

Information-Driven Dynamic Sensor Collaboration for Tracking Applications

Feng Zhao, Jaewon Shin and James Reich

IEEE Signal Processing Magazine, March 2002

CS321 Class Presentation – Fall 2005

Main Points

Motivation/Premise

- Sensor collaboration [i.e., identifying nodes for (a) sensing (b) routing (c) processing, etc.] can yield tangible benefits (?)

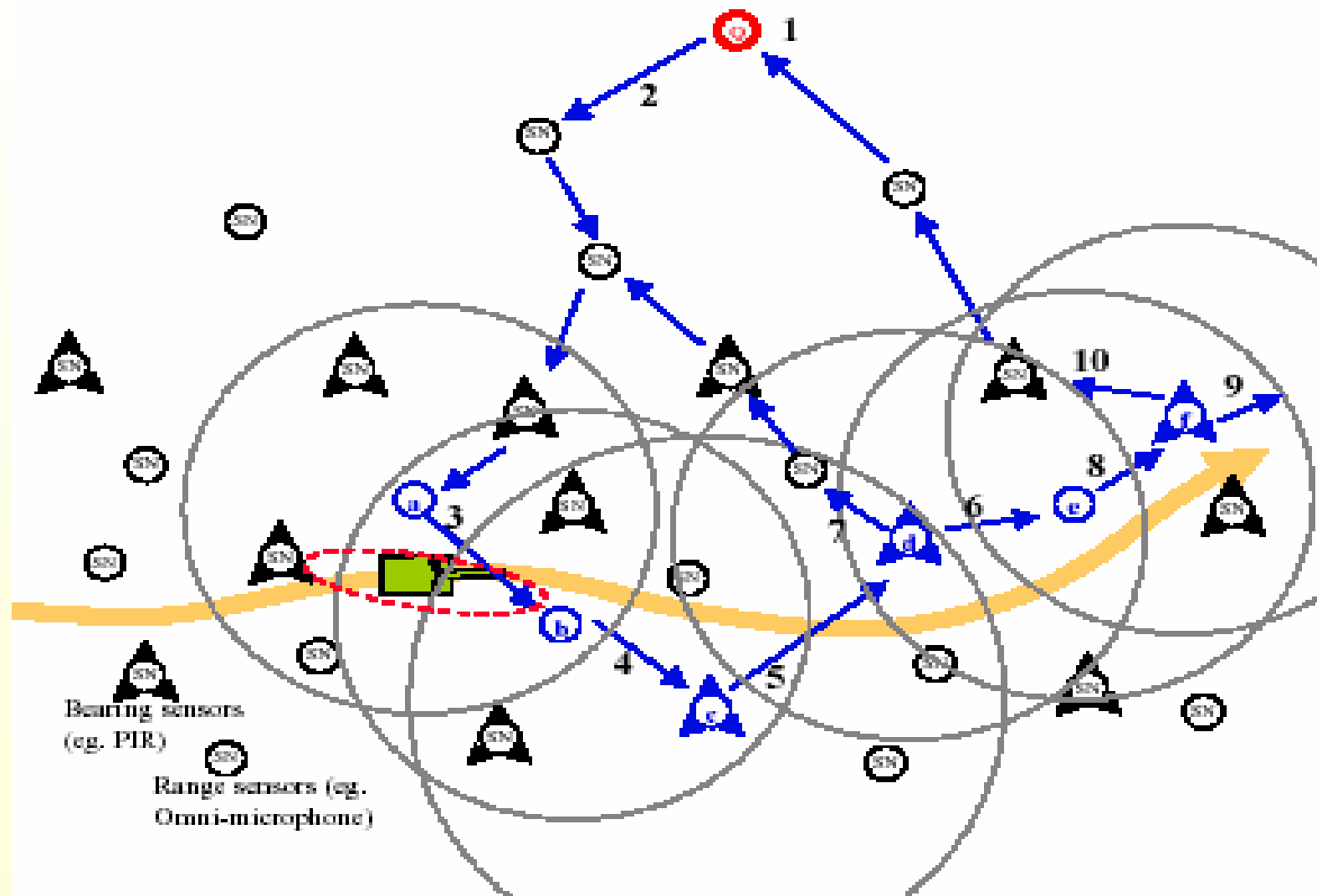
Approach

- Select next sensor to query to *maximize* information return while *minimizing* latency & bandwidth consumption
- Use **information utility measures**
 - E.g. Mahalanobis distance, volume of error covariance ellipsoid
- Incrementally query and combine sensor data
- Builds on top of Directed Diffusion idea

