

Lazy Inference on Object Identities in Wireless Sensor Networks

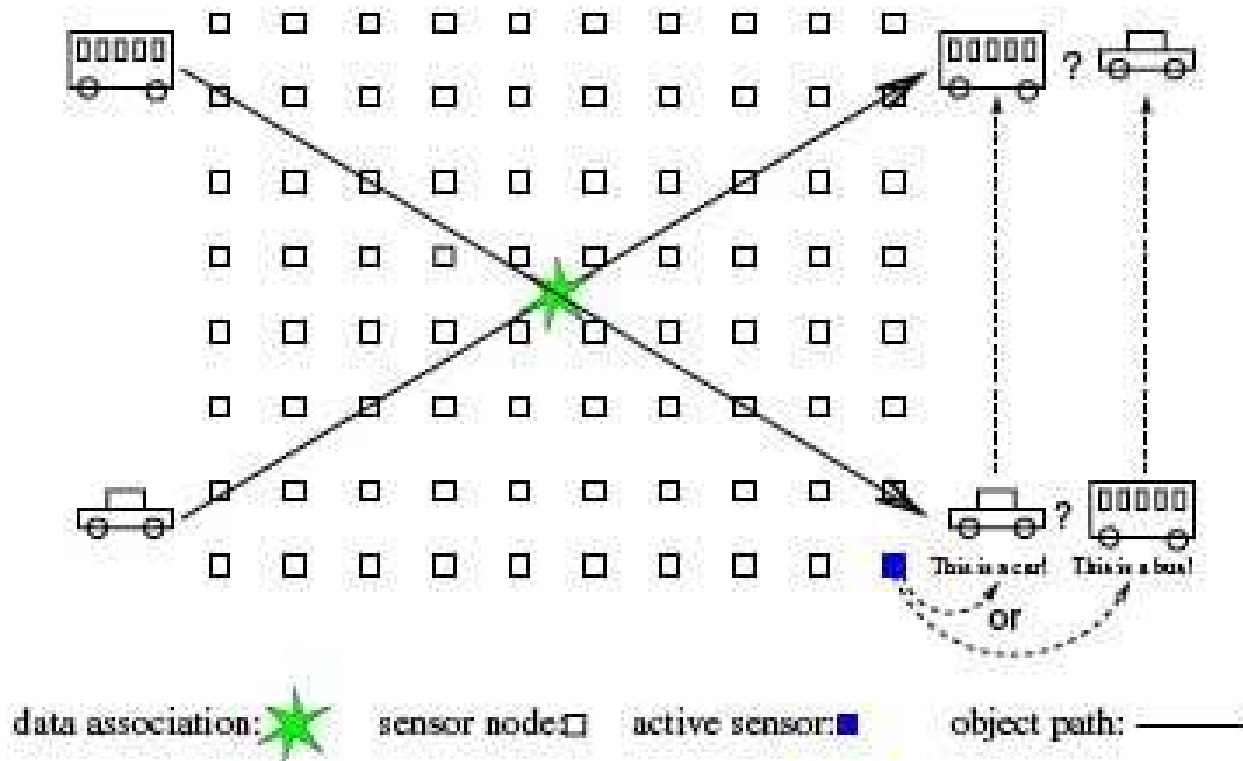
CMPSCI 791T, 04/27/05

Jaewon Shin, Nelson Lee, Sebastian Thrun, and Leonidas Guibas

Presented by Bryan Thibodeau

Identity Management Problem

- **Identity Management Problem:** tracking the identities of N moving objects in a wireless sensor network (WSN)



Identity Management Problem

- The joint identity state of the N objects is $X = (x_1, \dots, x_N)$, where $x_j = i \in \{1, \dots, N\}$ and if $a \neq b$ then $x_a \neq x_b$.
- There are $N!$ different possible joint identity states.
- $X \in S_N$, where S_N is the set of all $N \times N$ permutation matrices.

