat{x}_{MMSE} - x\|\|$	avg $\|\|\hat{x}_{MMSE} - \hat{x}\|\|^2$	avg belief size
0	18.48	510.96	208.9
10%	16.84	416.17	168.2
20%	16.45	379.03	154.0
30%	15.48	345.68	138.3
40%	14.01	331.55	122.2
50%	14.05	373.88	122.6
60%	12.87	375.45	112.2
70%	11.89	327.94	104.4
80%	11.67	349.60	102.1
90%	11.31	366.50	99.7
100%	10.75	351.02	95.4

Table 5: Tracking performance vs. percentage of DOA sensors.

Figure 10: AAVP7A1 tracked, armored assault vehicle.

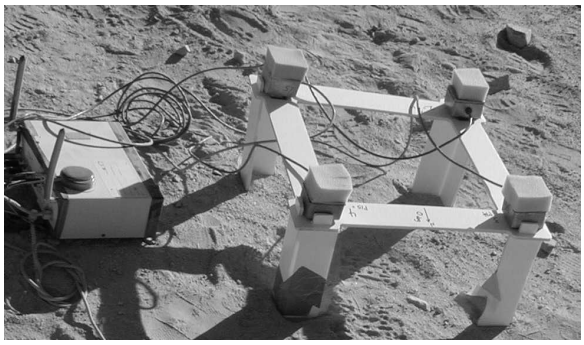


Figure 11: WINS 2.0 node with DOA-sensing microphone array.

5.2 Experiments on field data

5.2.1 Experimental setup

Data for our experiments was collected during a field experiment at the Marine Corps Air-Ground Combat Center (MCAGCC) in Twenty-nine Palms, California. The test vehicle was an AAVP7A1 tracked, armored assault amphibious vehicle. This vehicle, shown in Fig. 10, is a 4.1 meter-long, tracked, diesel-powered vehicle, weighing 21 tons (unloaded) and capable of speeds up to 72 kilometers/hour.

The sensor network consisted of a total of 70 WINS 2.0 nodes from Sensoria Corp. A picture of a node is shown in Fig. 11 along with a DOA array. The node specifications are provided in Table 6. Among the 70 nodes, 20-25 nodes were available for our use. They are randomly positioned at the intersection of two dirt roads. The sensor layout and the roads are shown in Fig. 12. Vehicle maneuvers were confined to the roads shown, although by choice, the algorithms of this paper do not assume this knowledge.

Manufacturer:	Sensoria Corp.
Processor:	Hitachi SH4 7751 Performance: 300 MIPS core, 1.1 GFLOPS floating point CPU Power Consumption: 400 mW
Memory:	16 MB RAM, 16 MB Flash
Operating System:	Redhat Linux 7
Data Acquisition:	4 channels, 16 bits @ 20 kHz (this experiment sampled at 5 kHz throughout)
Communications:	2 RF Modems @ 2.4 GHz Power: 10 or 100 mW Range: 25-100 m

Table 6: Specification of Sensoria nodes.

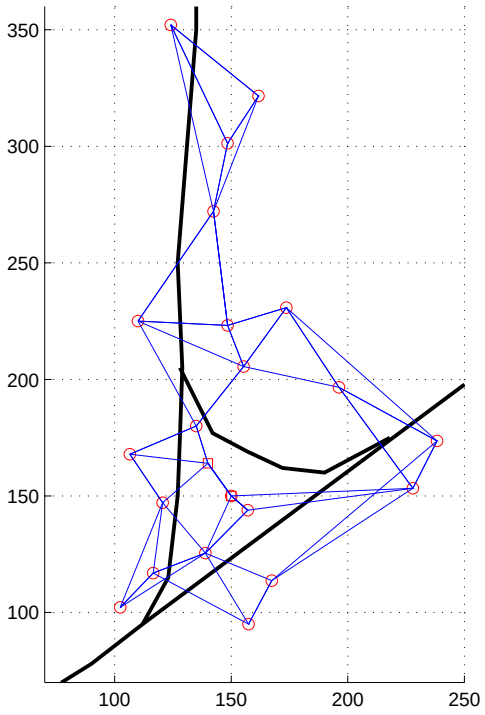


Figure 12: Road intersection with node layout at MCAGCC. The thick lines are the roads. The amplitude sensors are marked with a small circle, and the two DOA sensors (at (150, 150) and (140, 164)) are marked with small squares. The thin lines indicate the connectivity between sensors.