Collaborative Signal and Information Processing

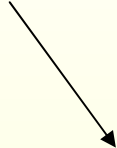
- Detection, Classification and tracking of moving and low-observable objects, requires non-local and non-trivial coordination among sensors
- A central issue is energy-constrained dynamic sensor collaboration
- Measurable benefits of sensor collaboration
 - Detection Quality: resolution, sensitivity, latency, ROC etc.
 - Track quality: tracking errors, track length and robustness
 - Survivability: Robustness against node/link failures
 - Resource usage: power and bandwidth consumption

A Tracking Scenario



Information Processing Steps

1. A user query enters the sensor network at node Q .
2. Meta knowledge guides the query towards a region of potential events.
3. Node a computes an initial estimate of vehicle state $\hat{\mathbf{x}}_a$, determines the next best sensor
 $Next(\hat{\mathbf{x}}_a, \lambda_i) = b, i \in neighbors(a)$, and hands off the state information to b . λ_i is sensor characteristics for node i .
4. Node b computes a new estimate by combining its measurement $\mathbf{z}_b$ with the previous estimate $\hat{\mathbf{x}}_a$ using, say a Bayesian filter: $\hat{\mathbf{x}}_b = \hat{\mathbf{x}}_a \oplus \mathbf{z}_b; Next = c$
5. Node c computes: $\hat{\mathbf{x}}_c = \hat{\mathbf{x}}_b \oplus \mathbf{z}_c; Next = d$
6. Node d computes: $\hat{\mathbf{x}}_d = \hat{\mathbf{x}}_c \oplus \mathbf{z}_d; Next = e$
7. Node d sends current estimate back to the querying node Q .
8. Node e computes: $\hat{\mathbf{x}}_e = \hat{\mathbf{x}}_d \oplus \mathbf{z}_e; Next = f$
9. Node f computes: $\hat{\mathbf{x}}_f = \hat{\mathbf{x}}_e \oplus \mathbf{z}_f; Next = \dots$
10. Node f sends current estimate back to querying node. ...



Knowledge of neighbors' capabilities

IDSQ: Information-Driven Sensor Querying

- Formulates distributed tracking as a sequential Bayesian estimation problem
- State of the target we wish to estimate is $\mathbf{x}$
- Choose the sensor j that is likely to provide greatest improvement in estimation with the lowest cost
- Leads to an optimization problem with the objective

$$M(p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_j)) = \alpha \cdot \varphi_{\text{Utility}}(p(\mathbf{x} | \mathbf{z}_1, \dots, \mathbf{z}_j)) + (1 - \alpha) \cdot \varphi_{\text{Cost}}(\mathbf{z}_j)$$

