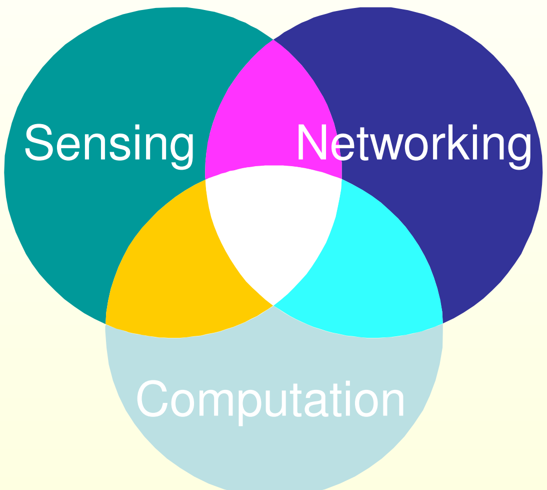
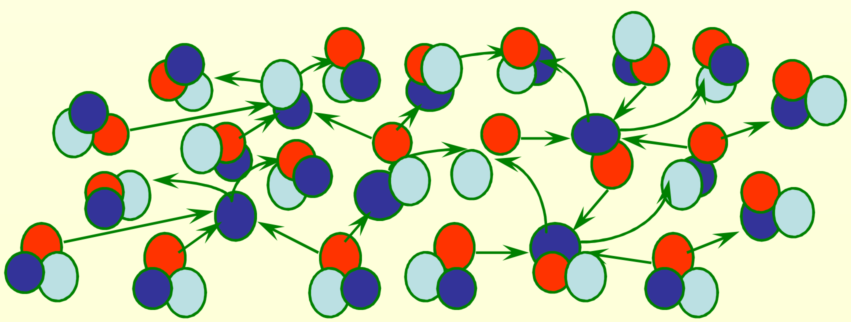
CS321: State Estimation and Tracking

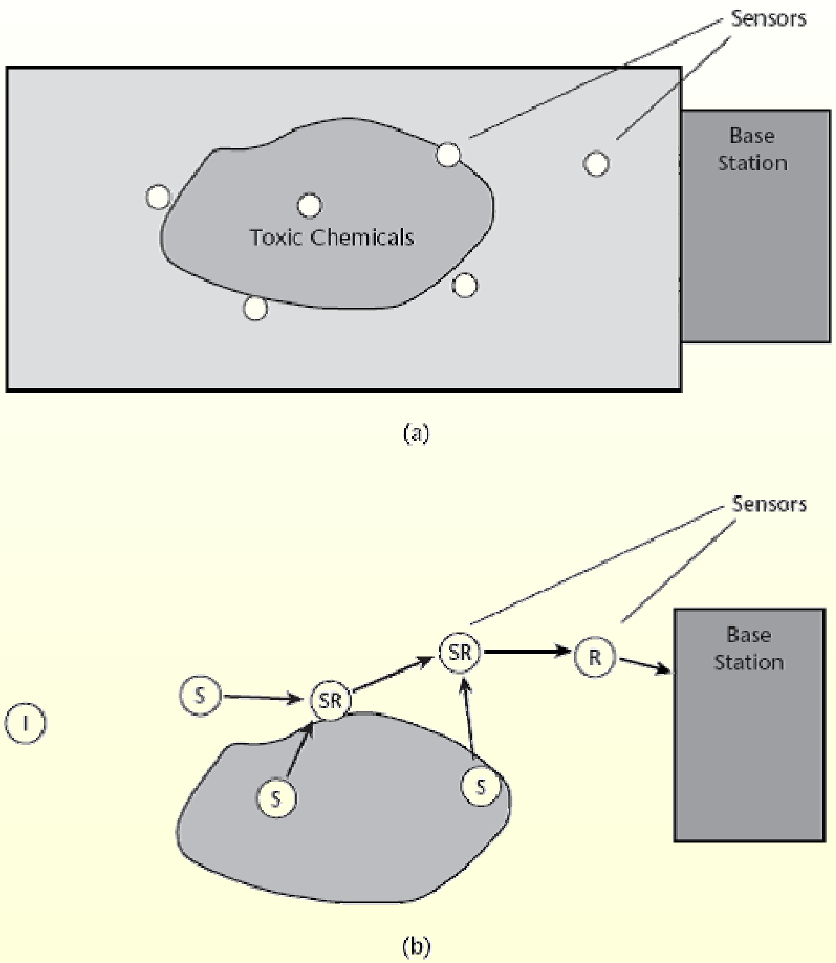


Leonidas Guibas
Computer Science Dept.
Stanford University



Node Roles in a WSN

- Nodes can play multiple roles in a sensor network
  - sensing
  - routing
  - aggregating information
  - monitoring
  - sleeping
- When tasking sensors, we must be aware of all the potential roles these nodes they can play



Papers/Books

- S. Thrun, W. Burgard, and D. Fox. **Probabilistic Robotics**. MIT Press, Cambridge, MA, 2005.
- Maurice Chu, Horst Haussecker, and Feng Zhao, Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks. Int'l J. **High Performance Computing Applications**, 16(3):90-110.
- F. Zhao, J. Liu, J. Liu, L. Guibas, and J. Reich, Collaborative Signal and Information processing: an Information-Directed Approach, **Proceedings of the IEEE**, 91, 8, pp. 1199- 1209, 2003.
- Q. Fang, F. Zhao, and L. Guibas, Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation, the 4th ACM International **Symposium on Mobile Ad Hoc Networking & Computing** (MobiHoc), 2003, pp. 165-176.
- J. Shin, L. Guibas and F. Zhao, Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-hoc Sensor Networks. 2nd Int'l Workshop on **Information Processing in Sensor Networks** (IPSN) 2003, pp. 223-238.

System State

Continuous and Discrete Variables

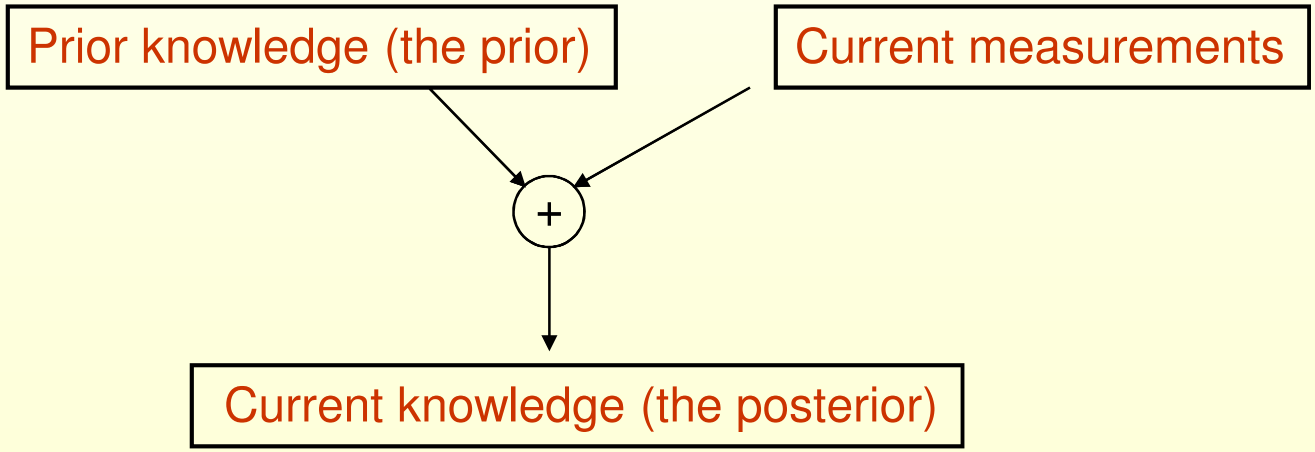
- The quantities that we may want to estimate using a sensor network can be either continuous or discrete
- Examples of continuous variables include
  - a vehicle's position and velocity
  - the temperature in a certain location
- Examples of discrete variables include
  - the presence or absence of vehicles in a certain area
  - the number of peaks in a temperature field

Uncertainty in Sensor Data

- Quantities measured by sensors always contain errors and have associated uncertainty – thus they are best described by PDFs.
  - interference from other signal sources in the environment
  - systematic sensor bias(es)
  - measurement noise
- The quantities we are interested in may differ from the ones we can measure – the former can only indirectly be inferred from sensor data. These hidden variables are also best described by PDFs.

Information Sources

- Prior information, together with knowledge of the physical laws for the system of interest
- Current sensor measurements



Recursive State Estimation

- State x:
  - external parameters describing the environment that are relevant to the sensing problem at hand (say vehicle locations in a tracking problem)
  - internal sensor settings (say the direction a pan/tilt camera is aiming)

While internal state may be readily available to a node, external state is typically hidden – *it cannot be directly observed but only indirectly estimated*.

States may only be known probabilistically.

Environmental Interaction

- Control u:
  - a sensor node can change its internal parameters to improve its sensing abilities
- Observation z:
  - a sensor node can take various measurements of the environment
- Discrete Time t: 0, 1, 2, 3, ...

$$x_t, u_t, z_t \qquad\qquad z_{t_1:t_2} \equiv z_{t_1}, z_{t_1+1}, z_{t_1+2}, z_{t_1+3}, \cdots, z_{t_2}$$

Basic Probability

- Random variables (discr. or cont.) and probabilities

$$p(X = x), \quad \sum_x p(x) = 1 \quad \text{or} \quad \int_x p(x)\,dx = 1$$

- Independence of random variables

$$P(X = x, Y = y) = p(x, y) = p(x)\,p(y)$$

- Conditional probability

$$p(x \mid y) = p(x, y)/p(y) \quad (= p(x) \text{ if } x \text{ and } y \text{ are independent})$$

$$p(x) = \sum_y p(x \mid y)\,p(y) \quad \text{(discrete case)}$$

$$p(x) = \int_y p(x \mid y)\,p(y)\,dy \quad \text{(continuous case)}$$

Bayes Rule

$$p(x \mid y) = p(y \mid x)\, p(x) \big/ p(y)$$

$$= \frac{p(y \mid x)\, p(x)}{\sum_{x'} p(y \mid x')\, p(x')} \quad \text{(discrete)}$$

$$= \frac{p(y \mid x)\, p(x)}{\int_{x'} p(y \mid x')\, p(x')\, dx'} \quad \text{(continuous)}$$

$$p(x \mid z) = \eta\, p(z \mid x)\, p(x)$$

probability of state *x*, given measurement *z*

probability of measurement *z*, given state *x* (the environment/ sensor model)

Expectation, Covariance, Entropy

- Expectation

$$E(X) = \sum_x x\, p(x) \;\text{ or }\; \int_x x\, p(x)\, dx \quad \Bigg| \quad E(aX+b) = aE(X) + b$$

- Covariance (or variance)

$$\mathrm{Cov}(X) = E(X - E(X))^2 = E(X^2) - E(X)^2$$

- Entropy

$$H(X) = E(-\lg p(X)) = -\sum_x p(x) \lg p(x)$$

Probabilistic Generative Laws

- State x_t is generated stochastically by

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t})$$

- *Markovian assumption* (state completeness)

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t)$$

The Bayes Filter

- Belief distributions

the prior belief

the posterior belief

$$b(x_t) = p(x_t \mid z_{1:t}, u_{1:t}), \quad \overline{b}(x_t) = p(x_t \mid z_{1:t-1}, u_{1:t})$$

- Algorithm Bayes_Filter $(b(x_{t-1}), u_t, z_t)$

for all x_t do

$\overline{b}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, b(x_{t-1})\, dx$ [prediction]

$b(x_t) = \eta\, p(z_t \mid x_t)\, \overline{b}(x_t)$ [observation]

endfor

return $b(x_t)$

Parametric PDFs: Gaussian Filters

- Beliefs are represented by multivariate Gaussian distributions

$$p(x) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

Here μ is the mean of the state, and Σ its covariance

- Appropriate for unimodal distributions

The Kalman Filter

- Next state probability must be a linear function, with added Gaussian noise [result still Gaussian]

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$$ ← Gaussian noise with zero mean and covariance R_t

$$p(x_t \mid u_t, x_{t-1}) = \det(2\pi R_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_t - A_t x_{t-1} - B_t u_t)^T R_t^{-1} (x_t - A_t x_{t-1} - B_t u_t)\right\}$$

- Measurement probability must also be linear in it arguments, with added Gaussian noise

$$z_t = C_t x_t + \delta_t$$ ← Gaussian noise with zero mean and covariance Q_t

$$p(z_t \mid x_t) = \det(2\pi Q_t)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(z_t - C_t x_t)^T Q_t^{-1} (z_t - C_t x_t)\right\}$$

Kalman Filter Algorithm

- Algorithm Kalman_Filter $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$
$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

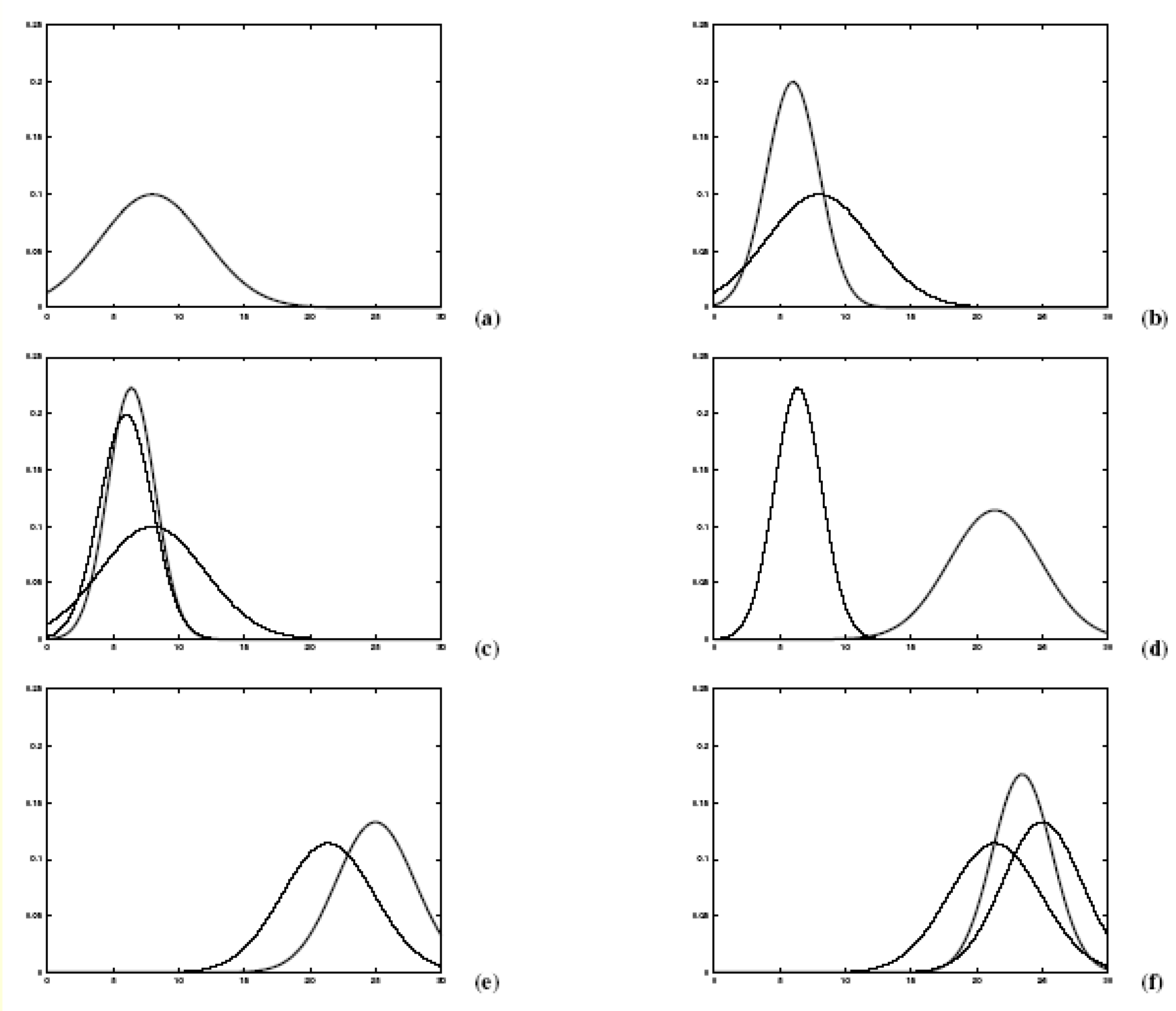
belief predicted by system dynamics

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q)^{-1} \quad \text{(the Kalman gain)}$$

$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

belief updated using measurements

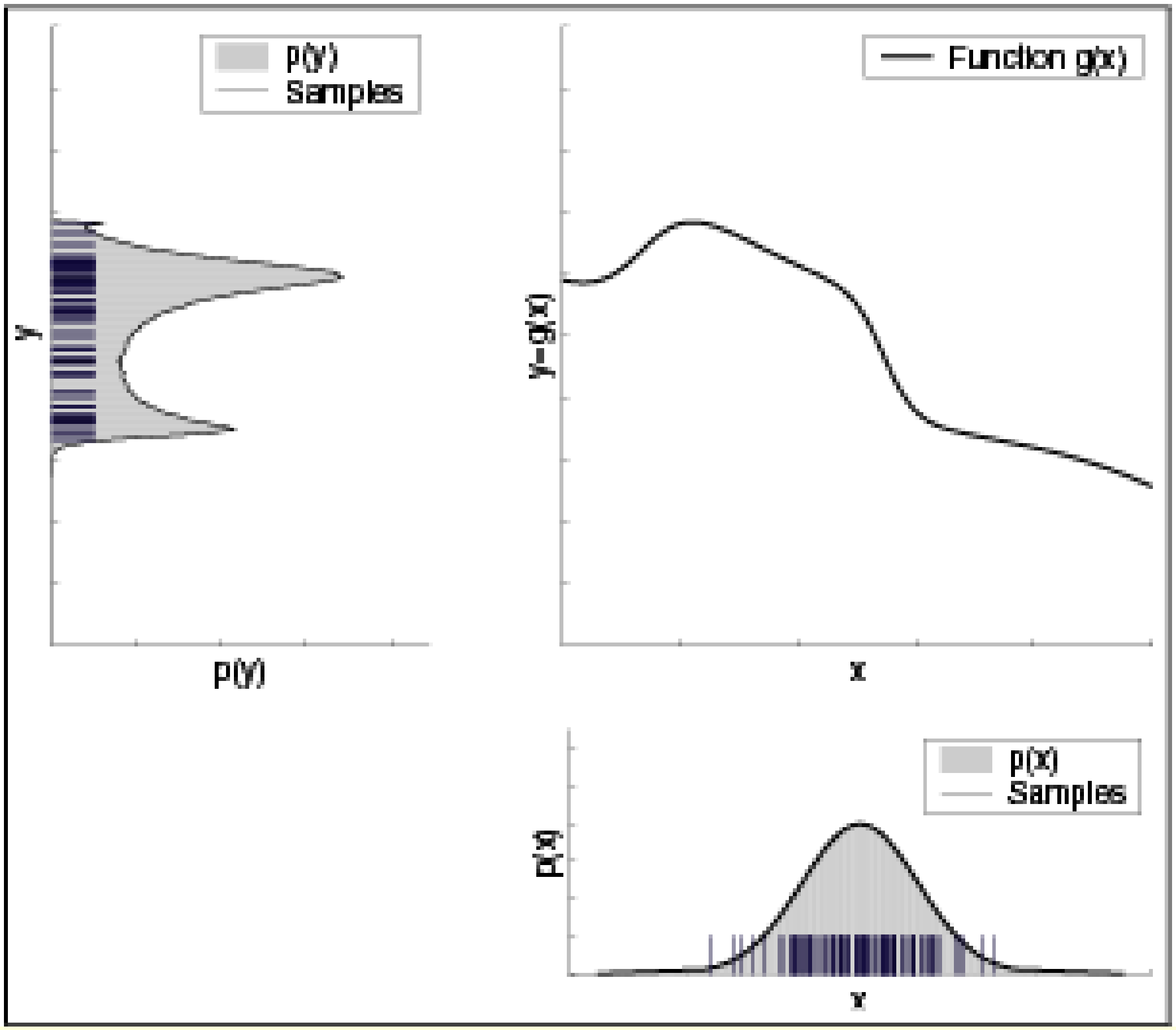
Kalman Filter Illustration



Non-Parametric Filters

- Parametric filters parametrize a distribution by a fixed number of parameters (mean and covariance in the Gaussian case)
- Non-parametric filters are discrete approximations to continuous distributions, using variable size representations
  - essential for capturing more complex distributions
  - do not require prior knowledge of the distribution shape

The Particle Filter



Samples from a Gaussian, passed through a non-linear function

The Particle Filter Algorithm

- Algorithm Particle_Filter (X_{t-1}, u_t, z_t)

$X_t = \overline{X}_t = \varnothing$

for $m = 1$ to M do

sample $x_t^{[m]} \in p(x_t \mid u_t, x_{t-1}^{[m]})$

$w_t^{[m]} = p(z_t \mid x_t^{[m]})$; $\overline{X}_t = \overline{X}_t \cup \left\langle x_t^{[m]}, w_t^{[m]} \right\rangle$

endfor

for $m = 1$ to M do

draw i with probability proportional to $w_t^{[i]}$

add $x_t^{[i]}$ to X_t

return X_t

number of particles of unit weight

stochastic propagation

importance weights

resampling, or
importance sampling

Sensor Models

Sensor Models

- To be able to develop protocols and algorithms for sensor networks, we need sensor models
  - To perform Bayesian inversion, we need to compute the value of a measurement, given an assumed state
  - We need to assess the importance of a sensor measurement
- Our state PDF representations must allow expression of the state ambiguities inherent in the sensor data
- Need also to be aware of the effect of sensor characteristics on system performance
  - cost, size, sensitivity, resolution, response time, energy use, calibration and installation ease, etc.

