Localization

CS 447– Wireless Embedded Systems
Outline

• Overview
• Distance Measuring Techniques
• Location Estimation Techniques
• Algorithm Types
• Centralized Algorithms
• Distributed Algorithms
• Distributed-Centralized Algorithms
Overview

Accurate sensor location info is **critical** for some domains

- E.g., Geophysics – need to know geophone locations to compute inversions
Overview

GPS is an option, but
- GPS doesn’t work indoors
- GPS is expensive ($)
- GPS consumes much power

• NOTE: sometimes GPS is necessary for “anchors”
  • More on this later.
Overview

Localization – process of calculating geometric location of wireless nodes

Many different algorithms, e.g.,
• TSL
• MDS-MAP
• dwMDS
• DV-Distance
• Robust Quad

I’ll briefly summarize these later
Overview

Current research is **simulation based**

- Assumes **noisy disk** model of radio ranging

Noisy disk model

- Spherical transmission pattern with Gaussian noise
- Not realistic!
Overview

Noisy disk model of radio ranging
Overview

More realistic transmission patterns
Overview

In general, localization algorithms have two steps:
1. Estimate inter-node distances
2. Calculate locations based on inter-node distances
Outline

• Overview
• **Distance Measuring Techniques**
• Location Estimation Techniques
• Algorithm Types
• Centralized Algorithms
• Distributed Algorithms
• Distributed-Centralized Algorithms
Measuring Distance

• Time of Arrival
• Received Signal Strength (RSS)
• Connectivity
Measuring Distance

Time of Arrival

- Travel time of radio signal from transmitter to receiver
- Velocity is speed of light
- Requires highly synchronized clocks
- Difficult to do without expensive hardware

\[
\text{distance} = \text{velocity} \times \text{time}
\]
Measuring Distance

Received Signal Strength (RSS)

- Distance estimated based on RSS and assumed properties of radio signal attenuation

RSS

Measured in dBm
i.e., milliwatt decibals
Measuring Distance

Received Signal Strength (RSS)

• RSS modeled as monotonically decreasing log-normal function that relates distance \((d)\) to receiver path loss \((P_r)\)

\[
P_r(d) = P_o(d_o) - 10n_p \log_{10}\left(\frac{d}{d_o}\right) + X_\alpha
\]

• Basically: farther away => decreased RSS
Measuring Distance

Received Signal Strength (RSS)

\[ P_r(d) = P_0(d_0) - 10n_p \log_{10}(\frac{d}{d_0}) + X_\alpha \]

- \( P_r(d) \) - path loss at receiver at distance \( d \) from transmitter
- \( P_0(d_0) \) - reference path loss (near transmitter)
- \( n_p \) - path loss exponent: models environmental effects
- \( X_\alpha \) - Gaussian distribution to model shadowing effects
Measuring Distance

- XBee radios track RSSI values for received packets
  - RSSI – Received Signal Strength Indicator
- XBee has PWM RSSI pin
- XBee API mode: RSSI value including in wireless packet
Measuring Distance

- Xbee RSSI is proportional to distance...
Measuring Distance

**Connectivity** – uses packet *hop count* to estimate distances

- Anchor nodes broadcast location
- All nodes that hear anchor broadcast are within one hop
- Distance estimated b/t anchors and other based on expected one-hop communication propagation length

- Coarse measurement
Measuring Distance

Connectivity – uses packet *hop count* to estimate distances

three hops

two hops

one hop

anchor
Outline

• Overview
• Distance Measuring Techniques
• Location Estimation Techniques
• Algorithm Types
• Centralized Algorithms
• Distributed Algorithms
• Distributed-Centralized Algorithms
Estimate Location

- Multi-dimensional Scaling (MDS)
- Linear Programming
- Statistical Estimation
- Trilateration
Estimate Location

Multi-dimensional Scaling
• Takes distance matrix between all nodes (aka proximity matrix)
• Outputs a coordinate matrix that minimizes some loss function
• Loss function called stress

E.g., given matrix of pairwise city distances, use MDS to estimate their x,y coordinates on a map
Multi-dimensional Scaling

<table>
<thead>
<tr>
<th>Index</th>
<th>Atlanta</th>
<th>Austin</th>
<th>Baltimore</th>
<th>Boston</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>0</td>
<td>1315.28</td>
<td>927.35</td>
<td>1505.11</td>
</tr>
<tr>
<td>Austin</td>
<td>1315.28</td>
<td>0</td>
<td>2166</td>
<td>2724.01</td>
</tr>
<tr>
<td>Baltimore</td>
<td>927.35</td>
<td>2166</td>
<td>0</td>
<td>577.85</td>
</tr>
<tr>
<td>Boston</td>
<td>1505.11</td>
<td>2724.01</td>
<td>577.85</td>
<td>0</td>
</tr>
<tr>
<td>Chicago</td>
<td>944.4</td>
<td>1571.76</td>
<td>973.23</td>
<td>1366.63</td>
</tr>
<tr>
<td>Dallas</td>
<td>1157.42</td>
<td>293.52</td>
<td>1947.28</td>
<td>2490.97</td>
</tr>
<tr>
<td>Denver</td>
<td>1945.42</td>
<td>1240.77</td>
<td>2422.32</td>
<td>2838.62</td>
</tr>
<tr>
<td>Houston</td>
<td>1126.72</td>
<td>235.7</td>
<td>2010.47</td>
<td>2578.59</td>
</tr>
<tr>
<td>Indianapolis</td>
<td>687.11</td>
<td>1495.43</td>
<td>819.41</td>
<td>1295.31</td>
</tr>
<tr>
<td>Jacksonville</td>
<td>458.63</td>
<td>1542.85</td>
<td>1096.09</td>
<td>1635.65</td>
</tr>
<tr>
<td>Las Vegas</td>
<td>2801.21</td>
<td>1741.71</td>
<td>3377.44</td>
<td>3809.81</td>
</tr>
<tr>
<td>Los Angeles</td>
<td>3108.01</td>
<td>1970.89</td>
<td>3722.45</td>
<td>4166.43</td>
</tr>
<tr>
<td>Memphis</td>
<td>541.43</td>
<td>900.47</td>
<td>1273.23</td>
<td>1824.47</td>
</tr>
</tbody>
</table>
Estimate Location