Information utility measure

Cost of communication and other resources

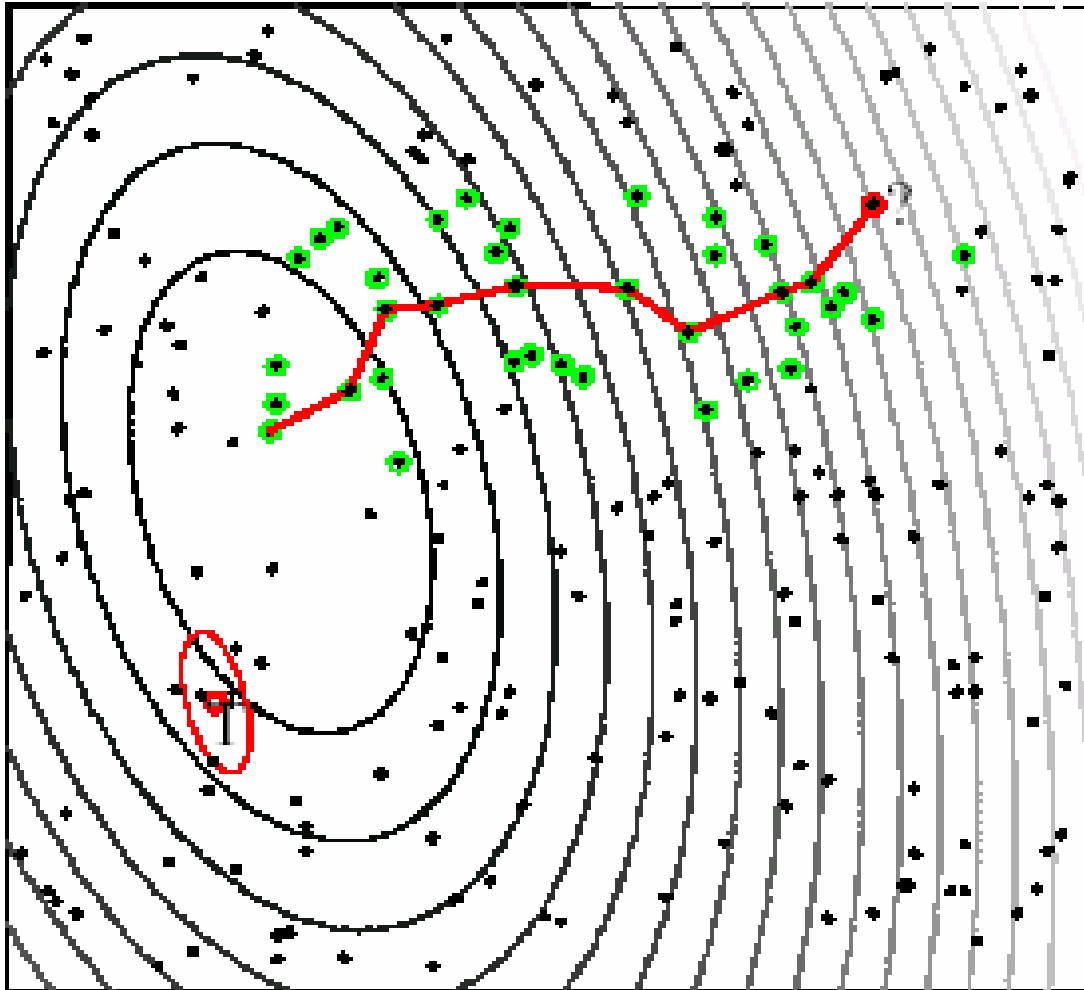
An Example Objective

- Information utility measured via Mahalanobis distance
- Cost measured as energy cost, translating to Euclidean distance

$$M(\mathbf{x}_j) = -\alpha (\mathbf{x}_j - \hat{\mathbf{x}}_r)^T \hat{\Sigma}^{-1} (\mathbf{x}_j - \hat{\mathbf{x}}_r) - (1-\alpha) (\mathbf{x}_j - \mathbf{x}_t)^T (\mathbf{x}_j - \mathbf{x}_t)$$

where $\hat{\mathbf{x}}_r, \hat{\Sigma}, \mathbf{x}_j, \mathbf{x}_t$ are the mean of the target position, its covariance, the position of queried sensor, and the position of querying sensor, respectively.

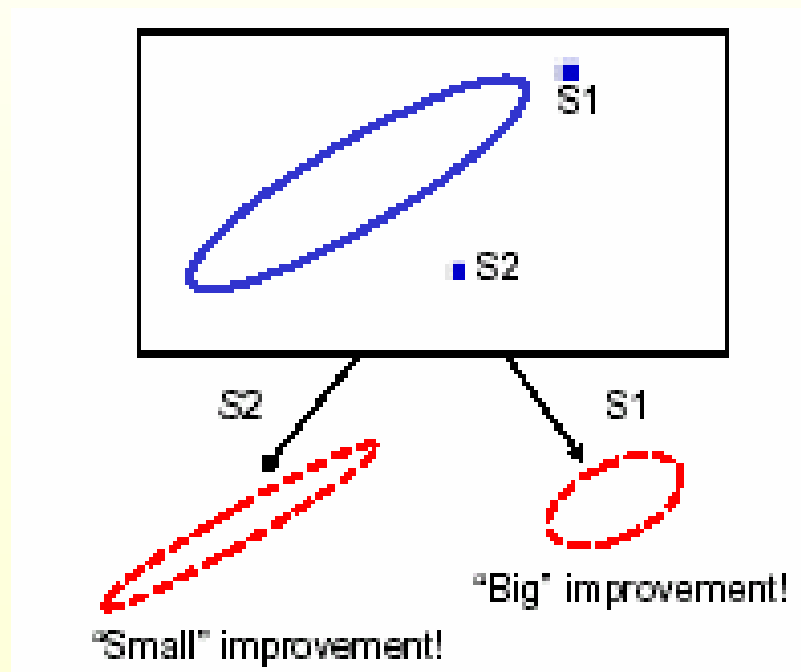
Application to Localization



- ? Is the node querying
- T is the target node
- Center of the ellipsoids is the region of interest
- Red ellipsoid is the uncertainty related to T
- Red line shows how the query is routed toward the max of objective function
- Mean and Covariance of target location are updated along the way
- Routing decisions are still made locally as these values are passed down