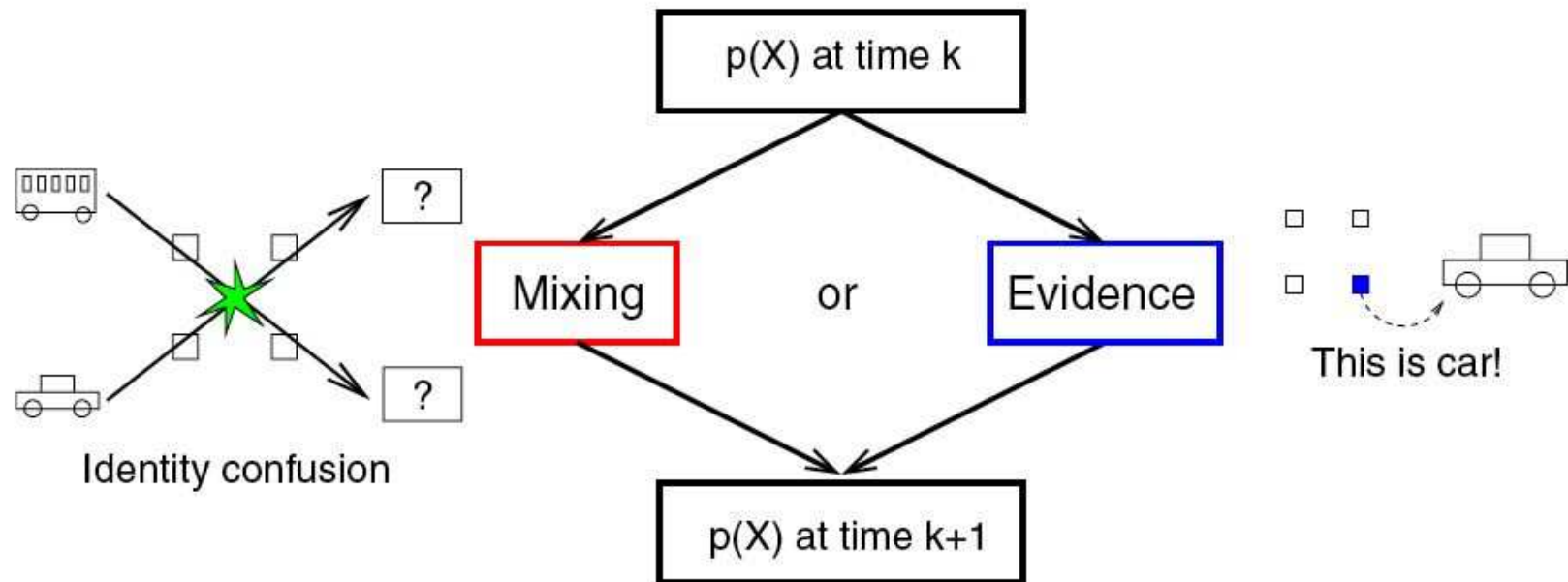
$$X = (x_1 = 1, x_2 = 2, x_3 = 3) \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

$$X = (x_1 = 2, x_2 = 1, x_3 = 3) \rightarrow \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I_{(1,2)}$$

Identity Management Problem

Let $p(X)$, $X \in S_N$ be the joint probability distribution over all $N!$ identity assignments.

Goal: To maintain $p(X)$ on-line while local evidence and mixing events are taking place.



Mixing Events

Mixing occurs when two objects are so close to each other that their positions cannot be distinguished.

Given $p(X)$ and a mixing probability distribution $m(X)$, the joint probability distribution after the mixing event $p'(X)$ can be determined using convolution:

$$p'(X = x) = p * m(X = x) = \sum_{s \in S_N} p(s) m(xs^{-1}),$$

where

$$m(X) = \begin{cases} 1 - \alpha & X = I \\ \alpha & X = I_{(i,j)} \\ 0 & \textit{otherwise} \end{cases}$$

Mixing Events

Simplified version of mixing equation:

$$p * m(X = x) = (1 - \alpha)p(x) + \alpha p(xI_{(i,j)})$$

Example: Mix columns 1 and 2 for $\alpha = 0.5$ and the following initial conditions

$$x_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$x_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ and}$$

$$p(x_1) = 1 \text{ (implies } p(x) = 0 \text{ for all } x \neq x_1\text{)}.$$

Local Evidence

Local evidence is obtained when an object passes by a sensor that can identify it (i.e. an RFID sensor)

The likelihood of an identity measurement Z on the i th object at a sensor node is defined as:

$$p = (Z = (i, j) | X) = \begin{cases} 0.9 & x_j = i \\ 0.1 / (N - 1) & x_j \neq i \end{cases}$$

Applying Bayes rule (assuming measurements are conditionally independent):

$$p(X | \mathbf{Z} = Z_1, \dots, Z_k) \propto p(X) p(Z_1) \cdots p(Z_k | X).$$

Discrete Bayesian Filter

Example

Given events $m_1(X), Z_1, Z_2, m_2(X)$:

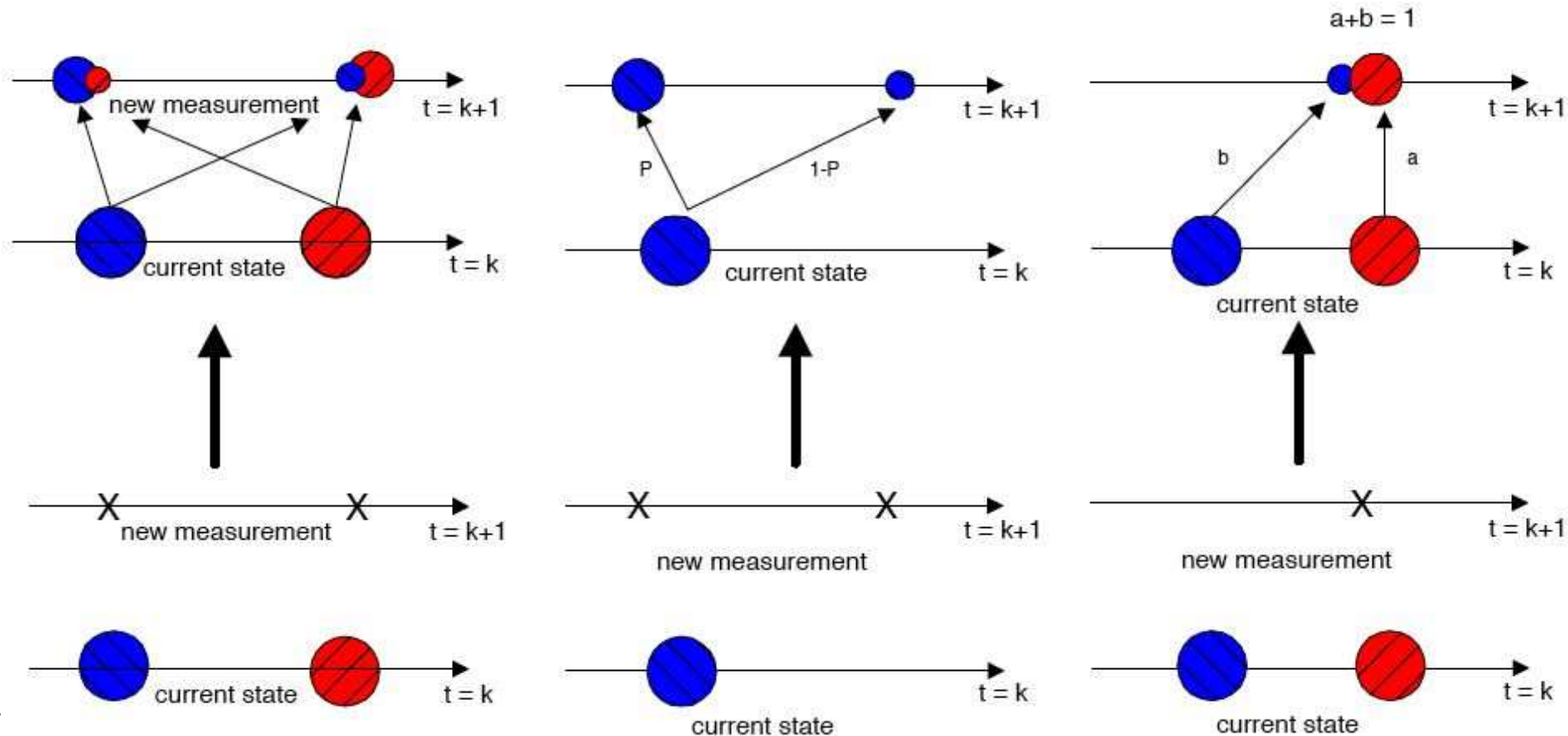
$$p(X) \propto [\{p_0(X) * m_1(X)\} p(Z_1|X)p(Z_2|X)] * m_2(X)$$

- Mixing corresponds to prediction.
- Local evidence corresponds to observation.
- Updating after a mixing is $O((N!)^2)$
- Updating after local evidence $O(N!)$

We need a way to approximate the joint distribution $p(X)$.

Previous Approach: Identity Mass Flow

Basic idea: Initially all objects are assigned a unit identity. The identity mass at time k is determined by the observed positions at time k and time $k - 1$.