Sensor Characteristics

Accuracy: The difference between the measured value and the actual value, reported as a maximum.

Precision: The difference between the instrument's reported values during repeated measurements of the same quantity.

Resolution: The smallest increment of change in the measured value that can be determined from the instruments read out. Usually similar or smaller than precision.

Sensitivity The change in the output of an instrument per unit change in the input.

Ex.: Acoustic Amplitude Sensors

- Lossless isotropic propagation from a point source

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

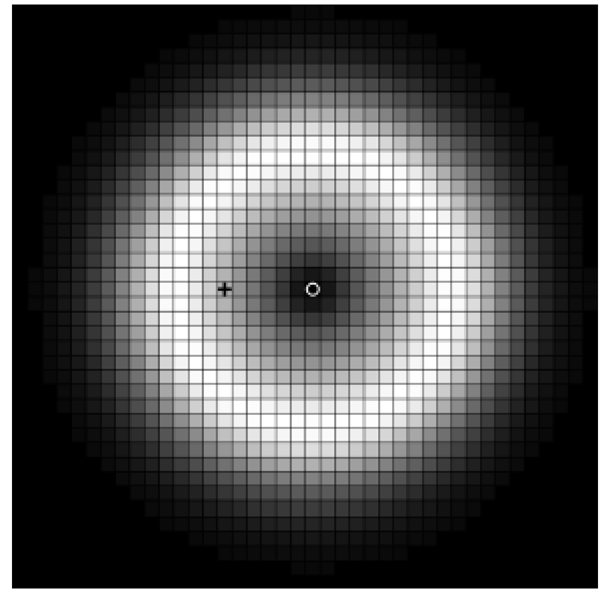
- Say w is Gaussian $N(0,\sigma)$, and a uniform in $[a_{lo}, a_{hi}]$

$$p(z \mid x) = \frac{r}{\Delta_a}\left[\Phi\left(\frac{a_{hi} - rz}{r\sigma}\right) - \Phi\left(\frac{a_{lo} - rz}{r\sigma}\right)\right]$$

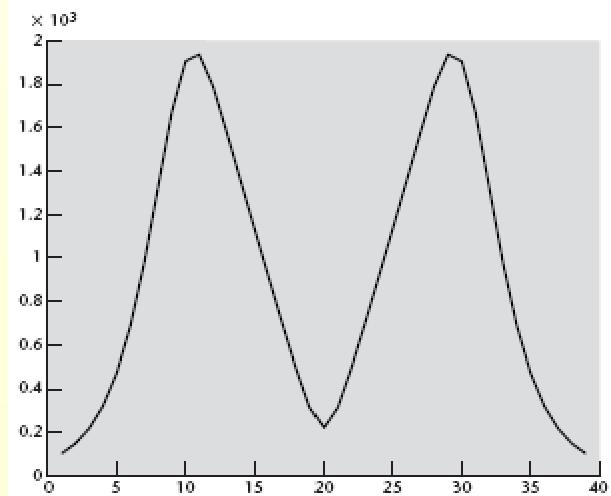
error function

$$\Delta_a = a_{hi} - a_{lo}$$

$$r = \| z - \zeta \|$$



(a)



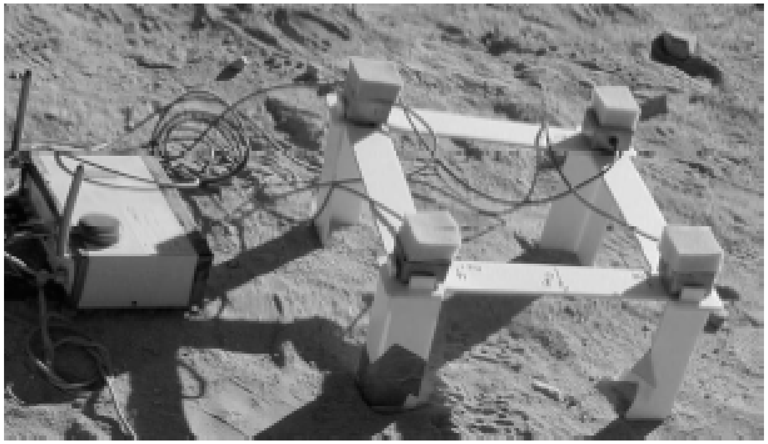
(b)

Ex.: Direction of Arrival (DoA) Sensors

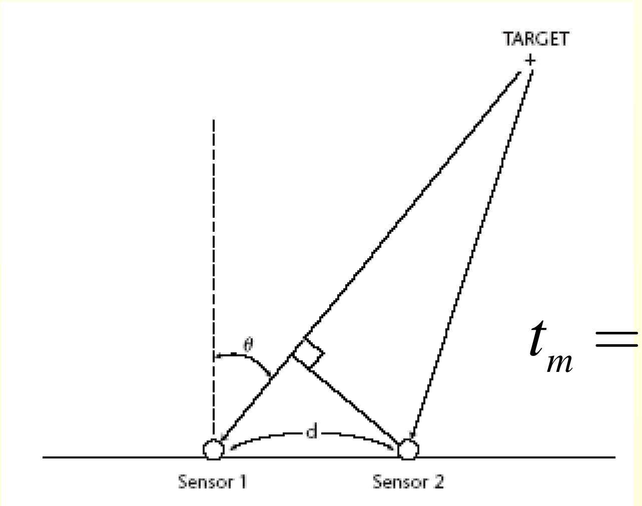
- Beam-forming with microphone arrays

$$g_m(t) = s_0(t - t_m) + w_m(t)$$

- Far field assumption

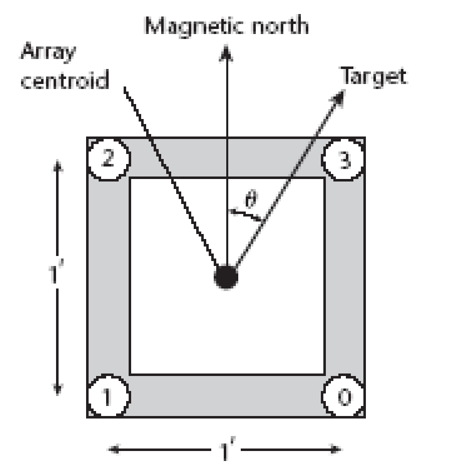


(a)



$$t_m = \frac{d}{c} \sin\theta$$

sound propagation speed



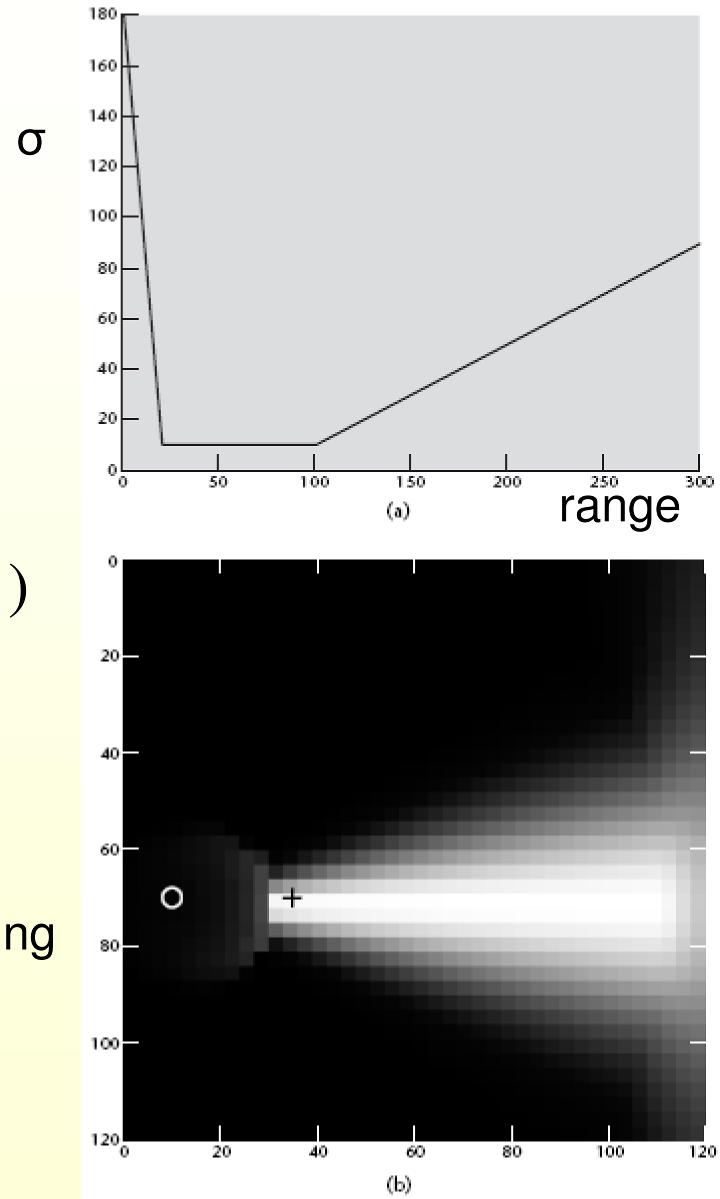
(b)

Beamforming Error Landscape

- Direction estimates are only accurate within a certain range of distances from the sensor

$$p(z \mid \theta) = (1/\sqrt{2\pi\sigma^2})\exp(-(z-\theta)^2/2\sigma^2)$$

PDF for beamforming sensor



Value of Information

What is the Value (Importance) of a Sensor Reading?

- Sensing operations cost energy – for both sensing and communication
- To properly task sensors, we need to estimate the cost of sensing ...
- But also the value of information to be potentially obtained – this is application dependent
  - but without making the measurement first!
  - information value must be based on prior beliefs and known sensor characteristics
  - it always carries uncertainty

Information Theoretic Metrics: Conditional Entropy & Mutual Information

- Entropy is a measure of uncertainty.
- Let $H(x)$ be the entropy of x based on previous observations.
- Let y be a new observation on x.
- We can measure the uncertainty of x after observing y by using the conditional entropy which is defined as:

$$H(x|y) = H(x,y) - H(y) \text{ with the property } 0 \leq H(x|y) \leq H(x).$$

- Here, $H(x,y)$ is the joint entropy of observations x and y.
- The conditional entropy $H(x|y)$ represents the amount of uncertainty remaining about x after y has been observed.
- If the uncertainty is reduced, then there is information gained by observing y. So we can measure the relevance of y by using conditional entropy.
- Another measure that is related to conditional entropy is mutual information $I(x,y)$ which is a measure of uncertainty that is resolved by observing y and is defined:

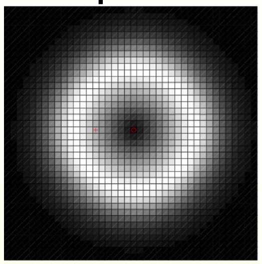
$$I(x,y) = H(x) - H(x|y) = H(x) + H(y) - H(x,y)$$

An Example Problem: Distributed State Estimation in Tracking

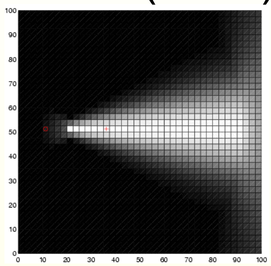
An Example Sensor Network Problem

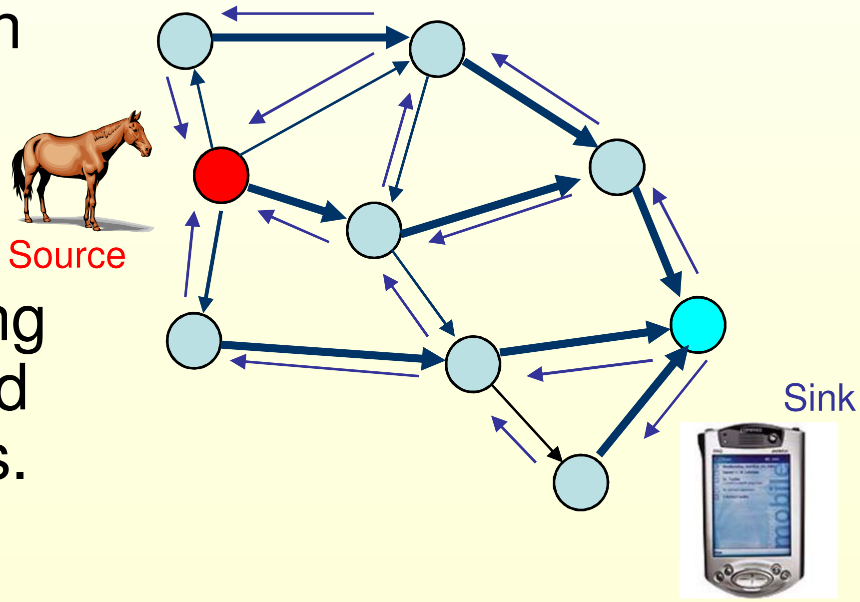
- One or more targets are moving through a sensor field
- The field contains networked acoustic amplitude and bearing (DoA) sensors
- Queries requesting information about object tracks may be injected at any node of the network
- Queries may be about reporting all objects detected, or focused on only a subset of the objects.

Acoustic Amplitude



Direction of Arrival (DOA)



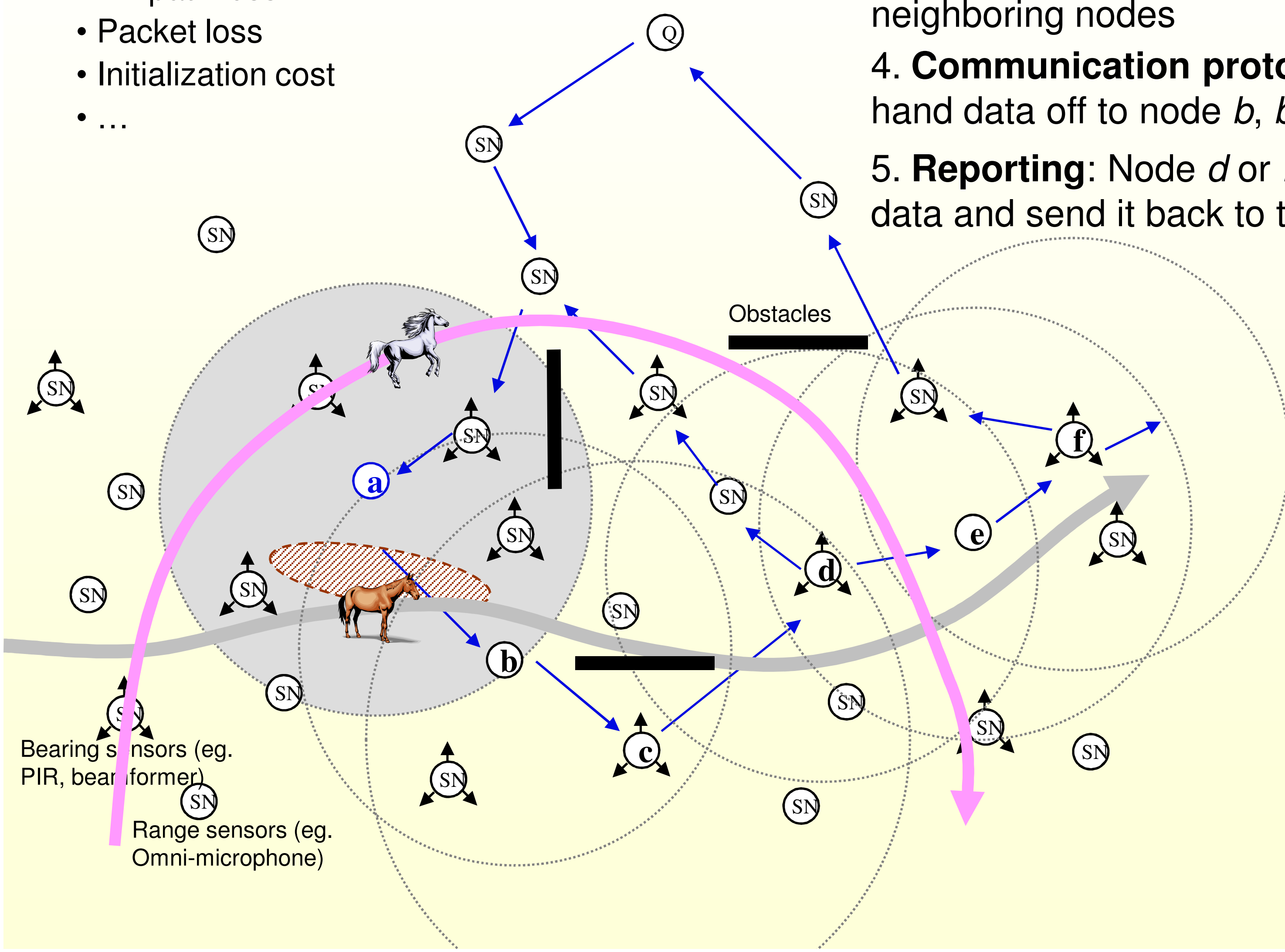


The Tracking Scenario

Constraints:

- Node power reserves
- RF path loss
- Packet loss
- Initialization cost
- …

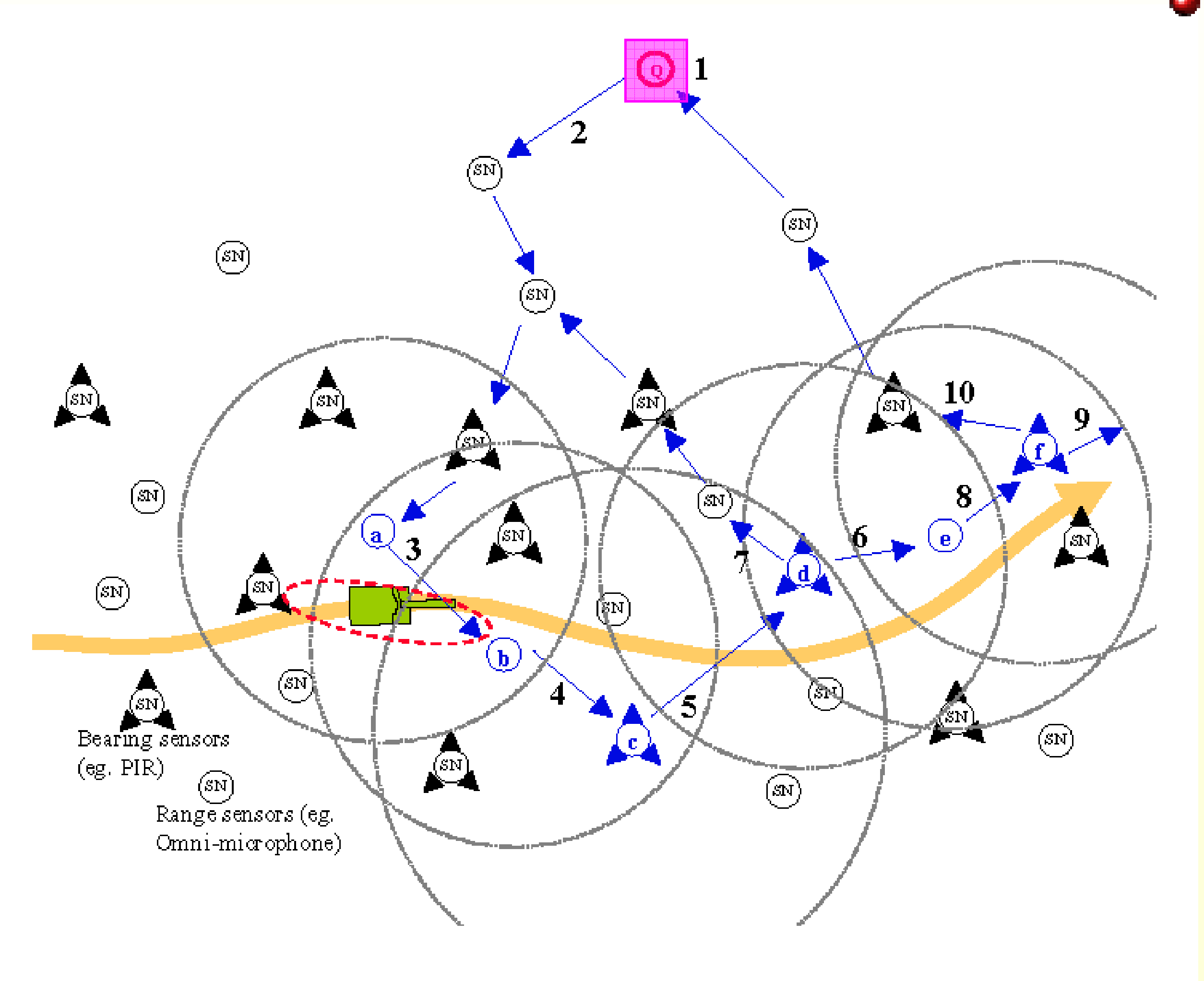
1. **Discovery**: Node *a* detects the target and initializes tracking
2. **Query processing**: User query *Q* enters the net and is routed towards regions of interest
3. **Collaborative Processing**: Node *a* estimates target location, with help from neighboring nodes
4. **Communication protocol**: Node *a* may hand data off to node *b*, *b* to *c*, …
5. **Reporting**: Node *d* or *f* summarizes track data and send it back to the querying node



What if there are other (possibly) interfering targets?