Sensor Selection

- Select an optimal subset of sensors and decide an optimal order of how to incorporate measurements into belief update
- Need to decide without knowing the actual measurements
 - Base decisions on *a priori* info, e.g. position, modality



Makes sense to use sensors along the major axis of the uncertainty ellipsoid

Information Utility Measures

- Entropy: Small $\rightarrow$ less randomness, more information

$$H_p(x) = -\sum_{x \in S} p(x) \log p(x)$$

$$H_p(x) = -\int_S p(x) \log p(x) dx$$

$$\varphi(p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \{\mathbf{z}_j\})) \triangleq -H_p(x).$$

U is the set of nodes already in the belief, j is a “potential node”

- Problem: Need to know the measurement before hand
- Mahalanobis distance is useful when current belief is approx. Gaussian, especially useful for range sensing with long tails

$$\varphi(\mathbf{x}_j, \hat{\mathbf{x}}, \hat{\Sigma}) = -(\mathbf{x}_j - \hat{\mathbf{x}})^T \hat{\Sigma}^{-1} (\mathbf{x}_j - \hat{\mathbf{x}})$$

where $\mathbf{x}_j$ is the position of sensor j , and $\hat{\mathbf{x}}$ is the mean of the belief (target position estimate).

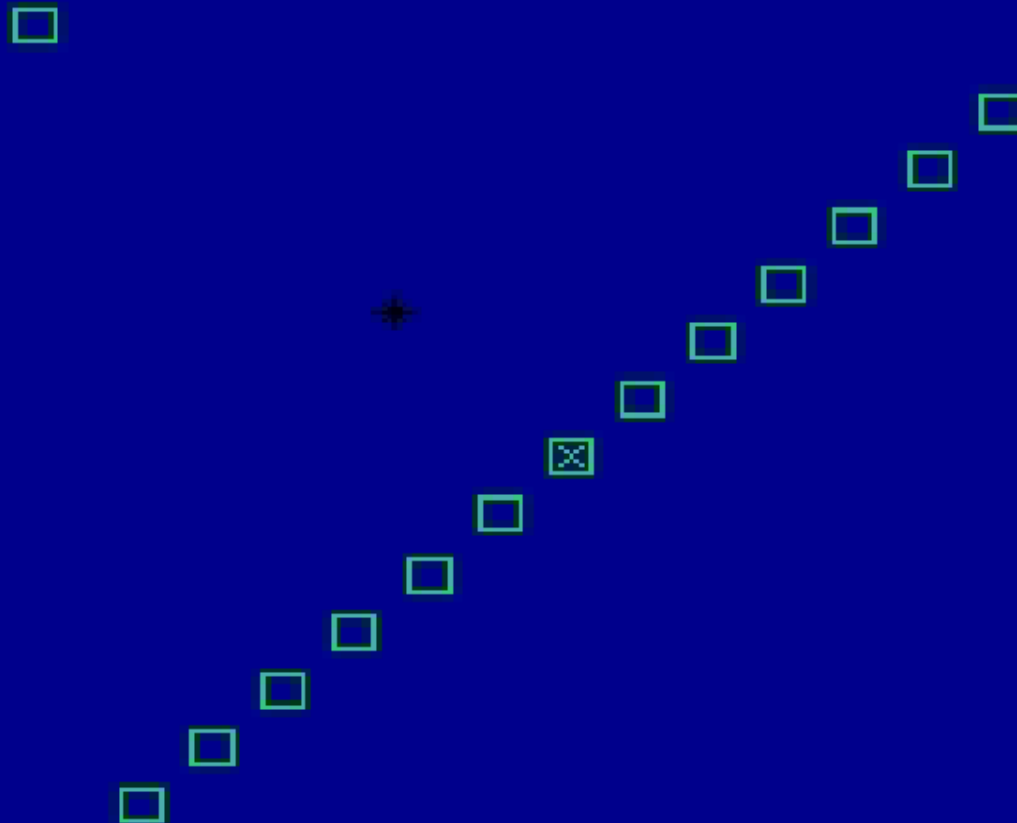
Expected Posterior Distribution

$$p(\mathbf{x}^{(t+1)} | \overline{\mathbf{z}^{(t+1)}}) = \\ C \cdot p(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)}) \cdot \int p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}) \cdot p(\mathbf{x}^{(t)} | \overline{\mathbf{z}^{(t)}}) d\mathbf{x}^{(t)}$$