Previous Approach: Identity Belief Matrix

$$B(k) = [\mathbf{b}_1(k) \ \mathbf{b}_2(k) \ \cdots \ \mathbf{b}_N(k)] \in [0, 1]^{N \times N}$$

$$\mathbf{b}_i(k) = \begin{bmatrix} p(x_i(k) \text{'s ID is } 1) \\ p(x_i(k) \text{'s ID is } 2) \\ \vdots \\ p(x_i(k) \text{'s ID is } N) \end{bmatrix} \in [0, 1]^{N \times 1}$$

This matrix needs to be renormalized for every mixing or localization event. This is too energy intensive for a wireless sensor network.

Information Matrix Approach

Information Matrix: A $N \times N$ matrix, whose elements l_{ij} are the sum of the log-likelihoods over past observations that object i has identity j .

$$l_{ij} = \sum_t \log(p(Z^t = (k, j) | x_j = i)), k \in \{1, \dots, N\}$$

To update the information matrix after a localization measurement $Z_{(i,j)}$, we add $\log(0.9)$ to l_{ij} and $\log\left(\frac{0.1}{N}\right)$ to all other members of column j .

Information Matrix Example

$$L = \begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix}, \Pi_k = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \log(l(X = \Pi_k)) &= \log(p(Z = (2, 1)|\Pi_k)) \\ &+ \log(p(Z = (1, 2)|\Pi_k)) \\ &+ \log(p(Z = (3, 3)|\Pi_k)) \\ &+ l_{22} + l_{12} + l_{33} \end{aligned}$$

$$l(X = \Pi_k) \propto \exp(\text{Tr}(\Pi_k^T L))$$

Information Matrix

To compute the joint distribution $p(X)$ from the information matrix L :

$$p(X = \Pi_k) = \frac{l_k}{\sum_i l_i} = \frac{\exp(\text{Tr}(\Pi_k^T L))}{\sum_i \exp(\text{Tr}(\Pi_i^T L))}$$

Localization events

Property: Adding a constant to each element of any of the rows or columns of an information matrix does not affect its underlying joint likelihoods.

The elements of the information matrix can now be computed as:

$$l_{ij} = n_{ij} \left(\log(0.9) - \log \left(\frac{0.1}{N-1} \right) \right),$$

where n_{ij} is the number of localization events $Z_{(i,j)}$ observed so far.

Mixing Events

Recall mixing in joint space:

$$p * m(X = x) = (1 - \alpha)p(x) + \alpha p(xI_{(i,j)})$$

Assuming that $\alpha = 0.5$, and substituting for $p(X = \Pi_k)$:

$$\frac{\exp(\text{Tr}(\Pi_k^T L))}{\sum_i \exp(\text{Tr}(\Pi_i^T L))} = \frac{1}{2} [p(\Pi_k) + p(\Pi_k I_{(i,j)})]$$

Yields a system of $\binom{N}{2}$ equations:

$$\frac{q(\Pi_m)}{q(\Pi_n)} = \frac{\exp(\text{Tr}(\Pi_m^T L_q))}{\exp(\text{Tr}(\Pi_n^T L_q))} = \frac{p(\Pi_m) + p(\Pi_m I_{(i,j)})}{p(\Pi_n) + p(\Pi_n I_{(i,j)})}$$

Mixing Events (cont')

Taking the logarithm of both sides:

$$\text{Tr}((\Pi_m - \Pi_n)L_q) = \log \left(\frac{p(\Pi_m) + p(\Pi_m I_{(i,j)})}{p(\Pi_n) + p(\Pi_n I_{(i,j)})} \right)$$

We can write this as:

$$\Phi \vec{\mathbf{I}} = \vec{\eta}$$

where Φ is a $\binom{N!}{2} \times N^2$ matrix, $\vec{\mathbf{I}}$ is a $N^2 \times 1$ vector and $\vec{\eta}$ is a $\binom{N!}{2} \times 1$ vector.

The system is over-constrained, and we can solve it using the pseudo inverse of Φ .

The cost is prohibitive for large N .

More Efficient Mixing

After a mixing event between the i^{th} and j^{th} objects:

- Only the i^{th} and j^{th} columns will be modified.
- The i^{th} and j^{th} columns will be equal to each other after mixing.

Under these assumptions we only need to solve for N unknowns.