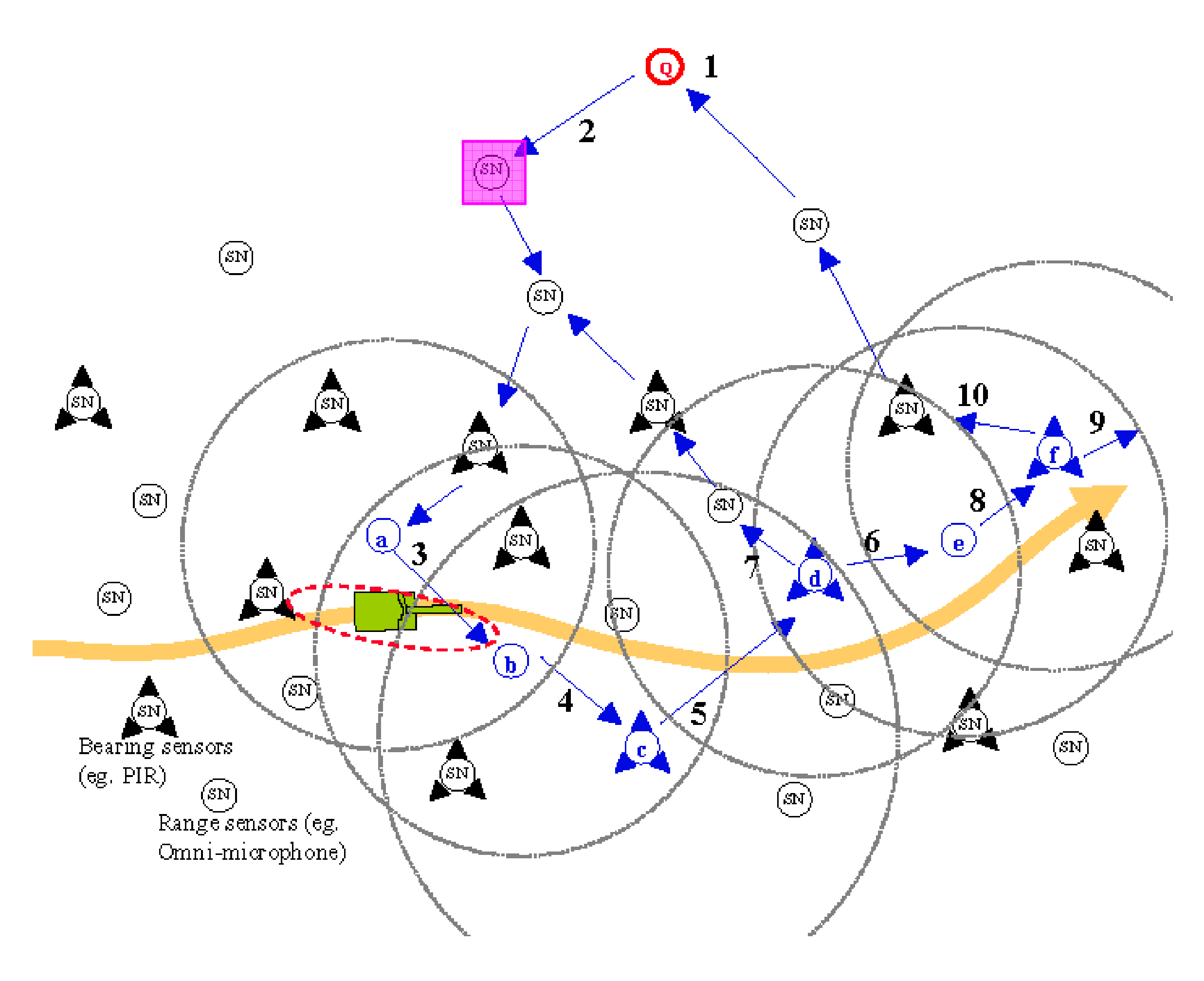
What if there are obstacles?