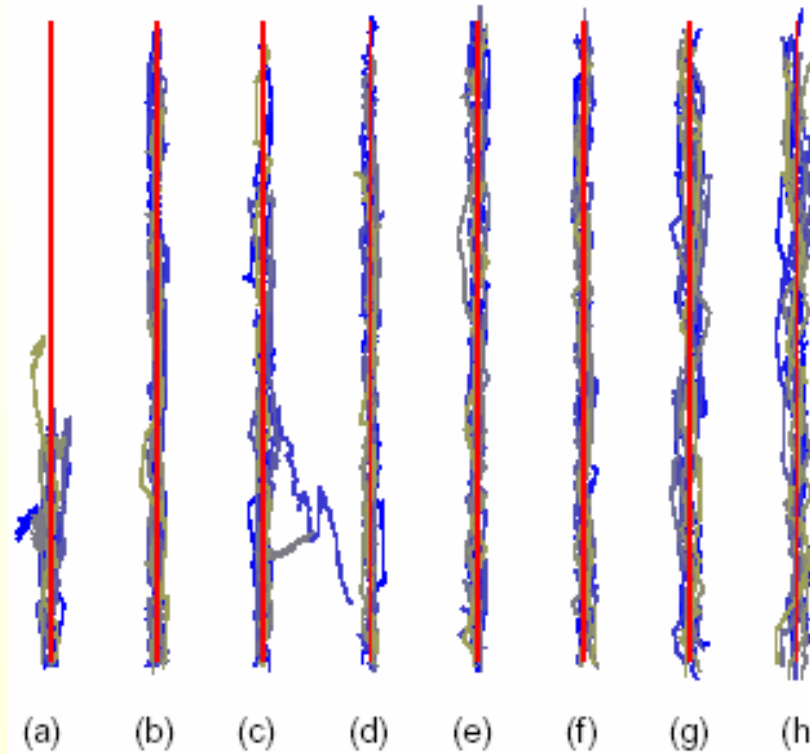
where $p(\mathbf{x}^{(t)} | \overline{\mathbf{z}^{(t)}})$ is the current belief given a history of the measurement up to time t : $\overline{\mathbf{z}^{(t)}} = \{\mathbf{z}^{(0)}, \dots, \mathbf{z}^{(t)}\}$, $p(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)})$ specifies the predefined dynamics model, and $p(\mathbf{z}_j^{(t+1)} | \mathbf{x}^{(t+1)})$ is the likelihood function from the measurement of sensor j .

The above non-parametric representation of belief state allows non-Gaussian distributions and nonlinear dynamics

Illustration of Localization Using Mahalanobis Distance



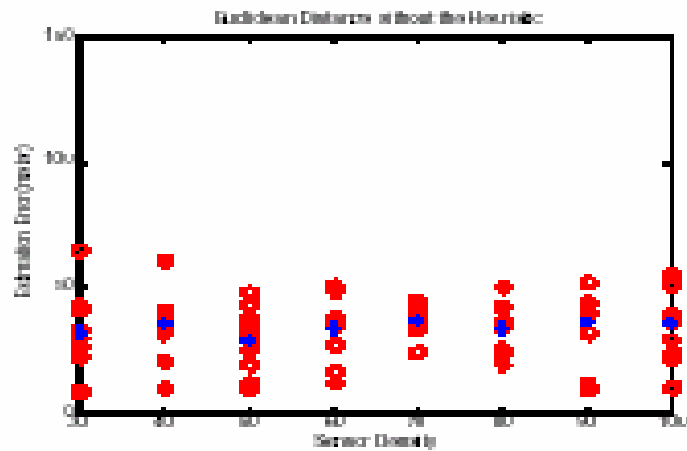
Tracking a moving object [non-Gaussian distribution]