Efficient Mixing Example

Let L_p and L_q be the information matrix before and after mixing respectively, and

Let L_p^* and L_q^* be the exponential versions of L_p and L_q respectively

$$L_p^* = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}, L_q^* = \begin{bmatrix} d_1 & d_1 & c_1 \\ d_2 & d_2 & c_2 \\ d_3 & d_3 & c_3 \end{bmatrix}$$

Efficient Mixing Example

Recall:

$$\frac{\exp(\text{Tr}(\Pi_k^T L))}{\sum_i \exp(\text{Tr}(\Pi_i^T L))} = \frac{1}{2} [p(\Pi_k) + p(\Pi_k I_{(i,j)})]$$

This yields the following system of equations:

$$d_1 d_2 c_3 = (a_1 b_2 c_3 + a_2 b_1 c_3) / 2$$

$$d_1 d_3 c_2 = (a_1 b_3 c_2 + a_3 b_1 c_2) / 2$$

$$d_2 d_3 c_3 = (a_2 b_3 c_1 + a_3 b_2 c_1) / 2$$

Efficient Mixing Example

Taking the log on both sides:

$$\log(d_1) + \log(d_2) = \log((a_1b_2 + a_2b_1)/2)$$

$$\log(d_1) + \log(d_3) = \log((a_1b_3 + a_3b_1)/2)$$

$$\log(d_2) + \log(d_3) = \log((a_2b_3 + a_3b_2)/2)$$

Provides a unique solution in the case of $N = 3$.

Mixing for $N > 3$

For $N > 3$ we get $\binom{N}{2}$ equations of the form:

$$\log(d_m) + \log(d_n) = \log((a_m b_n + b_m a_n)/2)$$

which we can write in matrix form as:

$$\mathbf{P} \cdot \boldsymbol{\beta} = \boldsymbol{\gamma}$$

where,

- $\boldsymbol{\beta} = [\log(d_1) \cdots \log(d_N)]^T$ is a $(N \times 1)$ vector
- $\boldsymbol{\gamma} = [\cdots \log((a_m \cdot b_n + b_m \cdot a_n)/2) \cdots]$, is a $\binom{N}{2} \times 1$ vector
- \mathbf{P} is a $\binom{N}{2} \times N$ vector.

Mixing for $N > 3$

$\mathbf{P} \cdot \beta = \gamma$ is an over-constrained system, we can find a least-squares approximation using the pseudo inverse:

$$\beta = \mathbf{P}^\dagger \gamma$$

This gives a computational complexity for mixing of $O(N^4)$.

Complexity

To compute the marginal probability p_{ij} of the j^{th} object having the i^{th} identity, we need to compute $N!$ joint probabilities. This requires $(N^3 N!)$ operations, which is infeasible.

Metropolis sampling was used as a heuristic to estimate the joint probabilities.

Complexity

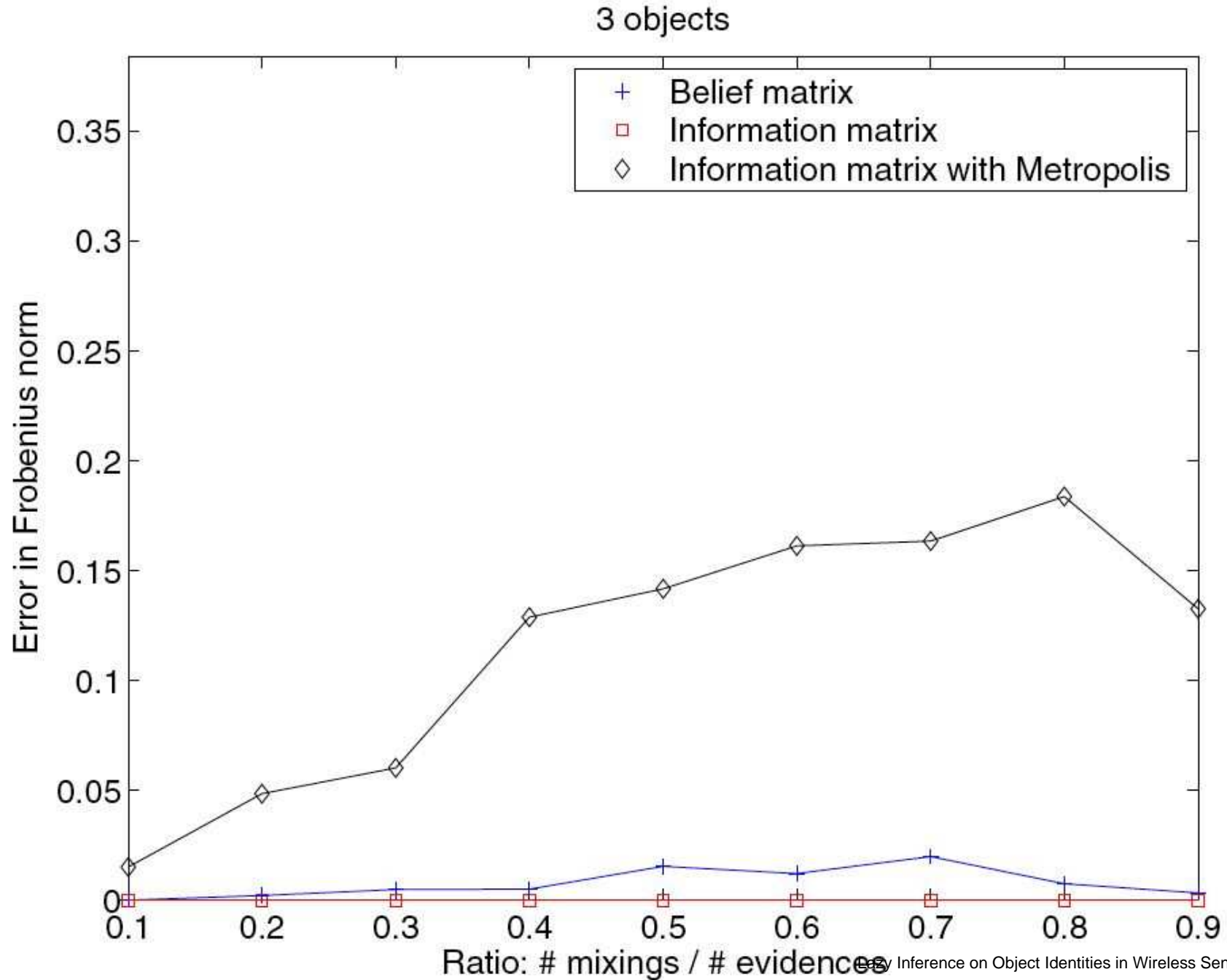
	Information Matrix	Belief Matrix
Local Evidence	$O(1)$	$O(N^2)$
Mixing	$O(N^4)$	$O(N)$
Storage	$O(N^2)$	$O(N^2)$