Tracking Scenario

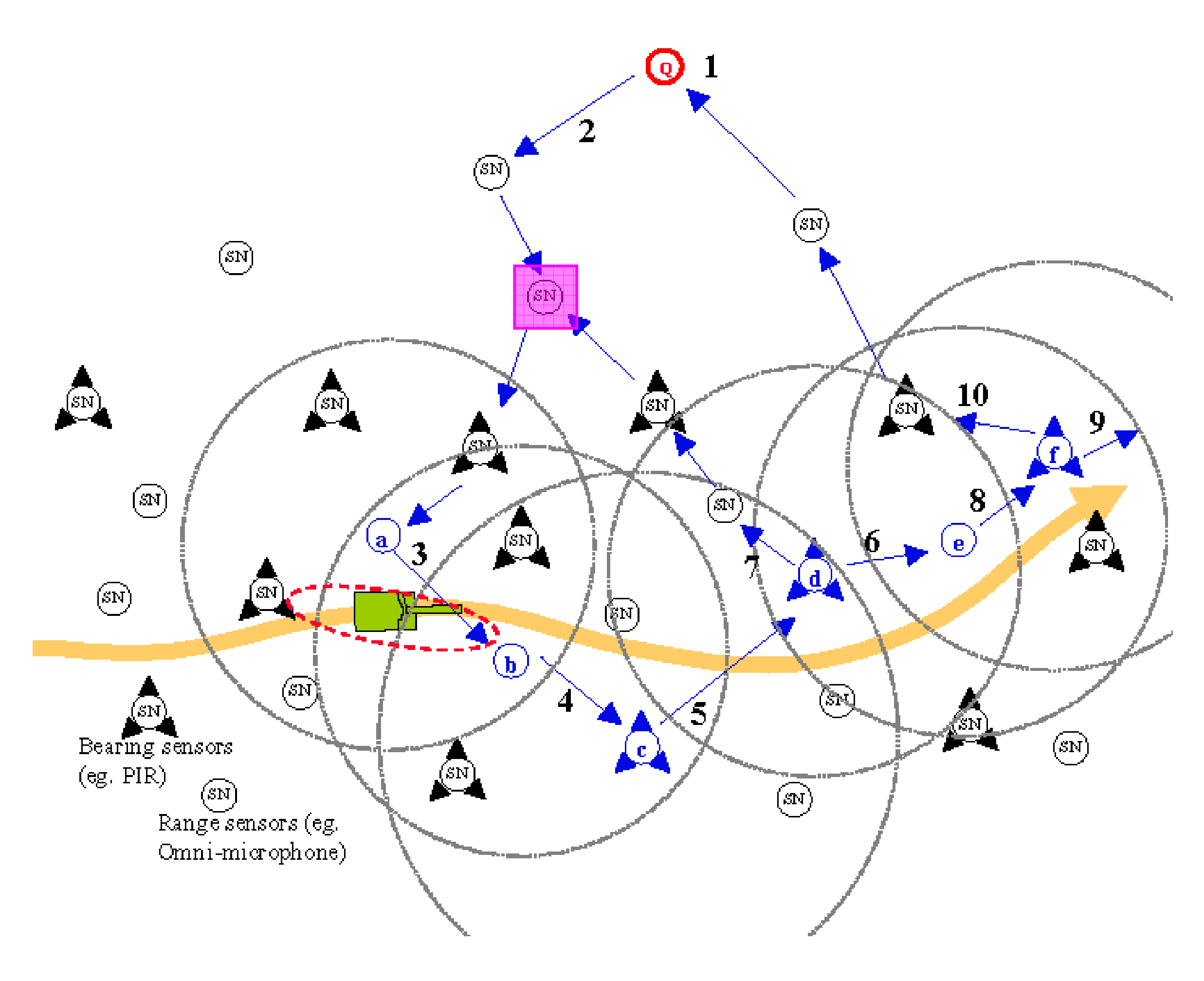


- Query must be routed to the node best able to answer it

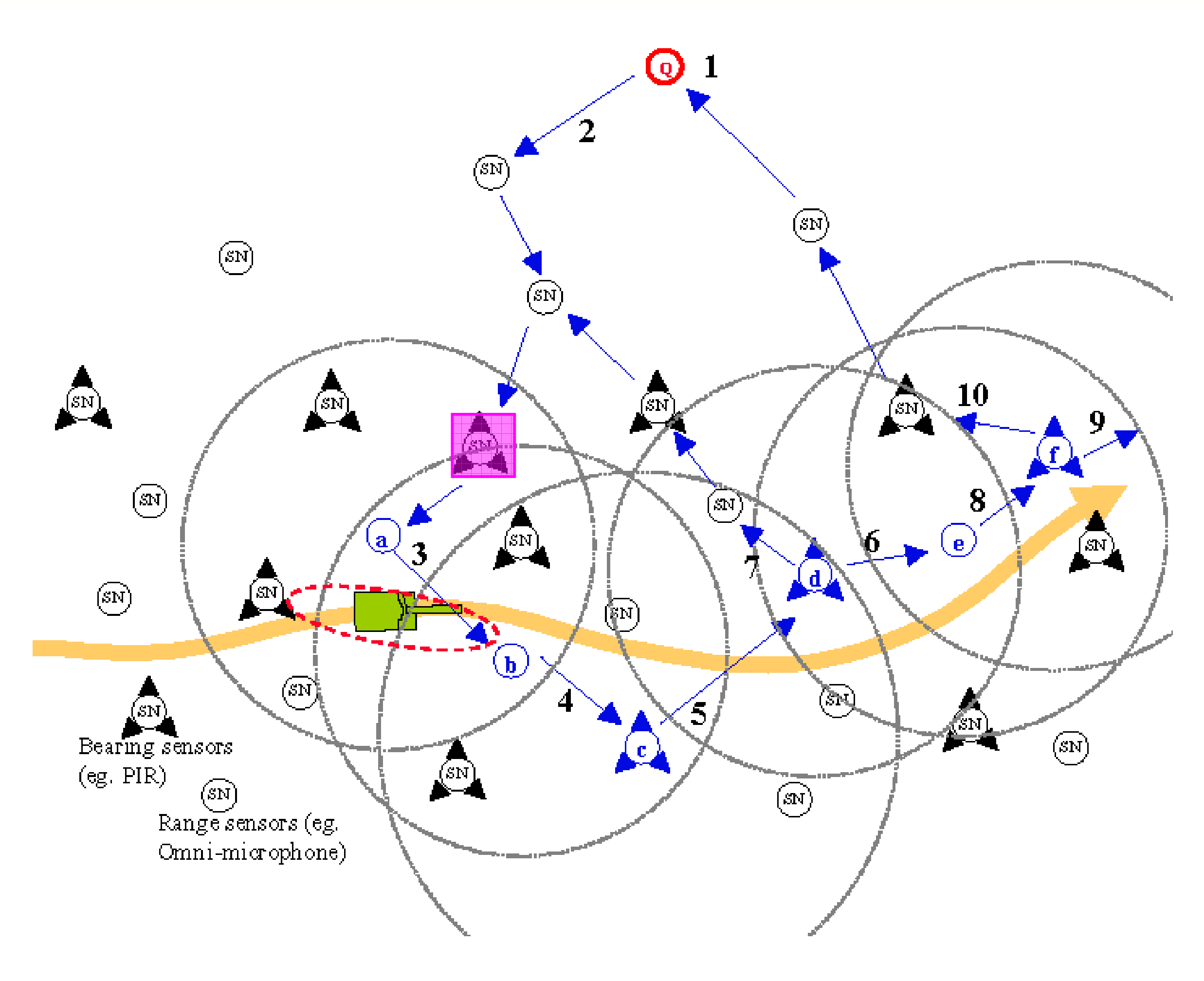
Tracking Scenario



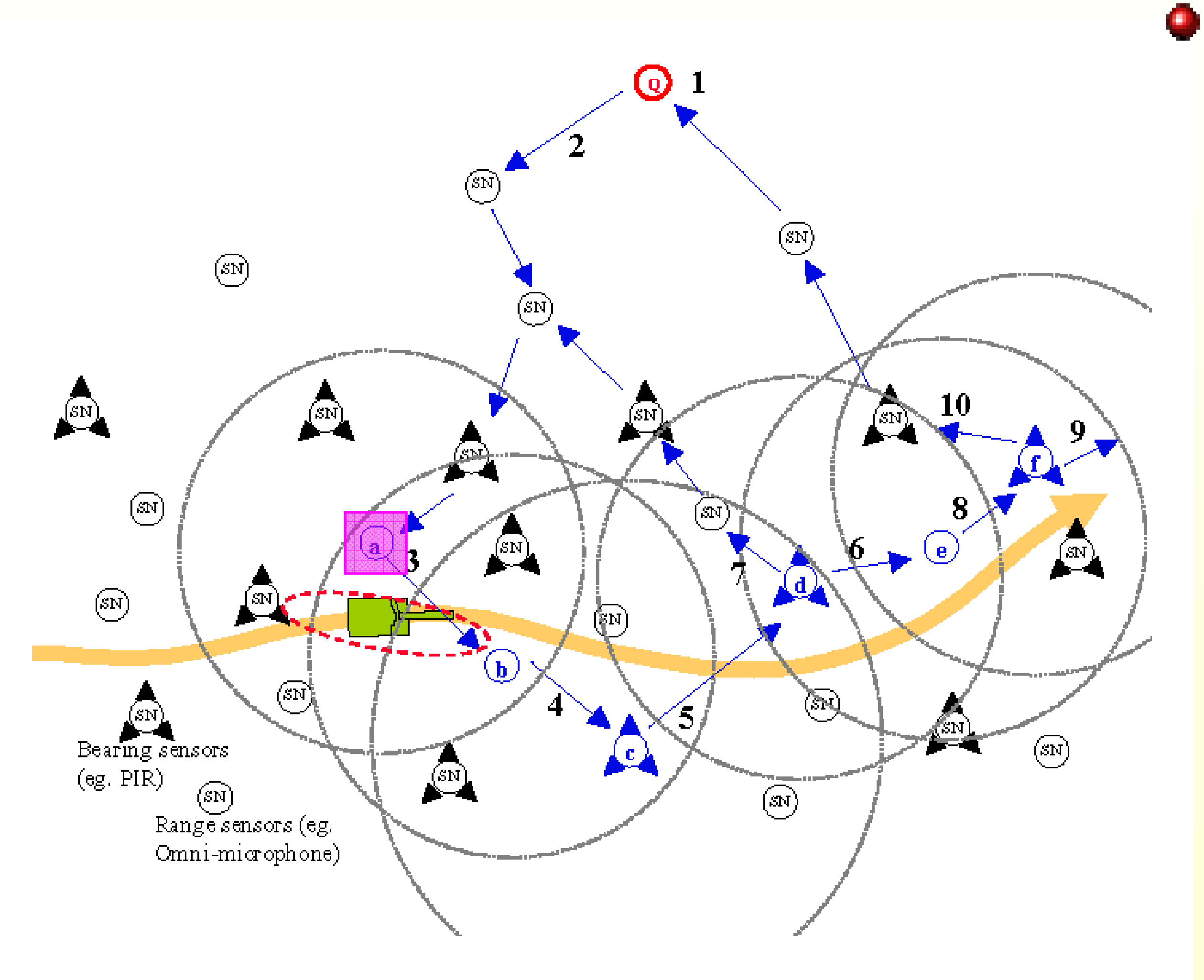
Tracking Scenario



Tracking Scenario

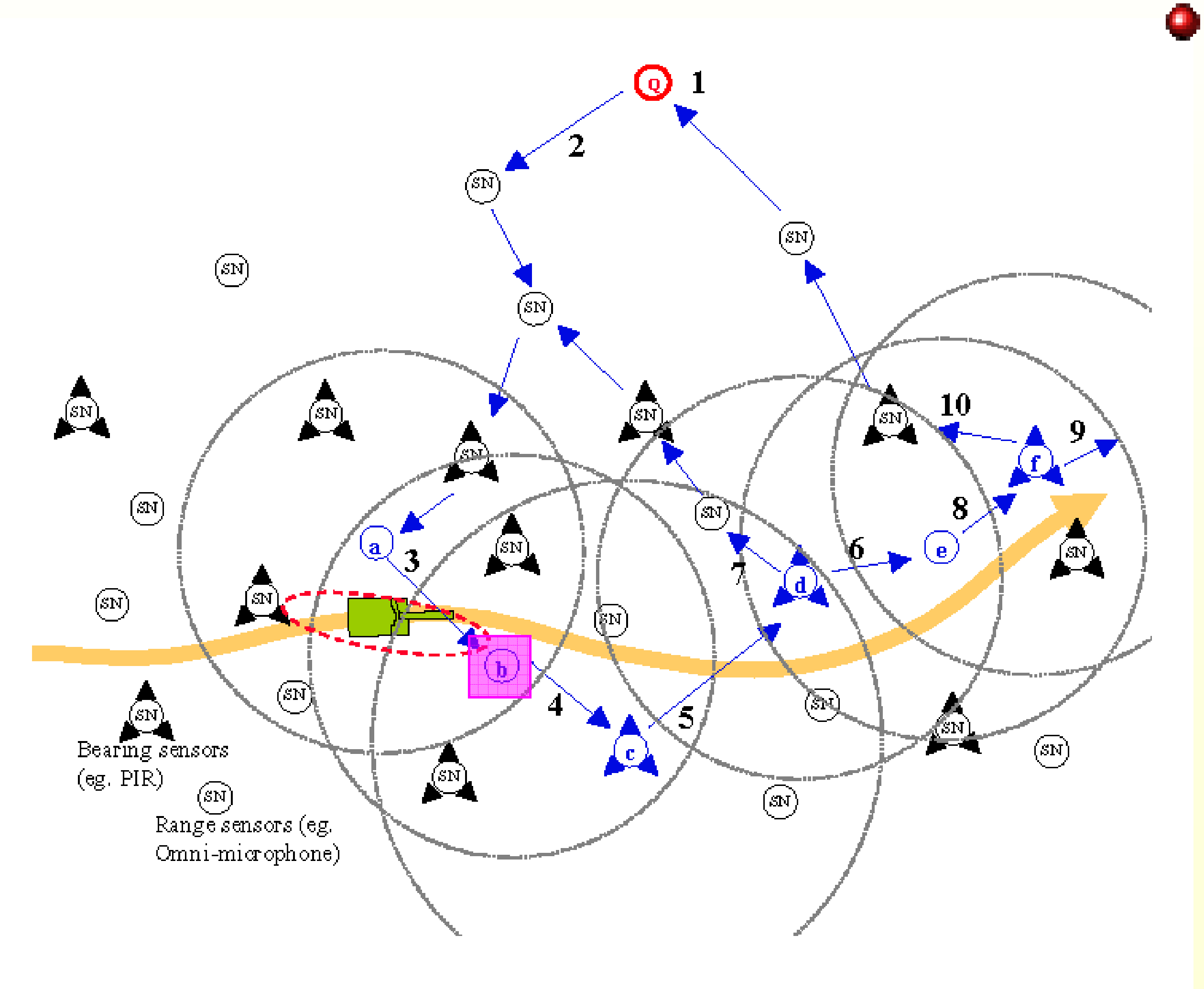


Tracking Scenario



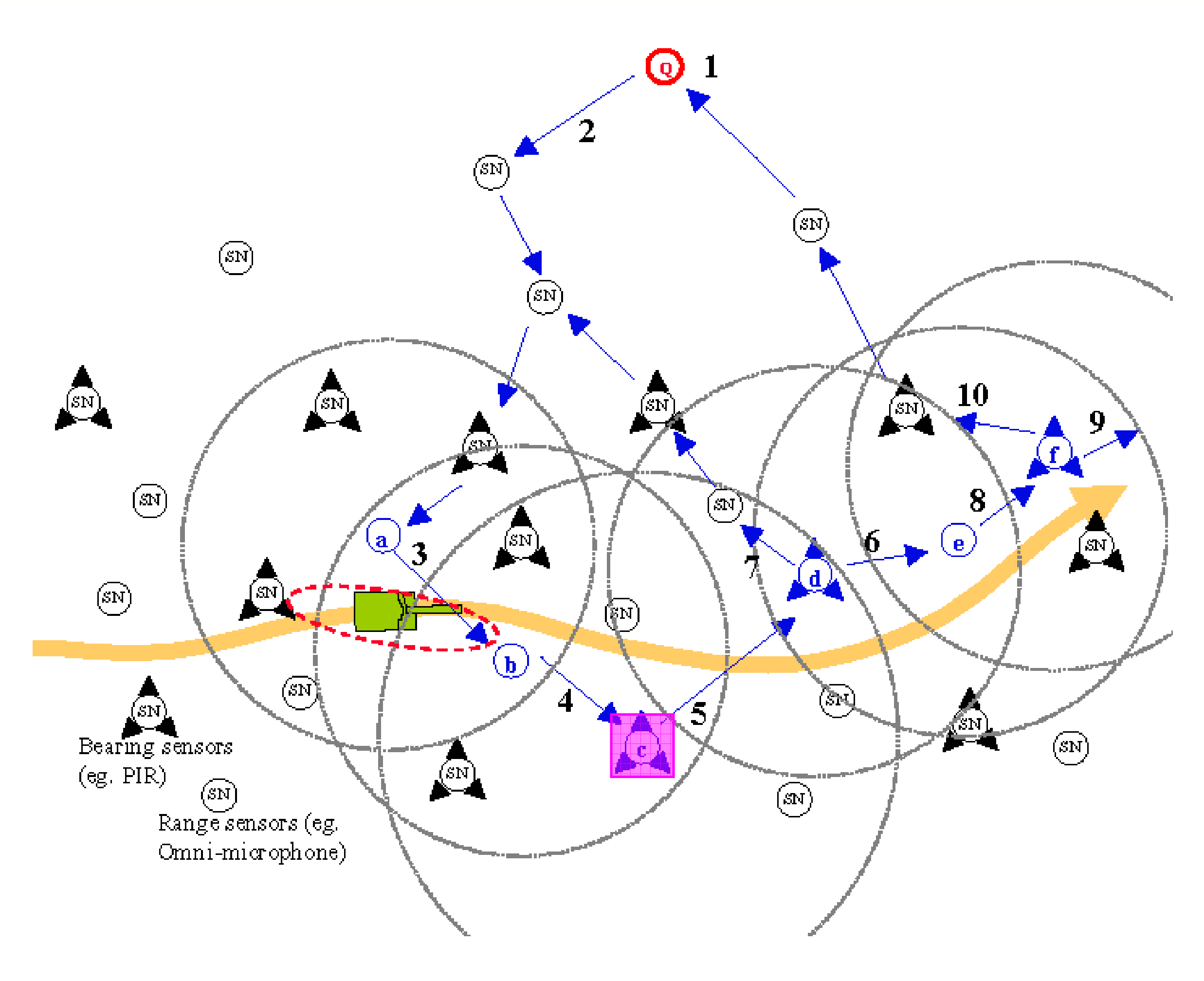
- Sensor *a* senses the location of the target and chooses the next best sensor

Tracking Scenario

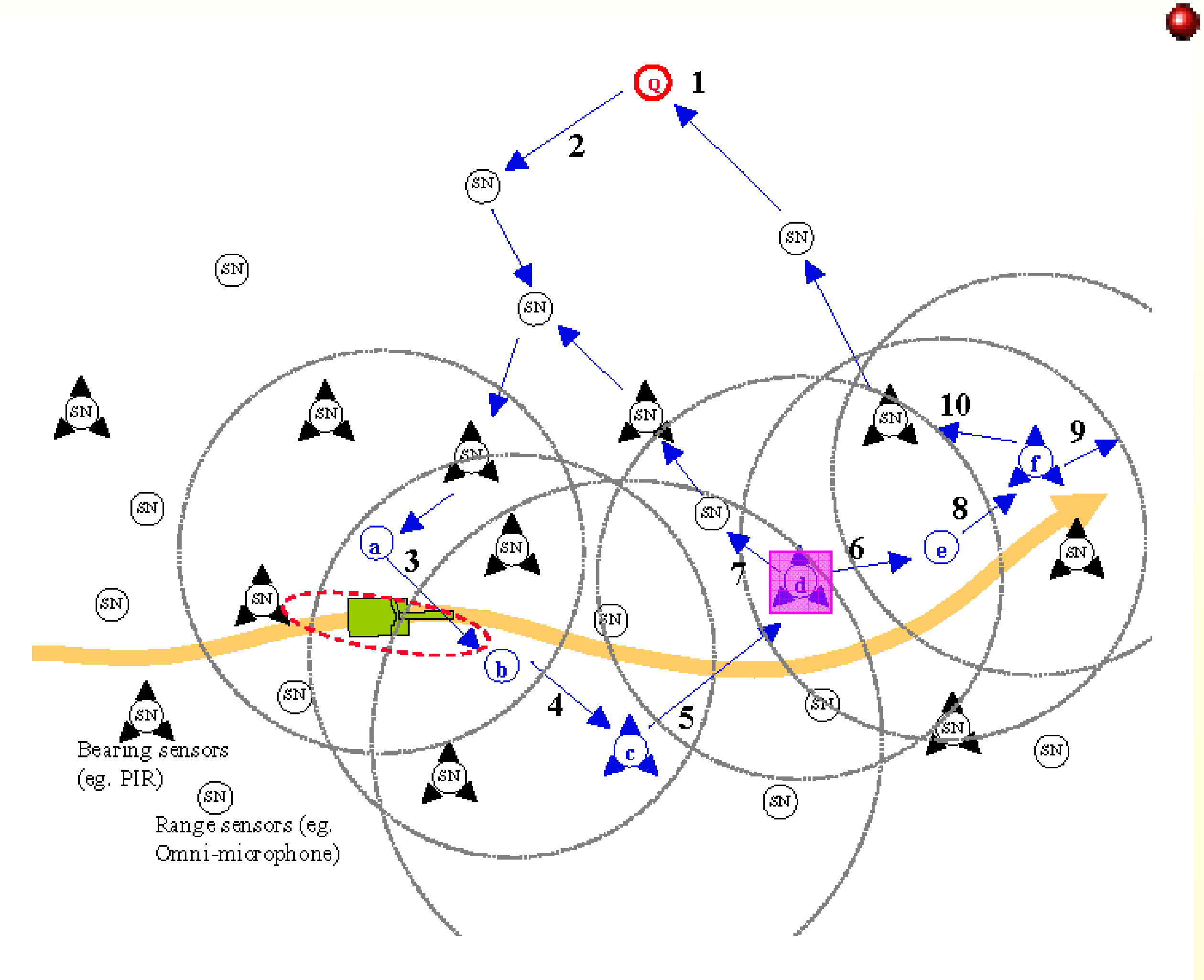


- Sensor *b* does the same

Tracking Scenario

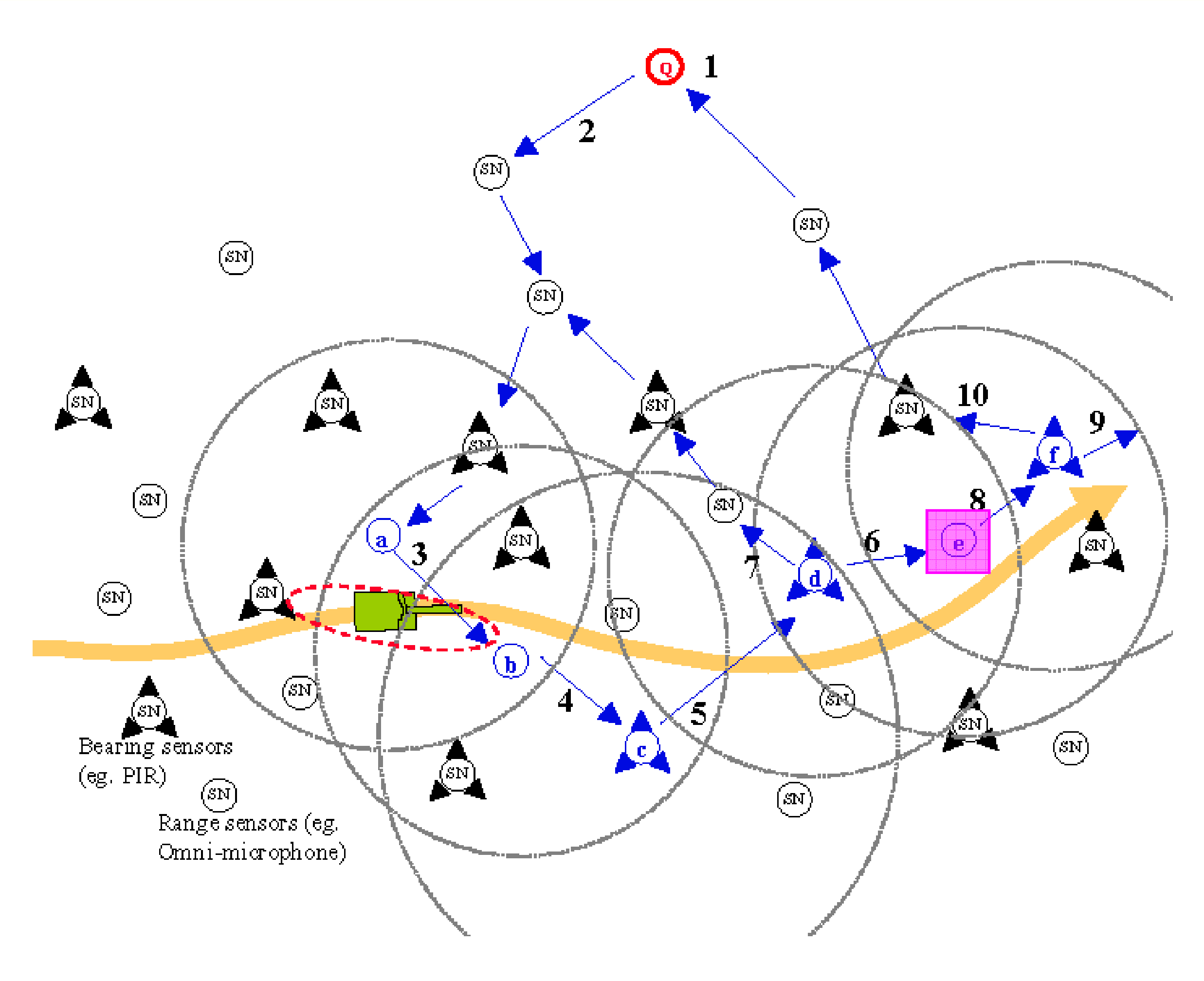


Tracking Scenario

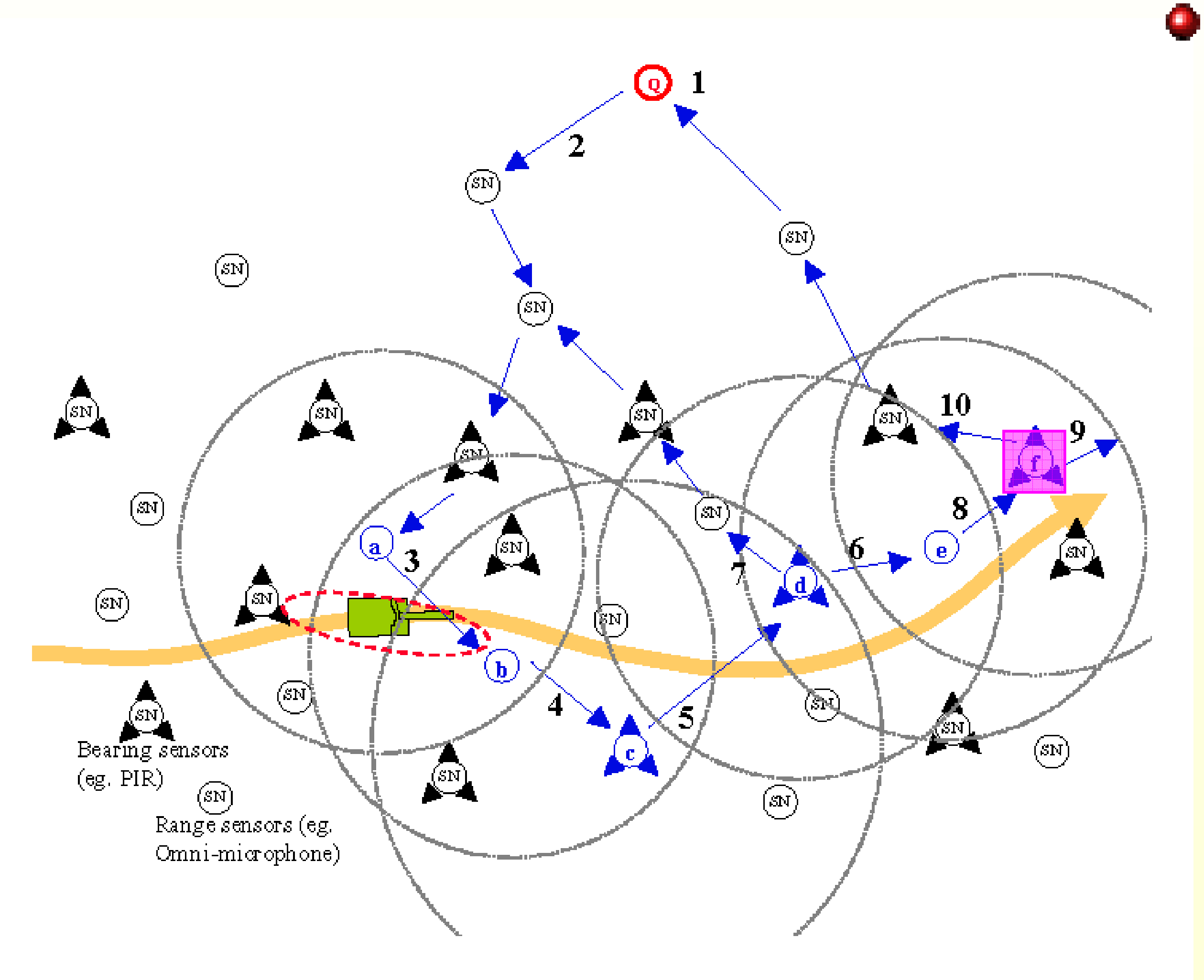


- Sensor *d* both chooses the next best sensor and also sends a reply to the query node

Tracking Scenario



Tracking Scenario



- Sensor *f* loses the target and sends the final response back to the query node

Key Issues

- How is a target detected? How do we suppress multiple simultaneous discoveries?
- How do nodes form collaboration groups to better jointly track the target(s)? How do these groups evolve as the targets move?
- How are different targets differentiated and their identities maintained?
- What information needs to be communicated to allow collaborative information processing within each group, as well as the maintenance of these groups under target motion?
- How are queries routed towards the region of interest?
- How are results from multiple parts of the network accumulated and reported?