DOA sensing arrays were identical square arrays with 1' on the side configured as shown in Fig. 11 and on legs 8" off the ground. The zero degree bearing is aligned to magnetic north as shown in Fig. 4. Microphones were electret, flat to frequencies above 16 kHz, omnidirectional to less than 2.5 dB, and were field calibrated to $\pm$ 2.5 dB.

5.2.2 Data acquisition and processing

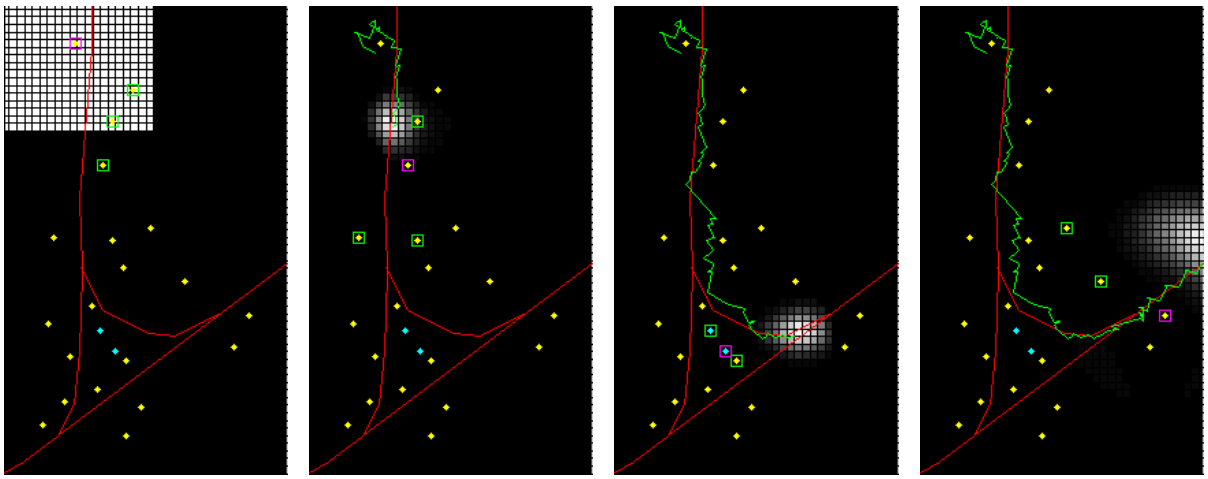
Filtered acoustic amplitude and estimated DOA were computed and stored on all nodes at time intervals of 0.5 seconds using the node signal processing routines described above. Additional data was collected for two DOA nodes using a portable data acquisition PC and sampled at an identical rate and resolution to the node's data acquisition. In the experiments shown here, only these two "virtual" nodes are used for DOA measurements.

The IDSQ tracker, implemented in Matlab, then postprocessed this data to simulate the node-to-node handoff and data source selection. Due to node dropout, the original network topology, as layed out in the field, has to be slightly augmented to prevent network segmentation; the resulting topology is shown in Fig. 12.

5.2.3 Tracking results

Fig. 13 shows a few snapshots of an AAV run on the north-east road. The vehicle is traveling at roughly a constant speed of 15 miles/hour. The tracker is designed for $v_{max} = 40$ miles/hour and updates its belief every 0.5 second. For acoustic amplitude sensors, the parameters are set as $a_{lo} = 0$, $a_{hi} = 80$, and $\sigma_k = 0.05$ based on calibration data. For DOA sensors, the variance is set as in Fig. 6. At time $t = 0$, the acoustic amplitude sensor with the highest reading is initialized as the leader. The initial belief is a big uniform square centered at the leader with 50 meters (10 grids) to each side, as plotted in Fig. 13a. The first few tracking steps can be considered as a "discovery" phase, where the tracker begins with very little knowledge and use sensor measurements to improve the belief. The green trail depicts the trajectory of the tracker estimates; in the discovery phase, it is gradually pulled over to the road (the red curve). Fig. 13b-d shows the progress of tracking. Although our tracker does not know the road configuration, it produces estimates which follows the road fairly closely.