To update the belief matrix using local evidence, the entire belief matrix must be normalized requiring communication among many nodes, but all updates to the information matrix can be done locally...

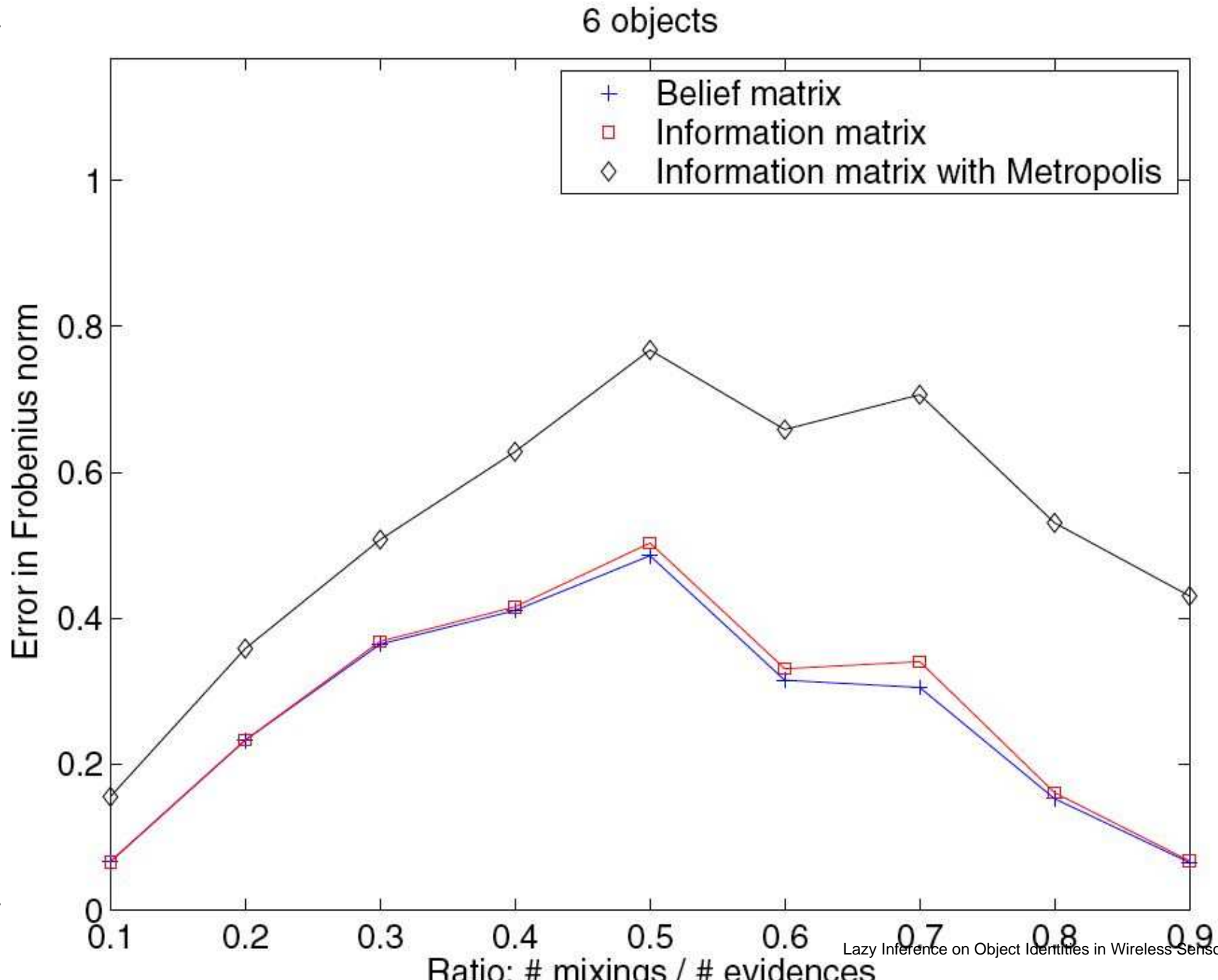
Simulation Setup

- Fifty events in each simulation
- The ratio of mixing to local evidence events was varied over different trials.
- Measured the difference between the marginal distributions given by the approximation methods and the true marginal probabilities.