Formulation

- Discrete time $t = 0, 1, 2 \ldots$
- K sensors; λ_i^t characteristics of the *i*-th sensor at time t
- N targets; x_i^t state of target i at time t; x^t is the collective state of all the targets; state of a target is its position in the *x*-*y* plane
- Measurement of sensor i at time t is z_i^t; collective measurements from all sensors together are z^t
- $\overline{z_i^t}$ and $\overline{z^t}$ denote the respective measurement histories over time

Sensing Model

- Back to estimation theory

$$z_i^t = h(x^t, \lambda_i^t)$$

measurement function

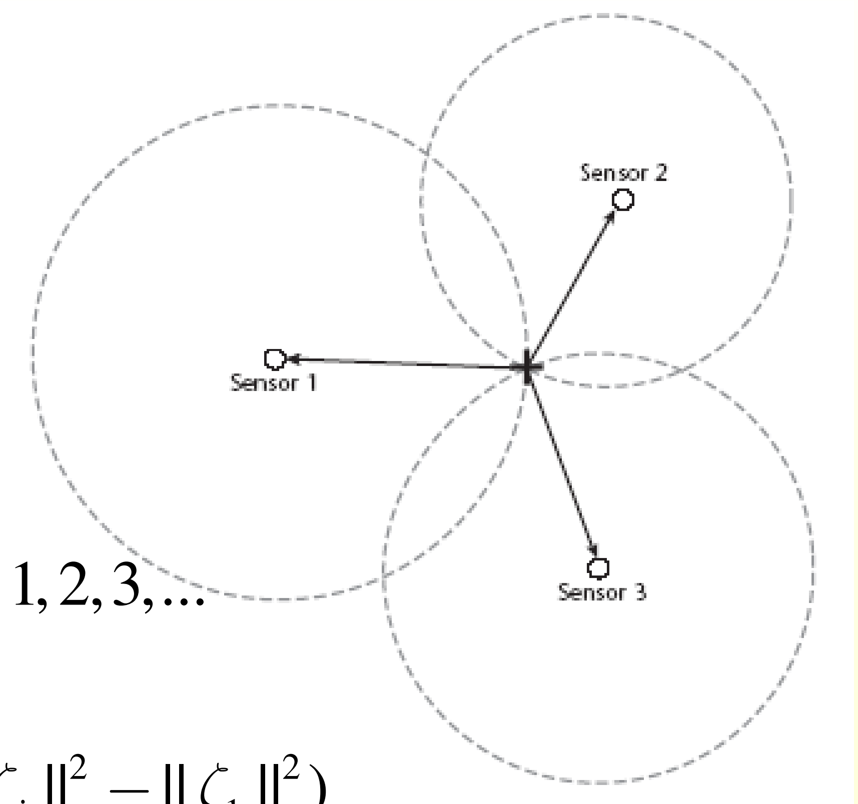
$$z_i^t = H_i^t(\lambda_i^t)\, x^t + w_i^t$$

- Assume time-invariant sensor characteristics
- Use only acoustic amplitude sensors

$$\lambda_i = \left[\zeta_i, \sigma_i^2\right]^T, \quad z_i = \frac{a_i}{\| x_i - \zeta_i \|^{\alpha/2}} + w_i$$

Collaborative Single Target Localization

- Three distance measurements are needed to localize a point in the plane (because of ambiguities)
- Linearization of quadratic distance equations



$$\| x \|^2 + \| \zeta_i \|^2 - 2x^T \zeta_i = \frac{a_i}{z_i}, \quad i = 1, 2, 3, \ldots$$

$$-2(\zeta_i - \zeta_1)^T x = a_i \left(\frac{1}{z_i} - \frac{1}{z_1} \right) - (\| \zeta_i \|^2 - \| \zeta_1 \|^2)$$

$$c_i^T x = d_i$$

subtract equation 1 from equation *i*

Least Squares Estimation

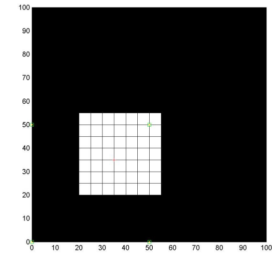
- Since the state *x* has two components, three measurements are needed to obtain two equations
- More measurements lead to an over-determined system -- which can yield more robust estimates via standard least squares techniques

$$Cx = d \qquad (K-1)\times 2,\, 2\times 1 = (K-1)\times 1$$

$$x = \left[\left(C^T C\right)^{-1} C^T\right] d$$

Least-squares solution

Bayesian State Estimation



Initial Distribution $p(x_0)$

Dynamic Model

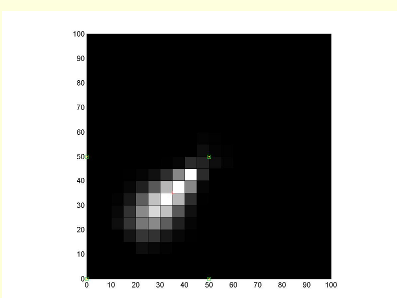
Prior = Posterior at time *k*-1

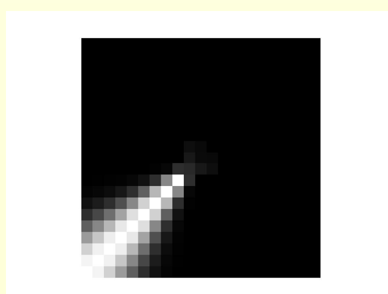
$$p(x_k \mid \bar{z}_k) \propto p(\bar{z}_k \mid x_k) \cdot \underbrace{p(x_k \mid x_{k-1})\, p(x_{k-1} \mid \bar{z}_{k-1})\, dx_{k-1}}$$

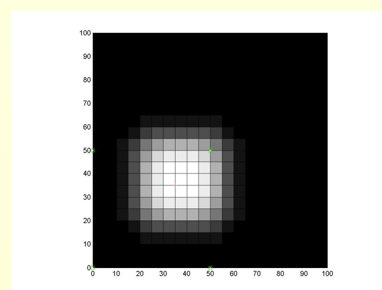
Posterior at time *k*

Observation at time *k*

Prediction at time *k*

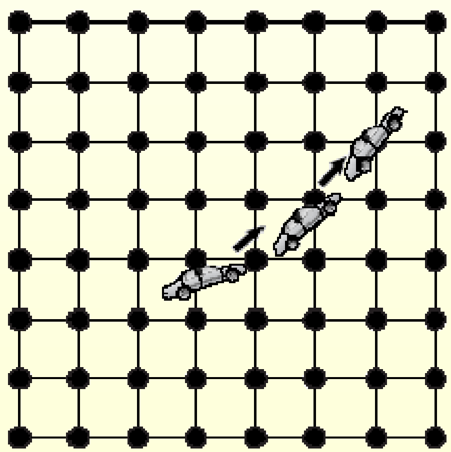




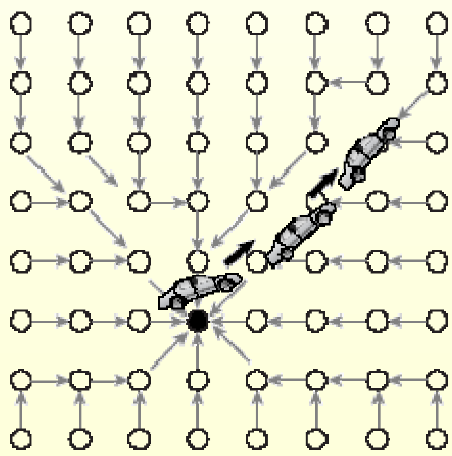


Distributed State Estimation

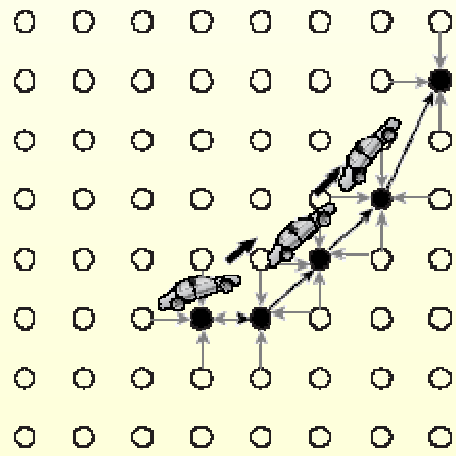
- Observations z are naturally distributed among the sensors that make them
- But which node(s) should hold the state x? Even in the single target case (N=1), this is not clear...



all nodes hold the state



a single fixed node holds the state



a variable node holds the state (the leader)

Many, Many Questions and Trade-Offs

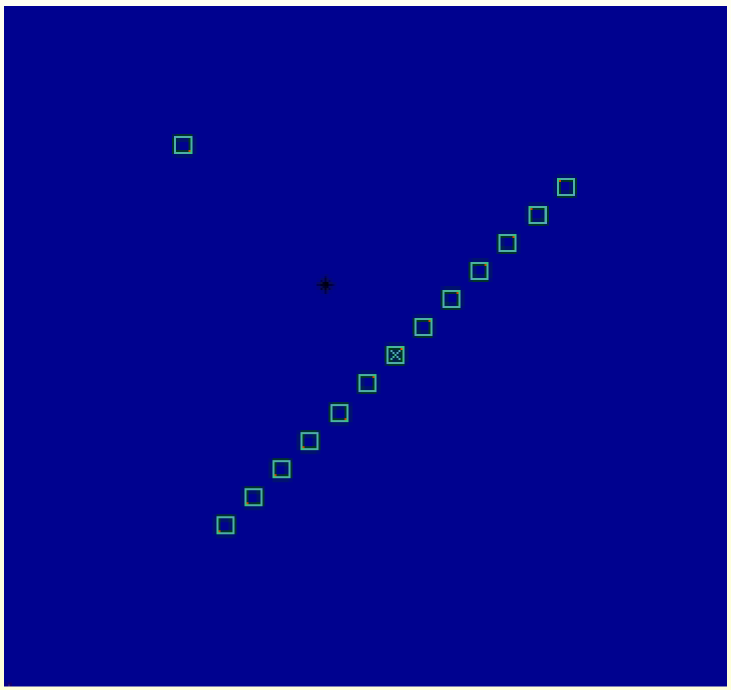
- How are leader nodes to be initially selected, and how are they handed off?
- What if a leader node fails?
- How should the distribution of the target state (= position) be represented? parametrically (Gaussian) or non-parametrically (particles)?

Best-possible state estimation, under constraints → Communication, Delay, Power

IDSQ:
Information-Driven Sensor Querying

IDSQ: Information-Driven Sensor Querying

Localize a target using multiple acoustic amplitude sensors



Challenge

- Select next sensor to query to *maximize* information return while *minimizing* latency & bandwidth consumption

Ideas

- Use information utility measures
  - E.g. Mahalanobis distance, volume of error covariance ellipsoid
- Incrementally query and combine sensor data

IDSQ Sensor Selection

- **Idea**: maximize the *predicted* information that a sensor's measurement will bring, given the current estimated distribution

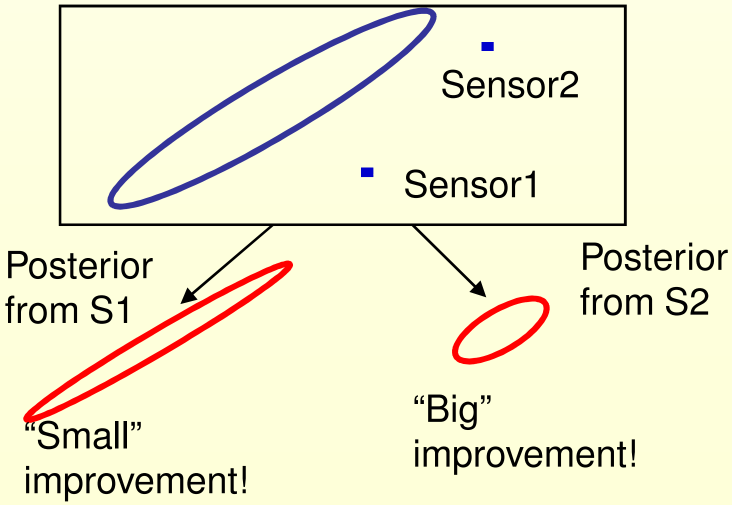
- Information is measured using mutual information

$$I(U;V) = E_{p(u,v)}\left[\log \frac{p(u,v)}{p(u)p(v)}\right]$$

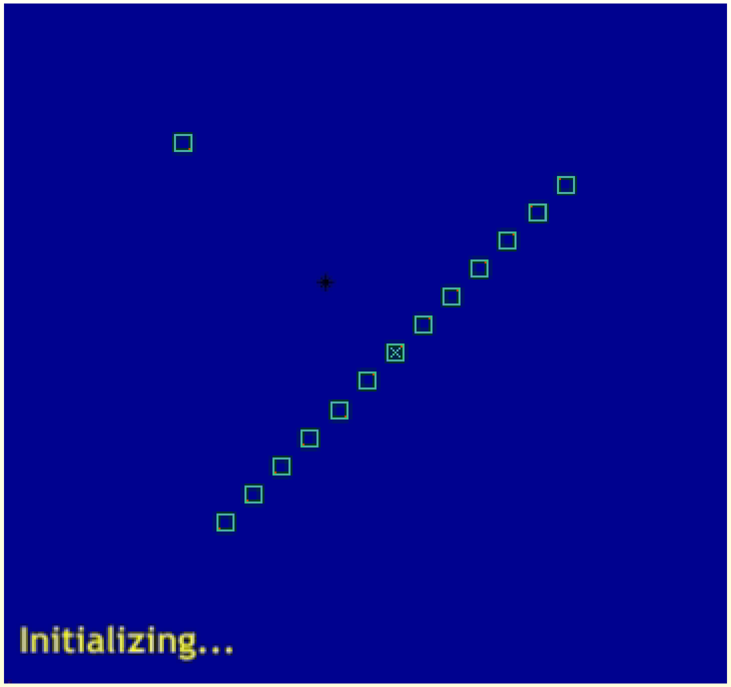
- IDSQ criteria: $k_{IDSQ} = \arg\max_{k \in N} I(X^{(t+1)}; Z^{(t+1)} \mid \overline{Z^{(t)}} = \overline{z^{(t)}})$

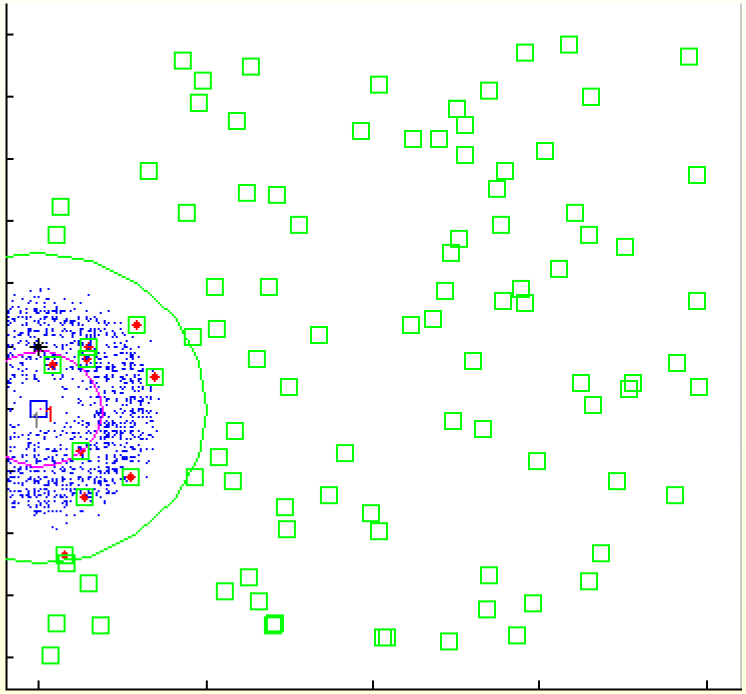
  where N is the set of candidate sensors (i.e. topological neighbors)

- This is equivalent to choosing the sensor which will give the greatest change to the current belief.



With Information-Driven Sensor Selection





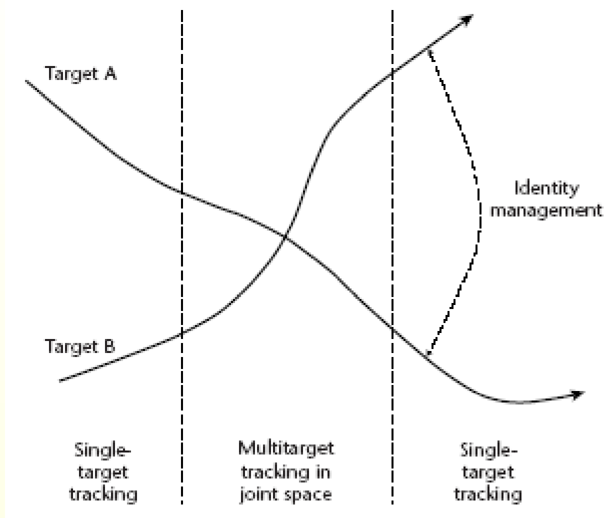
Leader election and hand-off

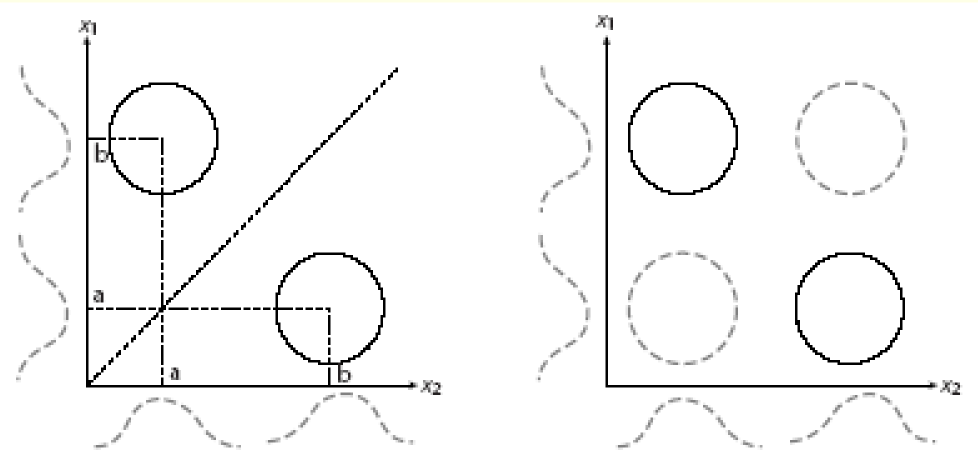
Tracking Multiple Objects

- New issues arise when tracking multiple interacting targets
  - The dimensionality of the state space increases — this can cause an exponential increase in complexity (e.g., in a particle representation)
- The distribution of state representation becomes more challenging
  - One leader per target?
  - What if targets come near and they mix (data association problem)?

State Space Decomposition

- For well-separated targets, we can factorize the joint state space of the *N* targets into its marginals
- Such a factorization is not possible when targets pass near each other
- Another factorization is between target locations and identities
  - the former require frequent local communication
  - the latter less frequent global communication





(a)

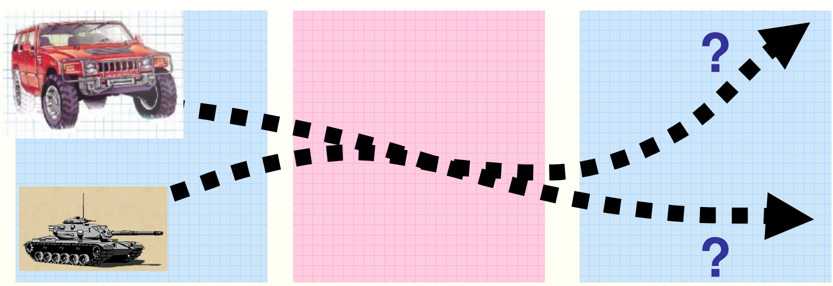
(b)

Data Association

- Data association methods attribute specific measurements to specific targets, before applying estimation techniques
  - Even when there is no signal mixing, the space of possible associations is exponential: $N!/K!$ possible associations (N = # of targets, K = # of sensors)
  - Signal mixing makes this even worse: 2^{NK} possible associations
- Traditional data association methods are designed for centralized settings
  - Multiple Hypothesis Tracking (MHT)
  - Joint Probabilistic Data Association (JPDA)
- Network delays may cause measurements to arrive out of order in the nodes where the corresponding state is being held, complicating sequential estimation

Identity Management

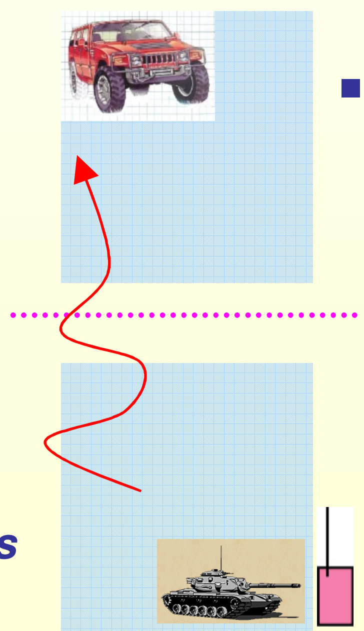
Distributed Identity Management



Mixing

- Multi-target Tracking (MTT)
  - Basic application of sensor networks
  - Data association problem → *Target mixing*

Action at a distance?



[Shin, Zhao, G., IPSN'03]

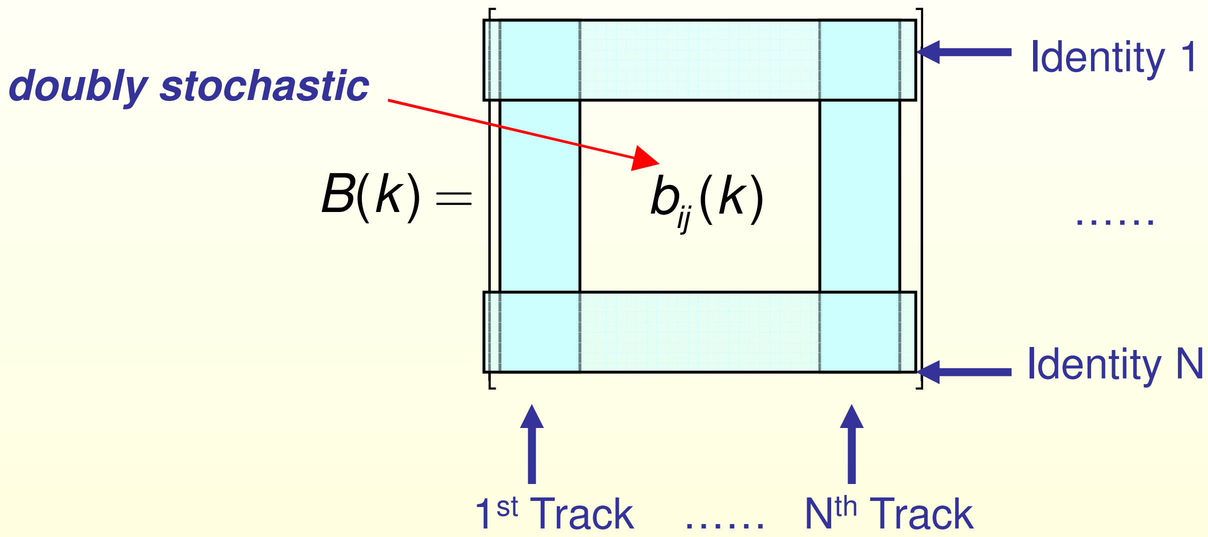
Sensor confirms tank

- Multi-target Identity Management
  - Represent and manage additional quantity called target identity
  - Simplified version
    - Position estimates are given
    - Fixed number of targets assumed

Approach: Belief Matrix

The belief matrix $B(k)$ is the main quantity that the algorithm maintains

$\rightarrow$ Which ID goes with which track, with what probability?