In our experiment, we have observed that including two DOA sensors in the sensor network



(a) (b) (c) (d)

Figure 13: Snapshots of an AAV run on the north-east road. Plotting convention is the same as in Fig. 8. The red curves are the roads. The green curve is the estimated track.

improves tracking accuracy. This is consistent with our results in Sec. 5.1. We suspect that including a few more DOA sensors may bring further improvement. Besides, since DOA sensors essentially uses beam-crossing for localization, placing the DOA sensors evenly across the sensor field to avoid co-linearity may be advantageous.

6 Discussion and Future Work

In our experiments, we have glossed over the issue of automatically initializing a track when the vehicle enters the sensor field and instead hardwired a uniform initial probability distribution around the entry point. In the future, we plan to deploy a small number of "watchdog" sensors along the boundary of the sensor field. While most part of the network remains idle, the watchdog sensors look out for events of interest. When a high-confidence detection occurs, sensors in that neighborhood wake up, elect a leader, causing the IDSQ tracker to initialize. When the target moves outside the sensor field, the network goes back to the sleeping mode, with only the watchdogs turned on.

Bandwidth is at a premium in a sensor network. To further reduce the bandwidth requirement during sensor collaboration, we can consider switching between nonparametric and parametric belief representations depending on the nature of the distribution. The parametric form is more compact, when it is feasible, since only the parameters of a distribution need to be communicated from the

current leader to the next. Even in the nonparametric representation, one may be able to encode the distribution using image compression techniques to significantly reduce the number of bits that needs to be transmitted. Another improvement to the tracker could come from using more realistic dynamics models for sound source amplitudes and vehicles during the Bayesian filtering. We used a crude form of DOA likelihood function. More accurate, experimentally validated characterization could also help improving the tracking.

Reliability is an important issue in the sensor network. A single leader tracking protocol, while conceptually simple, might suffer from node failure or degradation. A multi-thread IDSQ tracker could partially alleviate such problems by providing some degree of redundancy. The challenge here is to maintain consistencies among and fuse information from multiple threads of the tracker. In addition to the problem of node dropout, the tracker must also accommodate sensor measurement outliers, perhaps through a local voting mechanism.

Extending the approach to tracking multiple targets faces is a significant next step. When multiple target are far apart from each other, the network can partition into sub-networks and initialize independent trackers in parallel. The difficulty arises when some targets are in proximity with each other. The source separation and data association are two major technical hurdles that must be solved. The data association module will need to exploit classification knowledge of targets in order to better disambiguate between multiple targets. The information criterion in the IDSQ tracker, in this case, must be extended to account for both state estimation as well as classification.

7 Conclusion

This paper has presented a principled approach to sensor selection and a class of signal processing algorithms for distributed sensor network, based on a mutual information measure, models of acoustic amplitude and bearing sensing, and a computationally efficient implementation of the IDSQ tracking algorithm. The tracker is distributed, energy-efficient, and scalable. The approach has been demonstrated in both simulation and on field data of a moving vehicle.

References

[1] R. Brooks, C. Griffin, and D. Friedlander, "Self-organized distributed sensor network entity tracking," *International Journal of High-Performance Computing Applications*, vol. 16, no. 3, Fall 2002.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Proc. of Int. Conf. on Mobile Computing and Networks*, (Seattle, WA), Aug. 1999.

[3] M. Chu, H. Haussecker, and F. Zhao, "Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks," *International Journal of High-Performance Computing Applications*, vol. 16, no. 3, Fall 2002.

[4] V. Poor, *An Introduction to Signal Detection and Estimation, 2nd Ed.* New York, NY: Springer-Verlag, 1994.

[5] S. Haykin, *Adaptive Filter Theory, 3rd ed.* Upper Saddle River, NJ: Prentice Hall, 1996.

[6] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics*. New York, NY: John Wiley and Sons, Inc., 1999.

[7] J. Chen, R. Hudson, and K. Yao, "Joint maximum-likelihood source localization and unknown sensor location estimation for near-field wideband signals," *Proceedings of the SPIE*, vol. 4474, July 2001.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: John Wiley and Sons, Inc., 1991.

[9] J. Ziv and M. Zakai, "On functionals satisfying a data–processing theorem," *IEEE Trans. Info. Theory*, vol. 19, pp. 275–283, May 1973.

[10] A. O. Hero, "On the problem of granulometry for a degraded Boolean image model," in *Proc. of ICIP'99*, (Kobe, Japan), pp. II. 16–20, Oct. 1999.