Simulation Results for 3 Objects

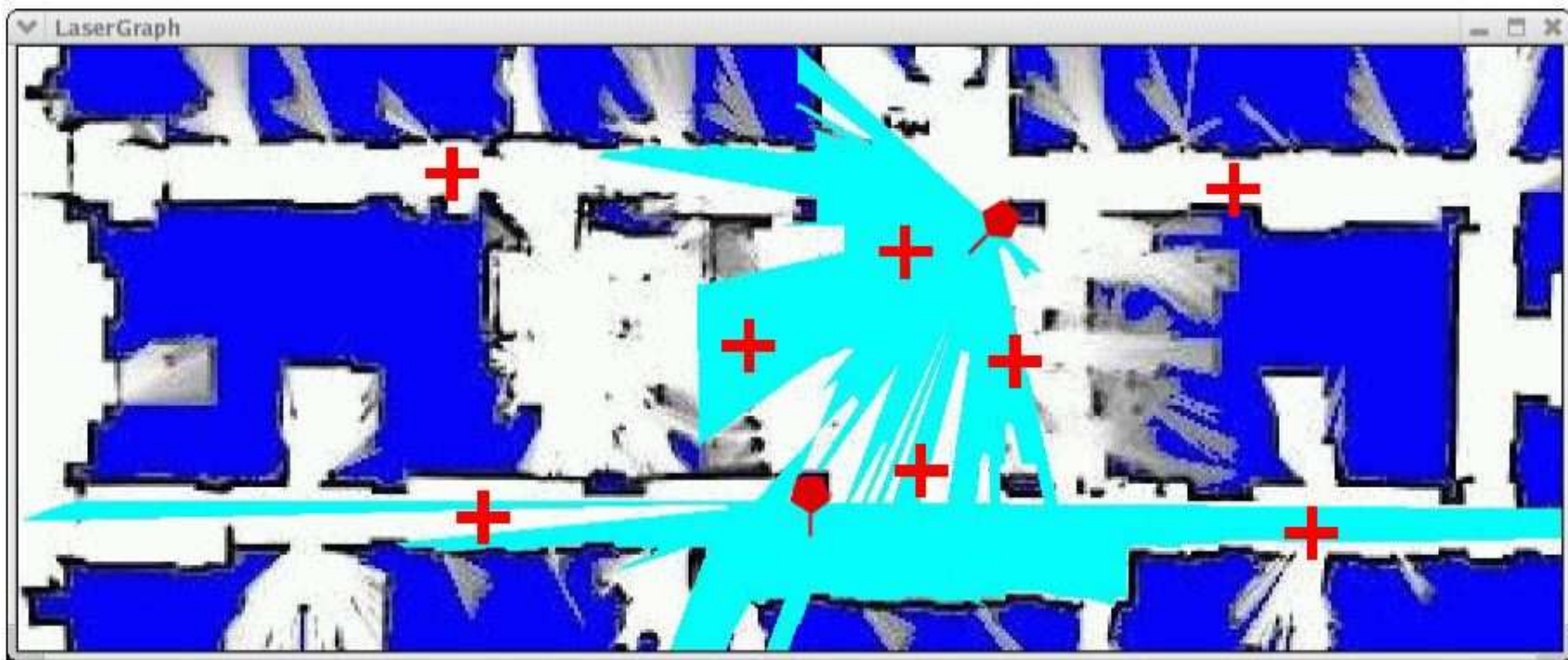


Simulation Results for 6 Objects



Experimental Setup

- 3 individuals carrying RFID tags
- 4 stationary RFID readers
- laser range finders?



Laser sensor