Linear Programming

• Localization formed as convex optimization problem
• Use mathematical model to estimate best outcome for node locations that minimize location estimate errors

Basics idea:

• Setup system of linear inequalities
• Have linear objective function to optimize (given constraints)
• Want to find max (or min) objective function among all points in feasible region
Linear Programming

• Basically, linear programming finds the min and max points of a feasible region for some objective function (dotted lines).
Estimate Location

Statistical Estimation

• Uses *maximum likelihood* (ML) estimator
• Select parameters of underlying statistical model (i.e., mote location estimates) given inter-mote distances that maximize the likelihood distribution:

\[ \hat{X} = \arg\max_X f(Z|X) \]

• \( Z \) – vector of inter-mote distance measurements
• \( X \) – potential coordinate vector of non-anchors
• \( f(Z \mid X) \) – conditional probability of \( Z \) given \( X \)
• \( X_{\text{hat}} \) – estimated locations
Estimate Location

Trilateration

• Geometric estimation technique
• Uses distances measurements and geometry of shapes
  • Spheres
  • Triangles
  • Bezier curves

• To work, known anchor locations must be
  • Non-collinear (in 2D)
  • Non-coplanar (in 3D)
Estimate Location

Trilateration
Outline

• Overview
• Distance Measuring Techniques
• Location Estimation Techniques
• Algorithm Types
  • Centralized Algorithms
  • Distributed Algorithms
  • Distributed-Centralized Algorithms
Algorithm Types

- Centralized
- Distributed
- Distributed-Centralized
- Anchor-Based
- Cooperative
Algorithm Types

Centralized

- Uses base station to calculate location
- Requires all motes transmit information to base station
Algorithm Types

Distributed
• Motes calculate its location in-network
• Decrease energy consumption (fewer Tx)
• Sacrifice accuracy and precision
Algorithm Types

Distributed-Centralized

- Run centralized algorithms on overlapping clusters in network
- Alleviates need to transmit all location data to single node
Algorithm Types

Anchor-based
• Uses motes with known locations to infer locations of others within the network

Cooperative
• No mote knows its own location before algorithm runs
Outline

• Overview
• Distance Measuring Techniques
• Location Estimation Techniques
• Algorithm Types
• Centralized Algorithms
• Distributed Algorithms
• Distributed-Centralized Algorithms
Centralized Algorithms

- MDS-MAP
- TSL
Centralized Algorithms

MDS-MAP

- Uses MDS to produce node coordinates that are best-fit for all measured inter-mote distances

- MDS is arbitrary in rotation and translation

- MAP – normalize coordinates with known anchor points
  - Rotate
  - Translate
Centralized Algorithms

TSL – Temporal Stability Localization
• Uses one-hop communication ranges and distance measurements

• Creates system of constraints to refine each mote’s possible location
  • Based on RSS or connectivity

• Uses gradient based local search algorithm
Outline

• Overview
• Distance Measuring Techniques
• Location Estimation Techniques
• Algorithm Types
• Centralized Algorithms
• Distributed Algorithms
• Distributed-Centralized Algorithms
Distributed Algorithms

DV-Distance

• Approximates distance between non-anchor mote $i$ and anchor mote $j$ as the shortest path $sp_{ij}$ between motes $i,j$

• Uses $sp_{ij}$ to constrain $(x_i, y_i)$ of non-anchor mote $i$ in terms of anchor $j$

\[ 0 = (x_i - x_j)^2 + (y_i - y_j)^2 - sp_{ij}^2. \]

• System of linear equations setup with at least three anchors ($j$)

• Equations solved for $i$’s coordinates using least squares method
Outline

• Overview
• Distance Measuring Techniques
• Location Estimation Techniques
• Algorithm Types
• Centralized Algorithms
• Distributed Algorithms
• Distributed-Centralized Algorithms
Distributed-Centralized

- dwMDS
- Robust Quadrilateral
Distributed-Centralized

dwMDS

- Distributed, weighted, MDS
- MDS protocol on clusters within network
- Pre-process distance estimates before MDS
  - Parameter $r$ (radius) – smaller than mote range
  - Uses $r$ to prune off large communication links
  - (if distance larger than $r$, dropped from MDS)
Distributed-Centralized

Robust Quadrilateral

• Uses inter-mote ranging info to identify all robust quadrilaterals that exist between neighbors

• Triangle of three nodes defined as robust if smallest angle (alpha) in triangle satisfies: \( d \cdot \cos^2(\alpha) > \Theta \)
  
  • \( d \) – shortest edge in triangle
  • \( \Theta \) – threshold parameter
Distributed-Centralized

Robust Quadrilateral

- Robust quad: iff all triangles in quad are robust

- Creates subgraph of overlapping quads within clusters

- Then uses trilateration to estimate location...