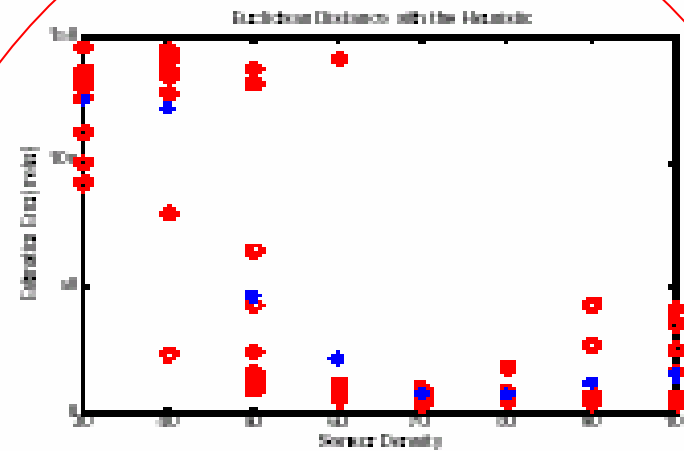
Heuristic prevents
each node from
being selected
more than N times

(a) Nearest neighborhood. (b) Nearest neighborhood with the heuristic. (c) Mahalanobis distance. (d) Mahalanobis distance with the heuristic. (e) Entropy. (f) Entropy with the heuristic. (g) Relative entropy (KL-distance). (h) Relative entropy with the heuristic.

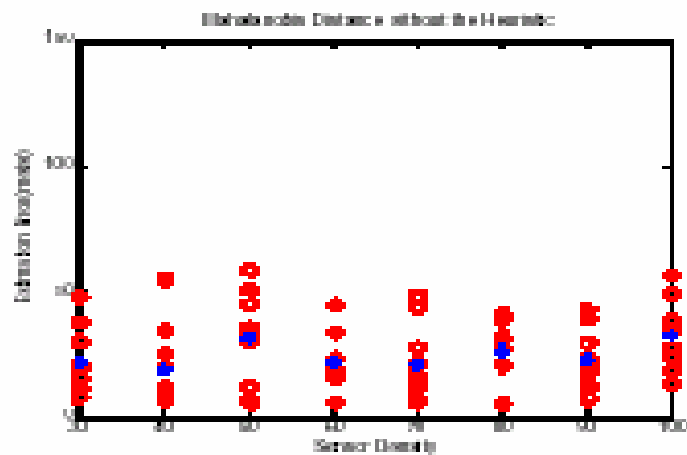
Tracking error vs Sensor Density



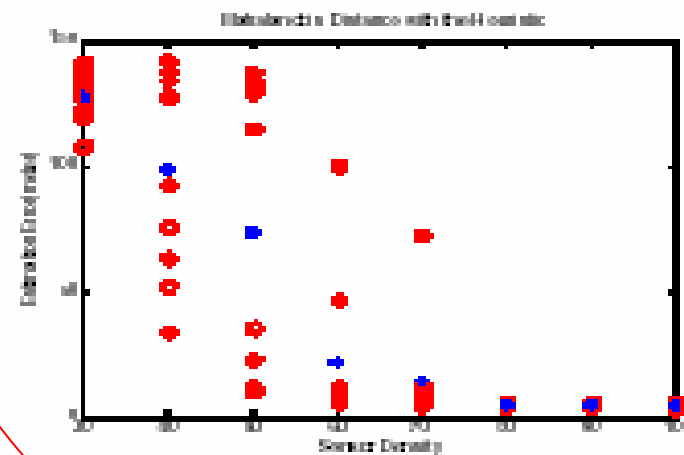
(a)



(b)



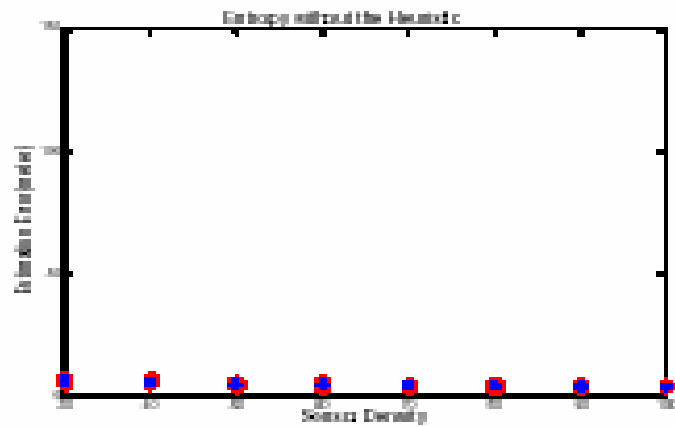
(c)