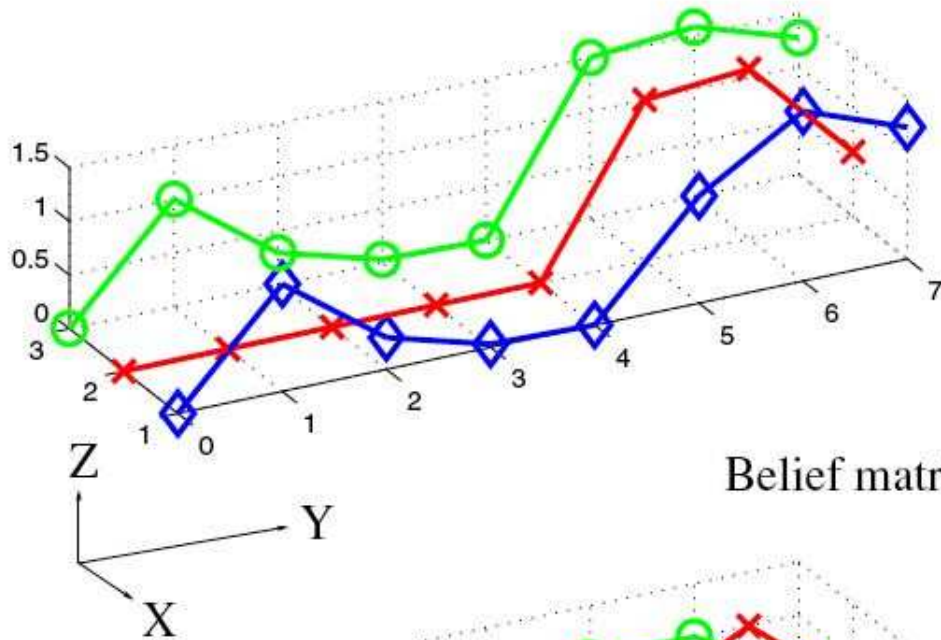


RFID reader

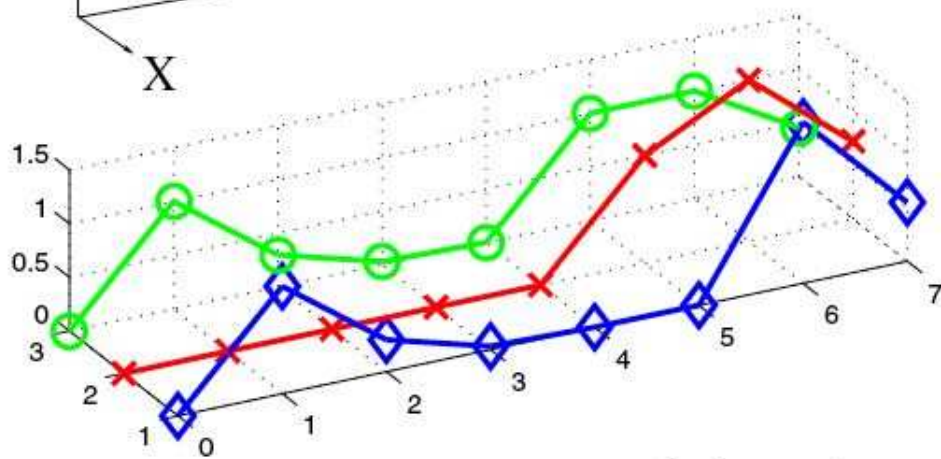


Laser sensing range

Experimental Results



0.4763	0.0545	0.4692
0.0473	0.8911	0.0616
0.4763	0.0545	0.4692



0.9046	0.0952	0.0002
0.0476	0.8571	0.0952
0.0478	0.0476	0.9046