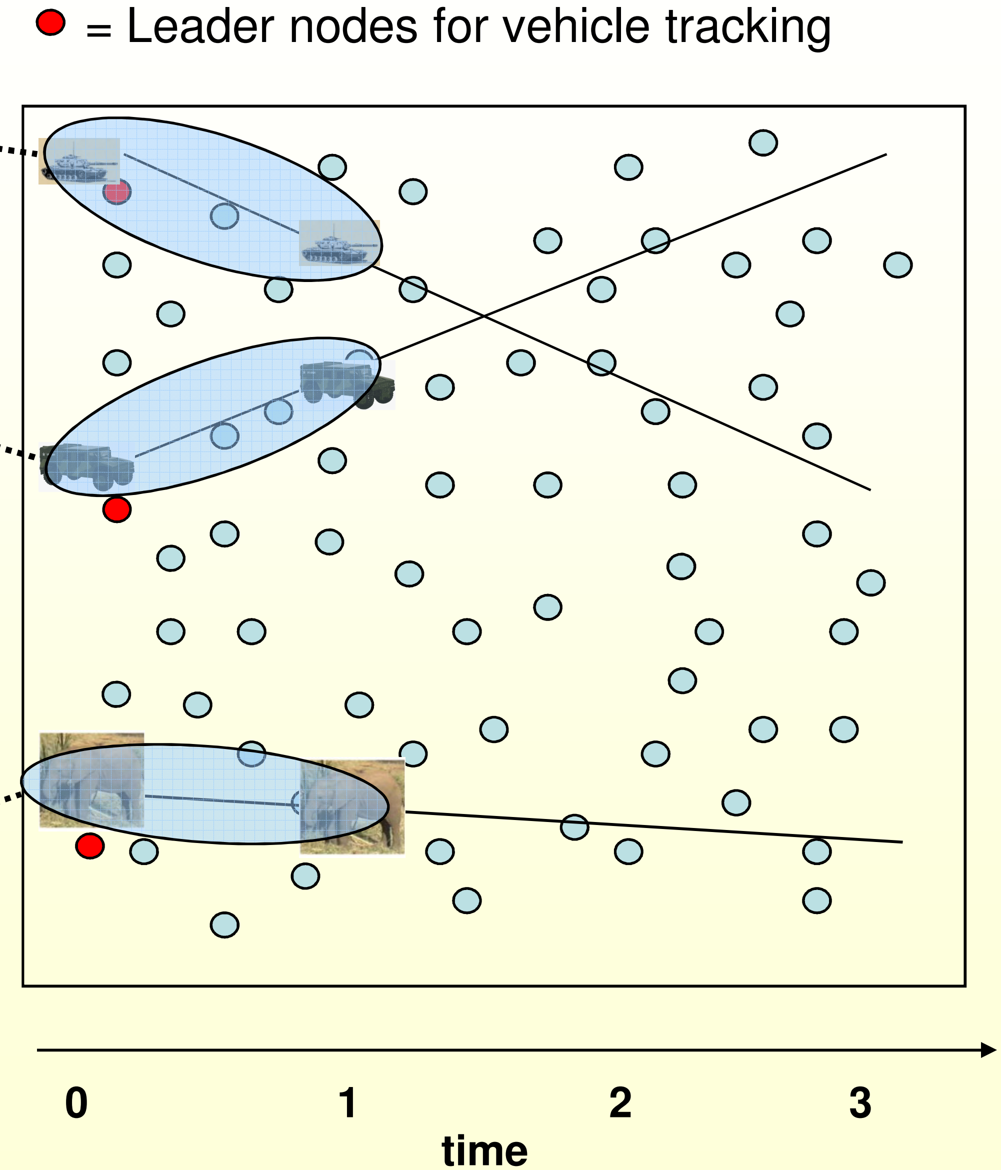
where $b_{ij}(k) = p(\mathrm{ID}(x_j(\mathrm{k})) = i)$ and $\vec{b_i}(k)$ is ID belief vector on i_{th} track.

Q: How to update $B(k)$, given $X(k+1)$?

Distributed Management of B(k), I

$$B(0) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$b_1(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad b_2(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad b_3(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

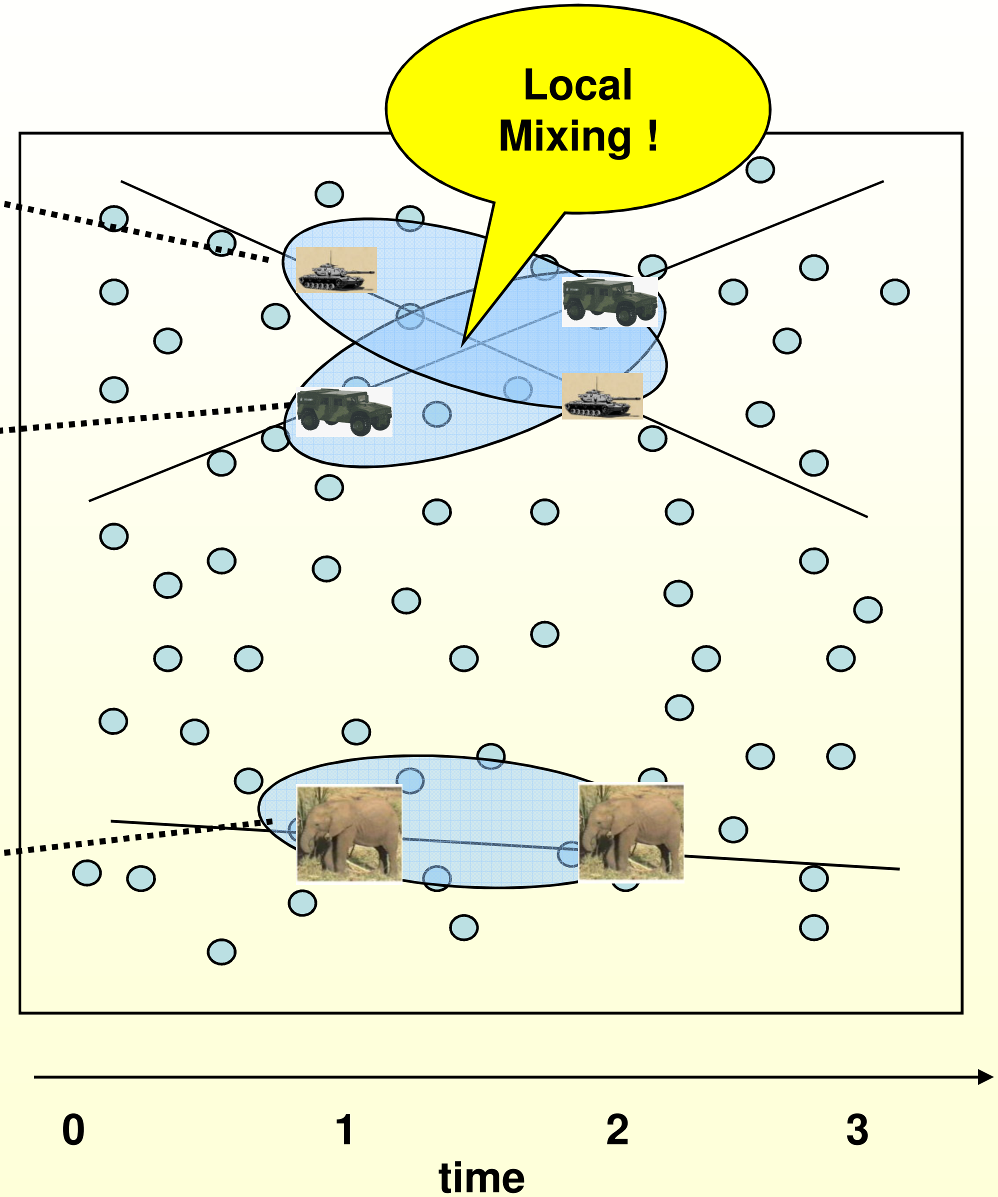


Distributed Management of B(k), II

$$b_1(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$b_2(1) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$b_3(1) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

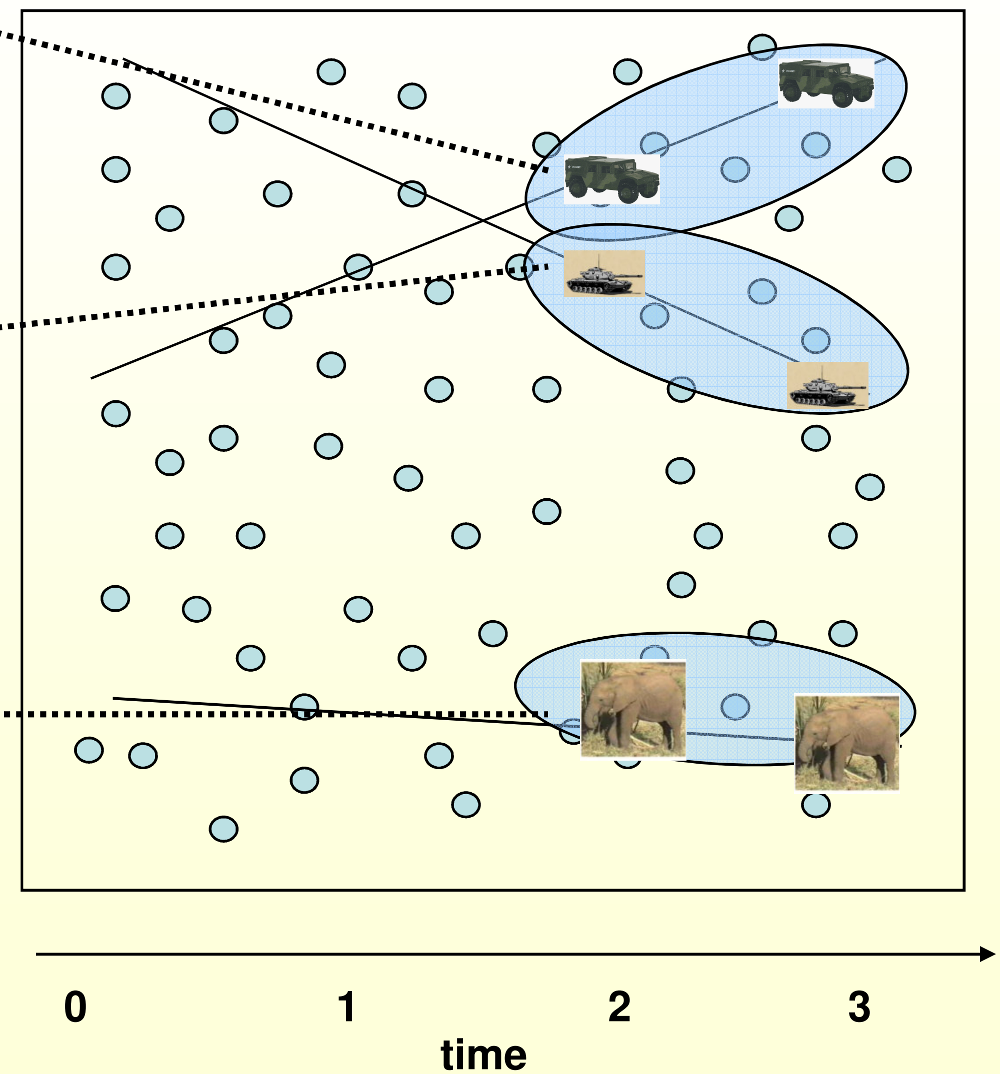


Distributed Management of B(k), III

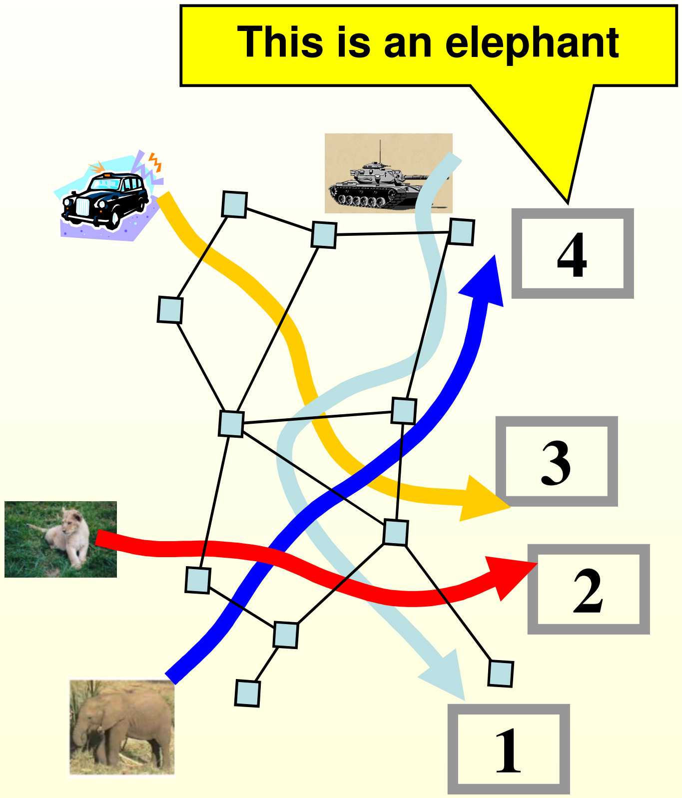
$$b_1(2) = \alpha b_1(1) + (1-\alpha) b_2(1) = \begin{bmatrix} \alpha \\ 1-\alpha \\ 0 \end{bmatrix}$$

$$b_2(2) = (1-\alpha) b_1(1) + \alpha b_2(1) = \begin{bmatrix} 1-\alpha \\ \alpha \\ 0 \end{bmatrix}$$

$$b_3(2) = b_3(1) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Incorporating Local Evidence



$$B = \begin{array}{c} \begin{array}{cccc} \boxed{1} & \boxed{2} & \boxed{3} & \boxed{4} \end{array} \\ \begin{bmatrix} 0.1737 & 0.0947 & 0.3447 & 0.3869 \\ 0.3527 & 0.6473 & 0 & 0 \\ 0.3201 & 0.1744 & 0.3043 & 0.2011 \\ 0.1535 & 0.0836 & 0.3509 & \boxed{0.4120} \end{bmatrix} \end{array}$$

$$\downarrow$$

$$B = \begin{bmatrix} 0.1737 & 0.0947 & 0.3447 & 0 \\ 0.3527 & 0.6473 & 0 & 0 \\ 0.3201 & 0.1744 & 0.3043 & 0 \\ 0.1535 & 0.0836 & 0.3509 & 1 \end{bmatrix}$$

Not doubly-stochastic → inconsistent!

Renormalization, Given Local Evidence

1. Ideal solution: Bayesian normalization,

given the priors, all the mixing events in history

and all the sensor evidence.

Exact solution in this framework. (Bayesian posterior)

Used as a reference - Desirable properties of the solution.

2. Realistic solution: Sinkhorn Iteration (repeatedly normalize rows and columns)

[Sinkhorn 1964,1967; Sinkhorn and Knopp 1967]

What Do We Want?

- The belief matrix represents a probability distribution. The matrix *A* represents our *a priori* belief, but violates sum constraints.
- We would like to find the sum-constrained (feasible) matrix *B* that is the *closest distribution* to *A* (which is infeasible).
- Use Kullback-Leibler distance (measure of distance between distributions):

$$KL(B:A) = \sum_{j=1}^{n} \sum_{i=1}^{m} b_{ij} \log \frac{b_{ij}}{a_{ij}}$$

Sinkhorn Scaling and the Kullback-Leibler Distance

$$KL(B:A) = \sum_{j=1}^{n} \sum_{i=1}^{m} b_{ij} \log \frac{b_{ij}}{a_{ij}}$$

Theorem: Given a prior matrix $A \in \mathbb{R}^{m \times n}$, the matrix *B* that satisfies the row and column sum constraints, and minimizes the KL-distance from the prior matrix *A* is **always** the solution of the Sinkhorn scaling process.

[Balakrishnan, Hwang,Tomlin '04]

Solve by interior point methods:

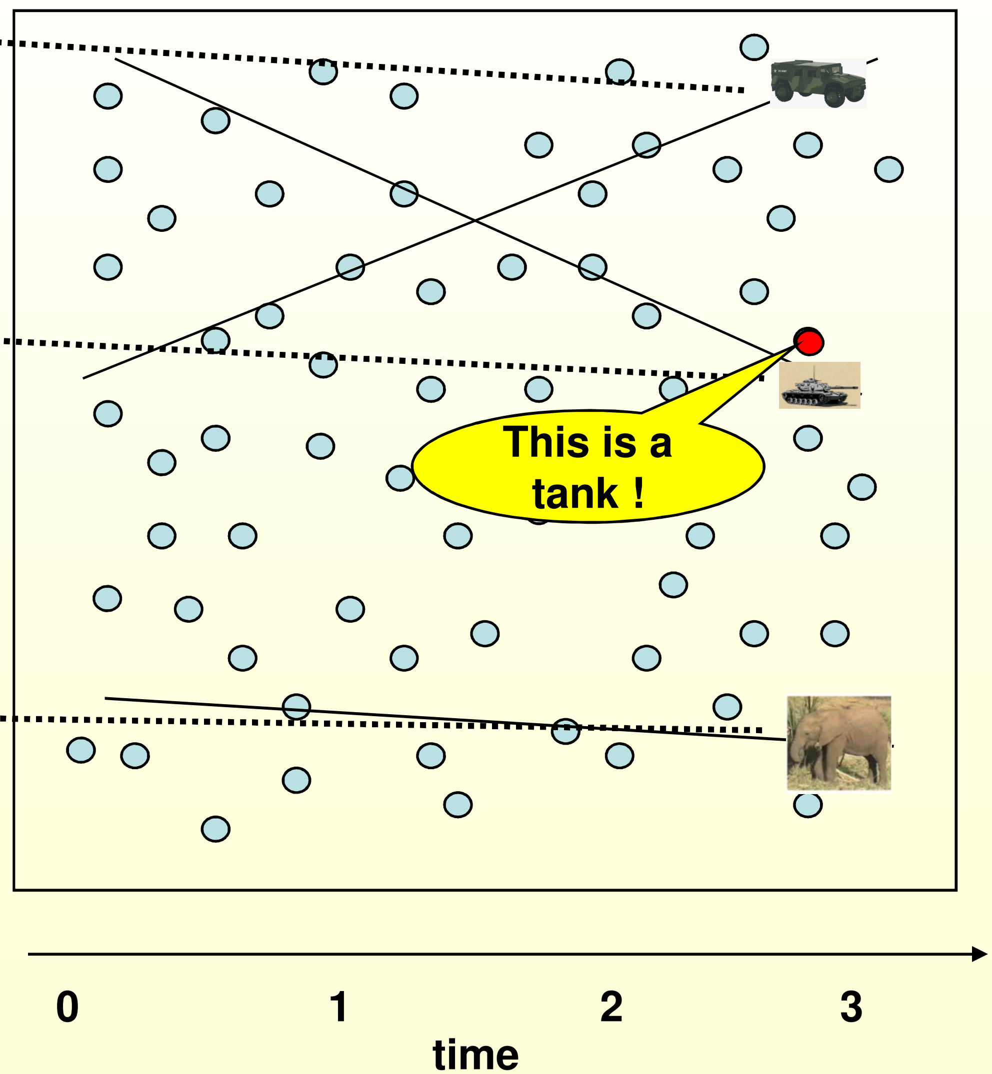
$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} \log \frac{b_{ij}}{a_{ij}} \\ \text{subject to} & \sum_{j=1}^{n} b_{ij} = r_i \; \forall i = 1 \cdots m \\ & \sum_{i=1}^{m} b_{ij} = c_j \; \forall j = 1 \cdots n \\ & b_{ij} \geq 0 \; \forall i = 1 \cdots m; \; j = 1 \cdots n \end{array}$$