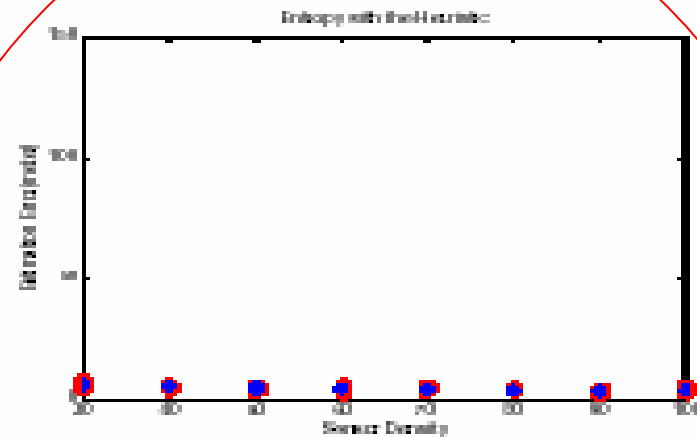
(d)

With heuristic

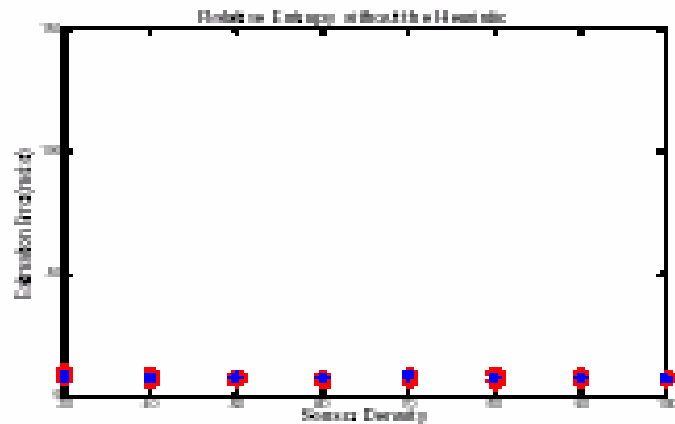
Tracking error vs Sensor Density, Cont'd



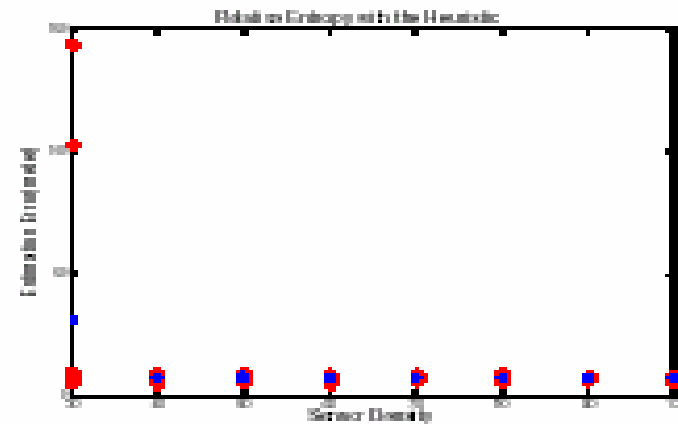
(e)



(f)



(g)



(h)

With heuristic

Statistics

	# Lost tracks/ total runs	Mean error
(a) Nearest neighbor	75/80	34.39
(b) Nearest neighbor with heuristic	37/80	44.79
(c) Mahalanobis distance	70/80	24.86
(d) Mahalanobis distance with heuristic	32/80	44.20
(e) Entropy	0/80	5.13
(f) Entropy with heuristic	0/80	5.05
(g) Relative entropy	0/80	8.09
(h) Relative entropy with heuristic	2/80	10.79

Open Issues

- Representation of belief state
 - Parametric, Particle, Grid
- Sequential vs. concurrent Information Exchange
 - Multiple leaders
- Query types
 - Taxonomy, in-network processing styles
- Bias in sensor selection
 - Proper representation of prior probability distributions
- Information quality vs Cost tradeoff
 - Not clear how the tracking quality improves if there were no cost constraints
- **Base paper:** M. Chu, H. Haussecker and F. Zhao, “Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks”, *Int’l J. of High Performance Computing Applications*, 2002.