Distributed Management of B(k), IV

$$b_1(3) = \begin{bmatrix} \alpha \\ 1-\alpha \\ 0 \end{bmatrix}$$

$$b_2(3) = \begin{bmatrix} 1-\alpha \\ \alpha \\ 0 \end{bmatrix}$$

$$b_3(3) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

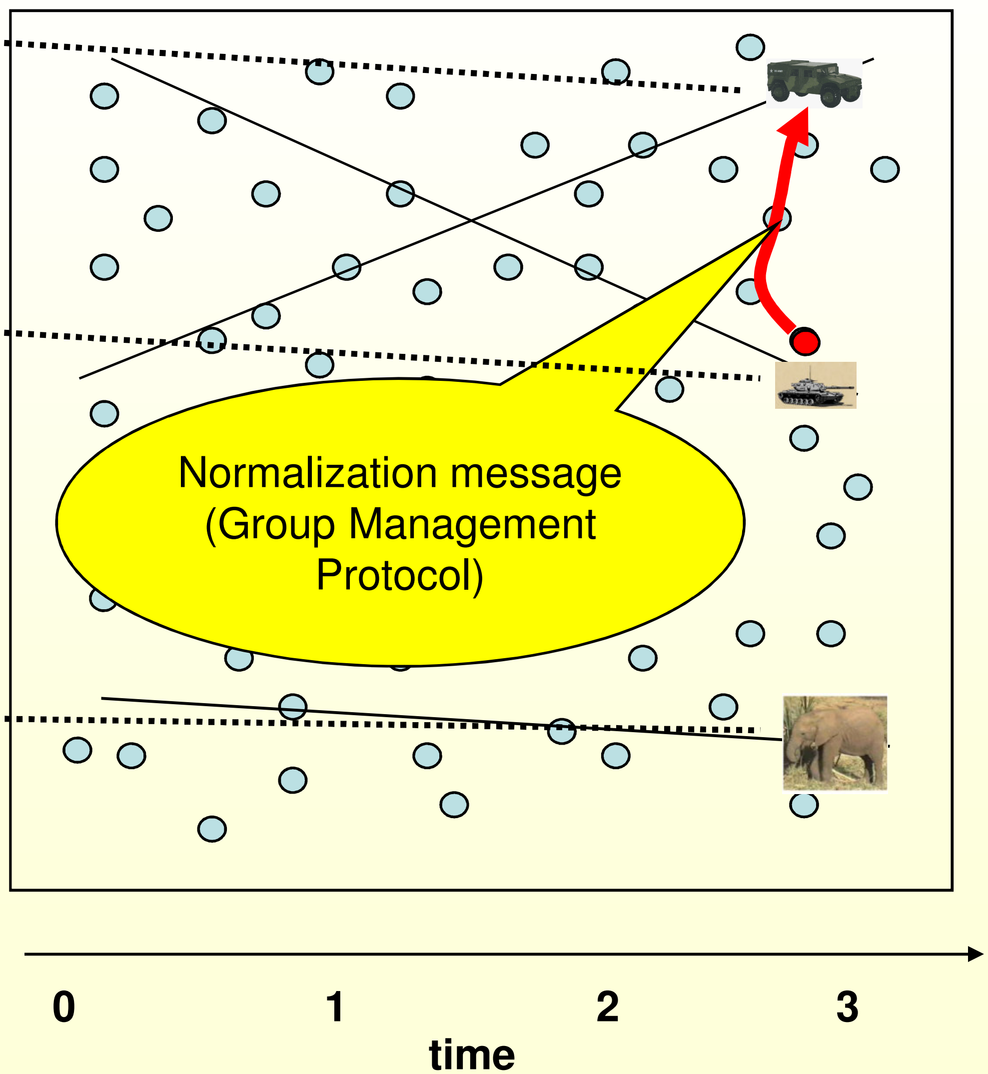


Distributed Management of B(k), V

$$b_1(3) = \begin{bmatrix} \alpha \\ 1-\alpha \\ 0 \end{bmatrix}$$

$$b_2(3) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$b_3(3) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

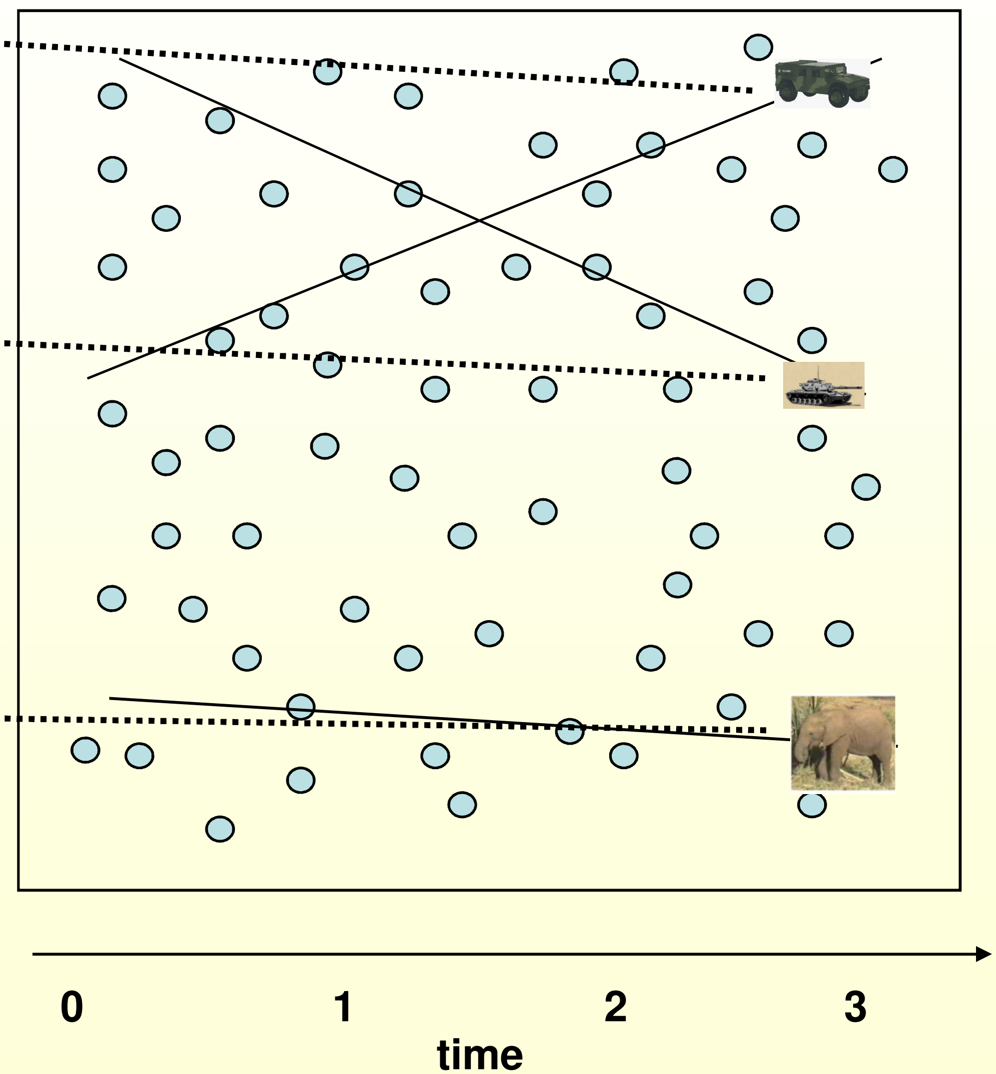


Distributed Management of B(k), VI

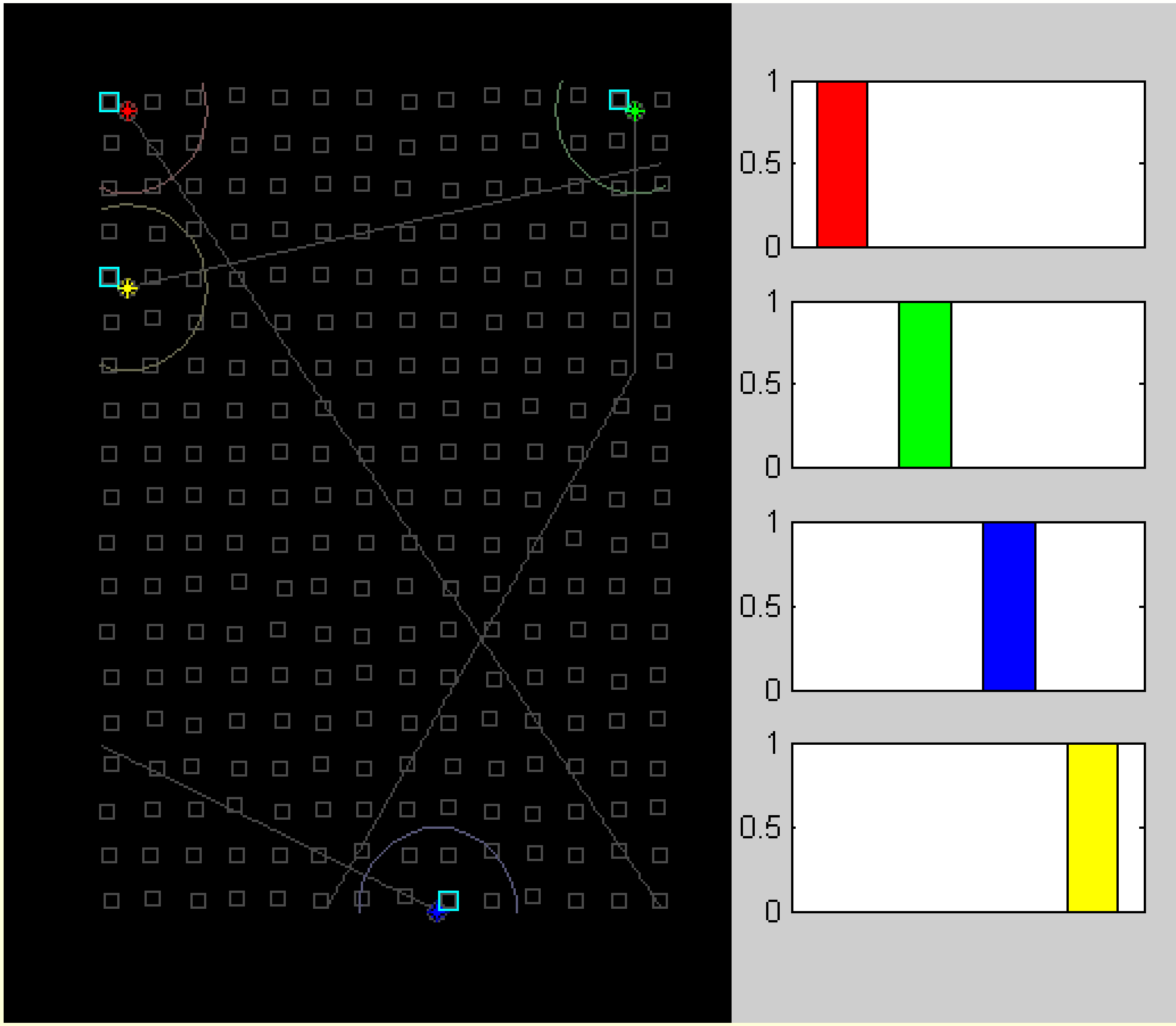
$$b_1(3) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$b_2(3) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$b_3(3) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Target Mixing Video



Particles filters are used to keep track about multiple hypotheses about the location of each vehicle

Distributed Implementation of Tracking and Normalization

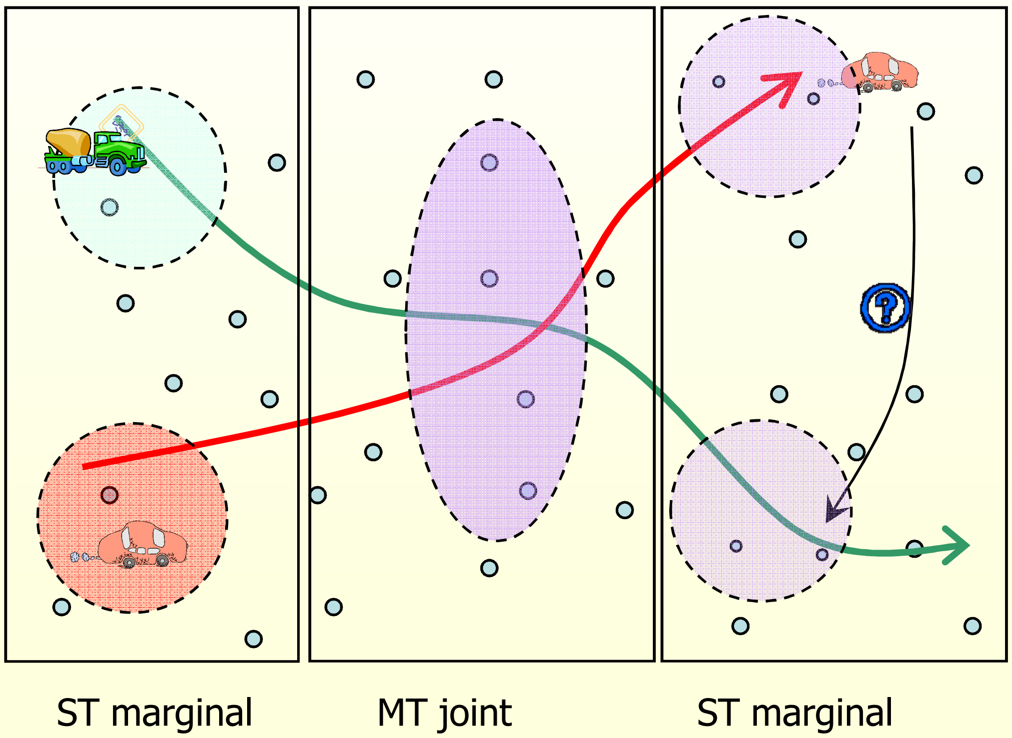
Q: Which nodes store and/or compute what information?

-- When do we track in joint space?

-- How to distribute B(k)?

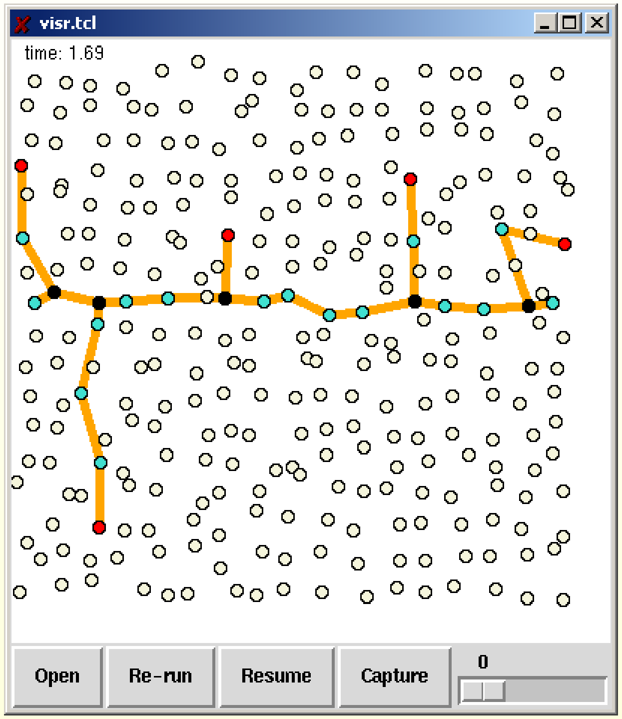
-- How to implement the probabilistic normalization?

-- What is required at the communication/network layer to make the above happen?



Managing Sensor Groups

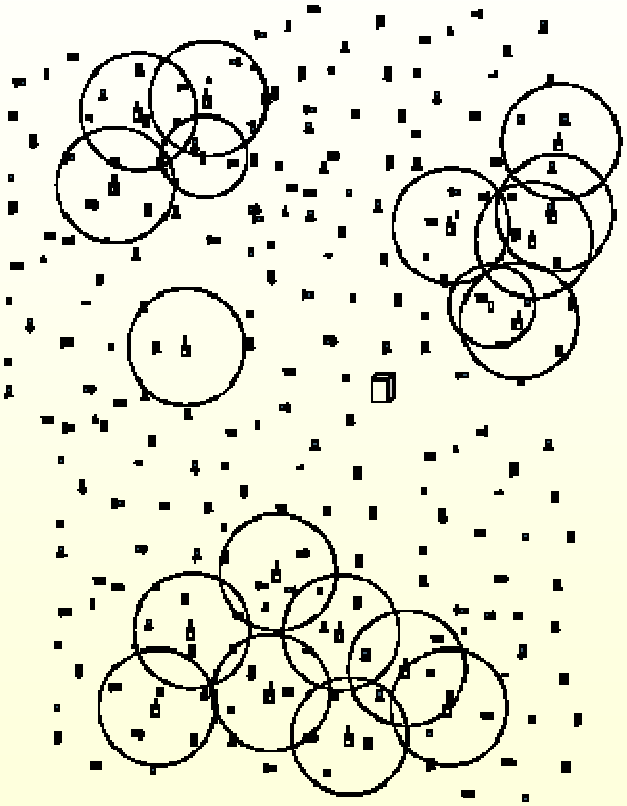
- We have distributed columns of B(k) to leader nodes tracking targets.
- When a leader initiates a normalization based on local evidence, it has to know where are the other leaders that have non-zero mass on the evidence ID.
- Group Management Protocol: Maintains the group membership based on the ID probability mass.
- Communication needs to be minimized.



RoamHBA protocol
[Fang, Liu, G., Zhao, IPSN'04]

Sensor Collaboration Groups

Target Counting



[Fang, Zhao, G., MobiHoc'03]

Problem Spec:

- A sensor network with multiple targets present. Targets can be stationary or moving;
- Each sensor can detect the local superimposed amplitude of target signals (e.g., acoustic) at any instant of time;

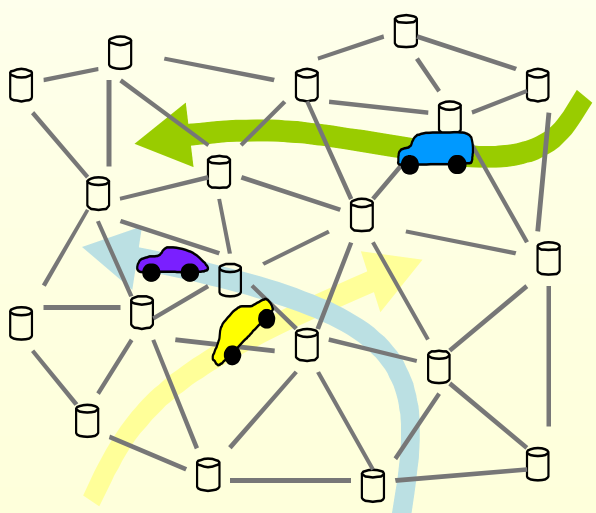
Objective:

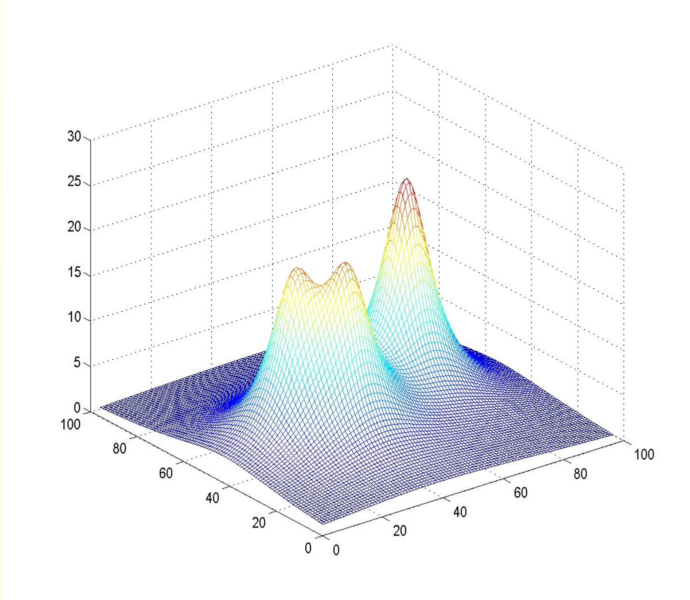
To determine the number of targets and their approximate locations in the field, forming an initial count and re-computing the count when targets move, enter, or leave the field.

Acoustic Signal Field Landscape

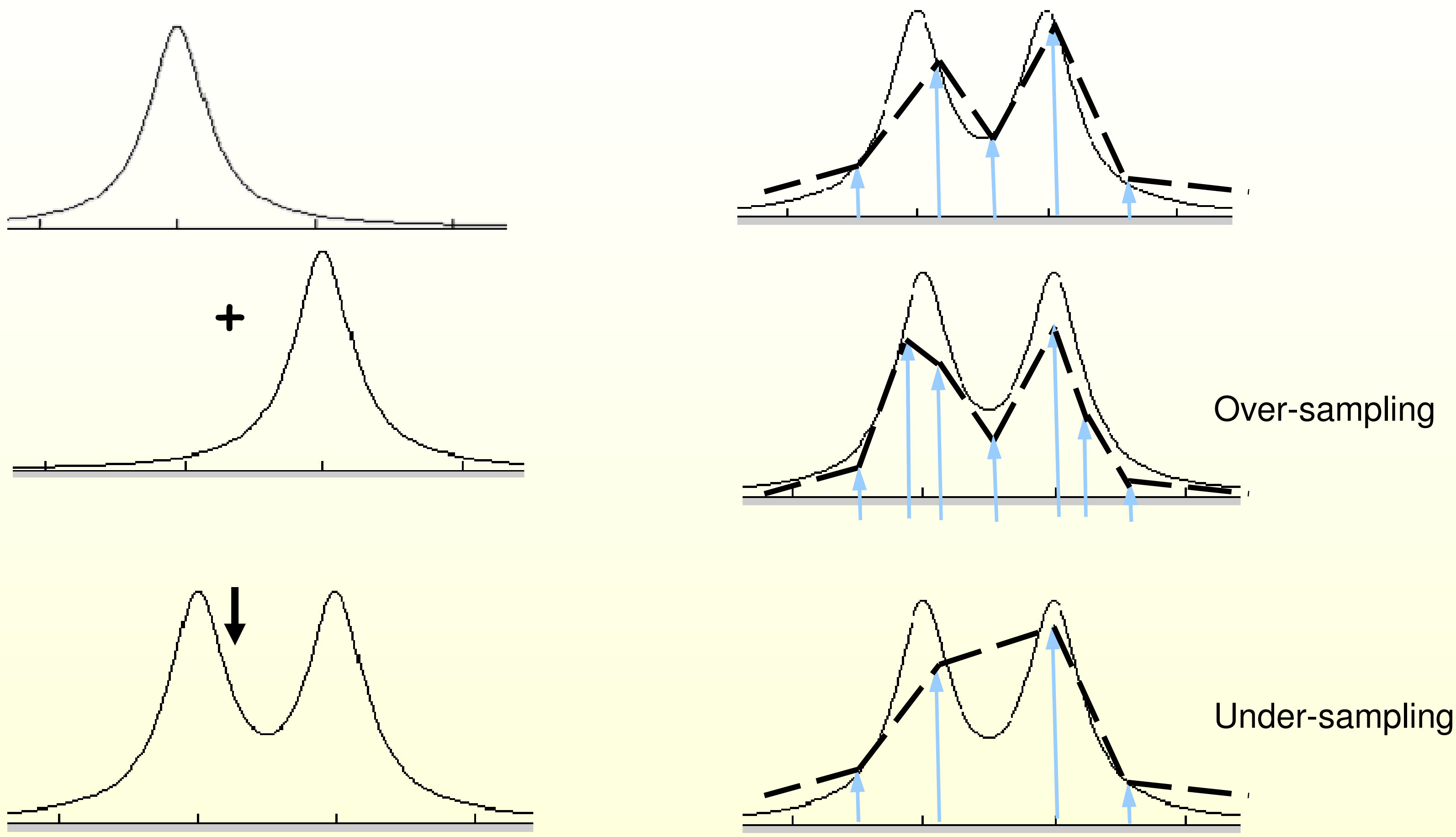
Goal: count and track the significant signal peaks in the field

1. Signal attenuation rate, and the spacing and communication range of sensors have big impact on "signal resolution"
2. Number of detected peaks may not equal number of targets due to sampling artifacts and/or noise.





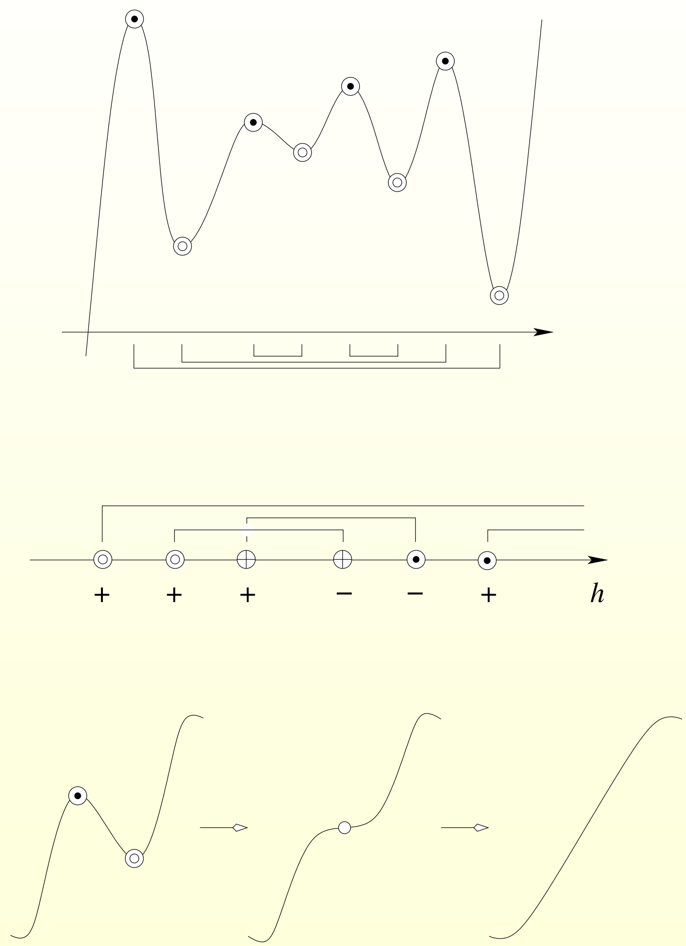
Scattered Amplitude Sampling



In 1-D, the number of peaks in sampled landscape cannot exceed the number of peaks in the true landscape. However, this is possible in 2-D.

Peak Landscape Simplification

- Some signal peaks may be noise
- Usually such peaks are near other critical points of the landscape
- Topological ideas, such as persistent homology can be used to simplify the landscape by canceling saddles with maxima, etc., removing noise

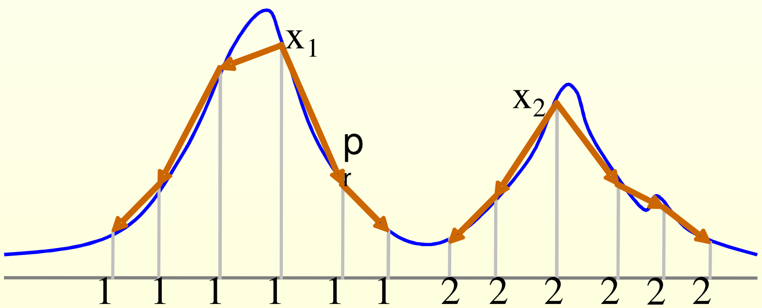


Downhill Flooding Protocol (DFP)

A sensor node is qualified as a **leader,** if its reading is higher than that of all its one hop neighbors.

Local *leader election* is conducted by sensors exchanging information with their neighbors via one hop broadcast.

p_r = received signal power at each node



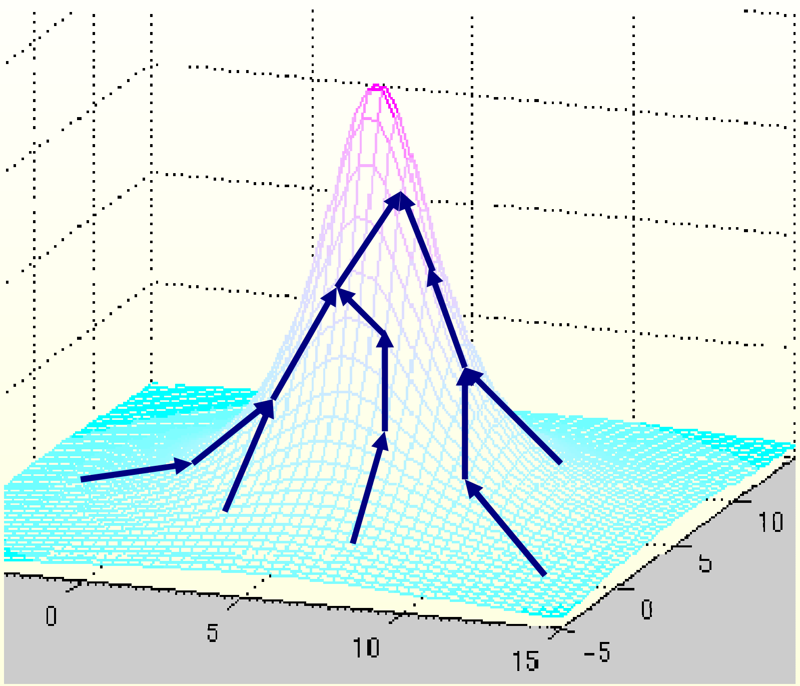
x_1 and x_2 are elected leaders,

other nodes join one of the groups formed by the leaders: each node joins the highest leader it can reach by a monotone ascending path

1. Only sensors with myPr > threshhold will participate
2. Each sensor emits a packet in each protocol period, broadcasting its reading Pr and its ID
3. Sensors pass on or drop a packet P from their neighbors, according to the following rule:

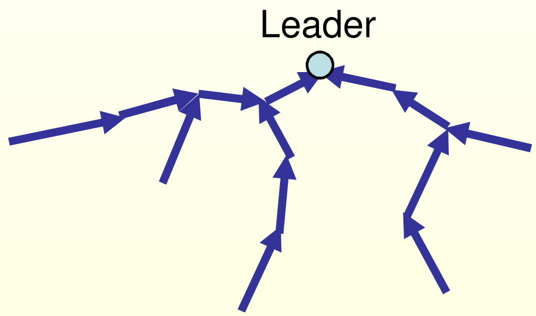
```
If (Pr recorded in the packet > myPr &
        Pr of the sensor relaying this
    packet > myPr)
    broadcast(P);
else
    drop(P);
```

Sensor Cluster Trees



For each node,

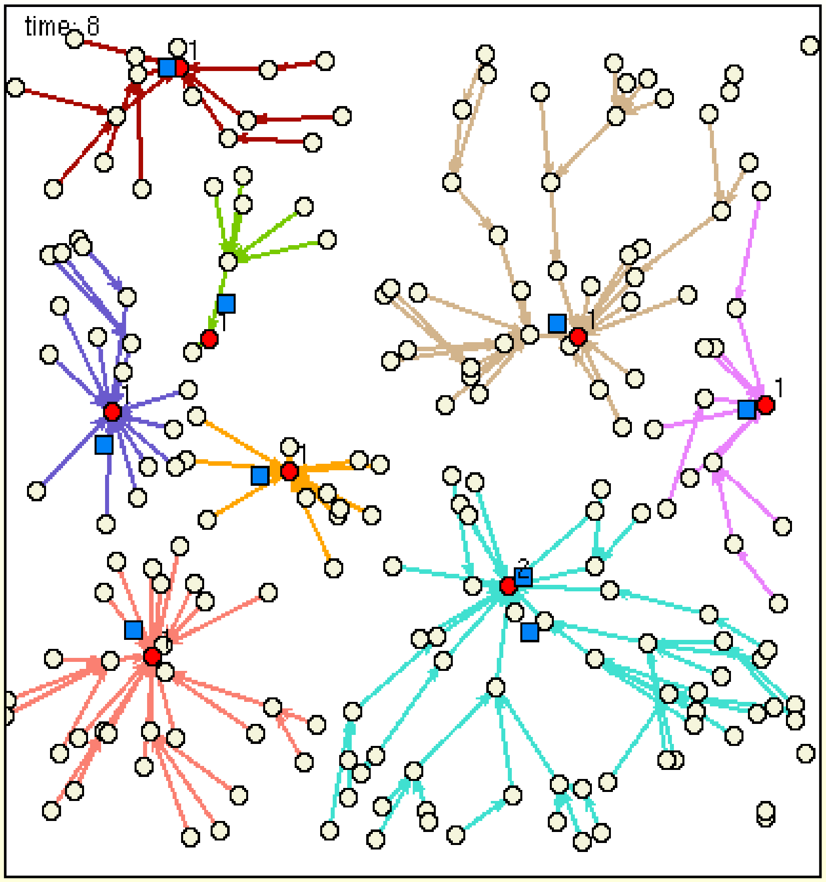
parent = the neighboring node with maximum Pr (received signal power)



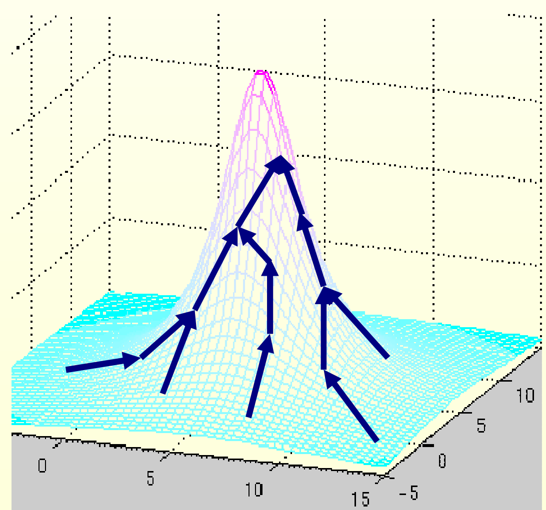
Bird's eye view of the cluster tree structure. Each node follows a strictly upwards path to the highest peak it can reach.

We call such groups of sensors aggregates, as they collaboratively perform a task.

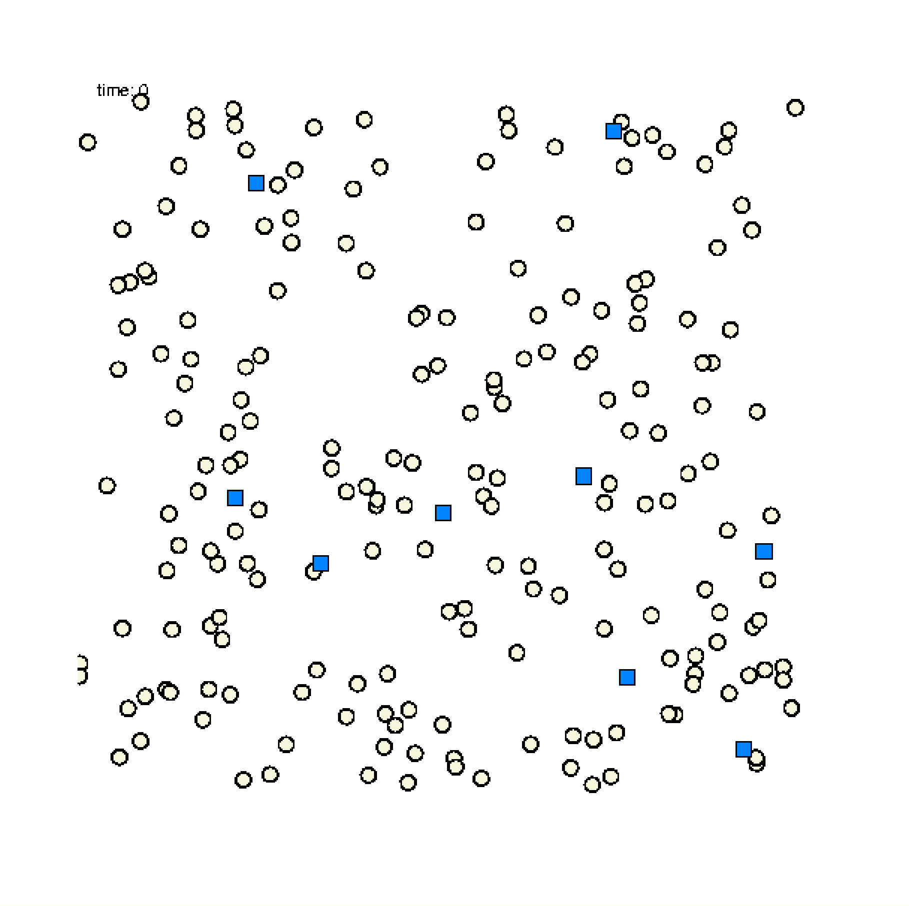
2-D View of a Sensor Field with Cluster Trees Formed Using DFP

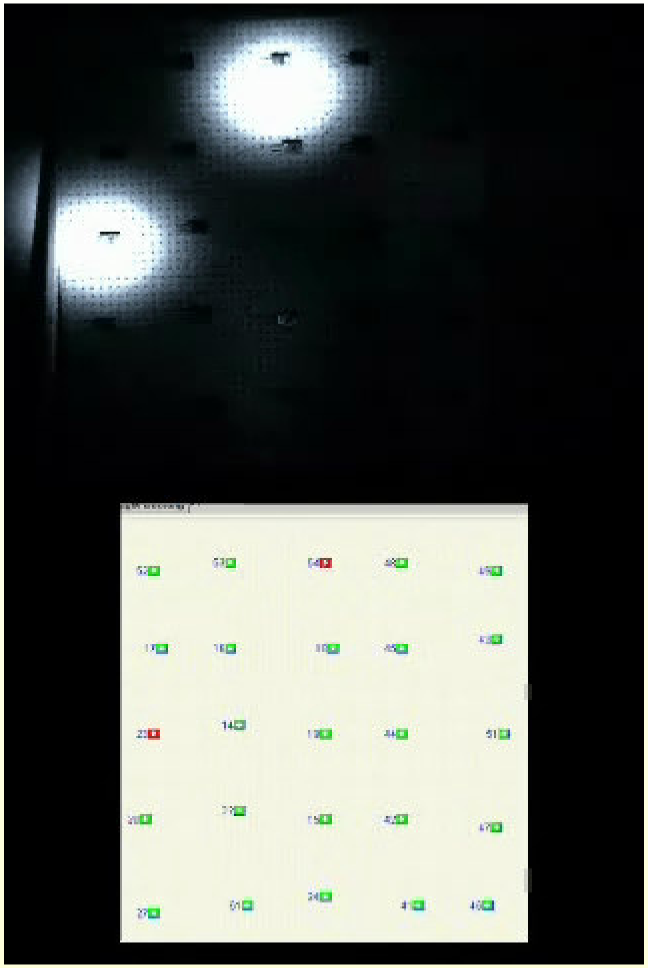


- Different colors mark different sensor clusters formed
- Each cluster has one leader



Target Counting Demo





Simulation with 9 moving targets (above);
Implementation on motes sensors (right)

Some Lessons and Issues

- Sensors naturally form *collaboration groups*. Target localization and counting can be performed in-network.
- These sensor collaboration groups must be maintained as the physical phenomena of interest change over time.
- Aggregates may be easier to sense than individual objects
- Equivalently, physical phenomena are translated into networking behaviors.
- Can such behaviors be programmed without naming the nodes individually?

Conclusion

- An appropriate state representation is crucial
  - Different representations may be needed at different times
  - The distribution of state raises many challenges
- Information utility:
  - Directs sensing to find more valuable information
  - Balances cost of power consumption and benefit of information acquisition

The End