MobiLoc: Mobility Enhanced Localization

Prabal Dutta
Sarah Bergbreiter
Acknowledgments

• Cory Sharp
  – Magnetometer
  – TinyOS Environment

• Fred Jiang
  – Ultrasonic transducer

• Kamin Whitehouse
  – Localization discussions
  – Matlab algorithms

• Rob Szewczyk
  – Conceptual discussions
  – Photoshop magic
  – Late night runs to Nation’s
Outline of the Talk

- Observations
- Motivation
- Problem Formulation
- Related Work
- Linear Trajectory
- General Trajectory
- Experimental Setup
- Results
- Conclusions
- Future Work
Observations

- Many sensor network localization algorithms depend on ranging of neighbors.

- Ranging can be a random function of space, which leads to ranging errors (e.g. multi-path and occlusions).

- Ranging may require specialized hardware, which adds cost and is only useful for localization.

- Tracking is a canonical application in which sensors may be able to range targets.

- Lazy localization – May not need localization until target actually appears (exception: geographic routing).
Motivation

• In certain applications, sensors may be **required** to range targets but not other sensors

• What if we could avoid using special hardware for localization only?

• What if we could average range readings over time *and* space, allowing better diversity and reducing effects of multi-path and occlusions?

• What if we could indirectly range neighbors by ranging a target and then combining readings?
Problem Formulation

• Given
  – A mobile (and potentially) adversarial target
  – A time synchronization service
  – An ability to range a mobile target
  – An inability to range neighboring sensor nodes directly

• Find
  – An algorithm to determine the distance to nearby neighbors
Related Work

  - Friendly mobile beacon
  - Beacon is GPS-equipped
  - Using RSSI for ranging

- Other work in progress…
GPS Beacon Position Estimate

How Much Does GPS Wander?

• In our test data set:
  – Min Lat: 48° 43.410’
  – Max Lat: 48° 43.427’
  – Min Lon: 123° 23.932’
  – Max Lon: 123° 23.912’

• Which means in 24 hours
  – Latitude variation:
    • 0.017 * 1853 * 0.67 = 21m
  – Longitude variation:
    • 0.020 * 1853 * 0.67 = 25m

• Wander
  – 0.75m/s (max wander rate)
  – 0.1m/s (typical wander rate)

• What about ground truth?
  – Don’t know if mean converges to actual location

• Let’s consider another idea…

SIDEBAR

• Longitude, at the equator
  – 1 degree = 1° = 111.2km
  – 1 minute = 1’ = 1853m
  – 1 second = 1” = 30.9m

• At other latitudes
  – Multiply by conversion factor: \( \cos(\text{latitude}) \)
  – \( \cos(48°) = 0.67 \)

• Latitude computation similar

Note: Computations may be slightly off
Linear Trajectory – Between Sensors

\[ d = \sqrt{r_{1,1}^2 + r_{2,1}^2 - 2r_{1,1}r_{2,1} \cos(\pi/2 + \theta_2)} \]

\[ \hat{d} = \sqrt{r_{1,1}^2 + r_{2,1}^2 + 2r_{1,1}r_{2,2}} \]

\[ \hat{\hat{d}} = \sqrt{r_{2,2}^2 + r_{1,2}^2 + 2r_{1,1}r_{2,2}} \]
## Error Sensitivity – Trajectory Between Sensors

<table>
<thead>
<tr>
<th>Error</th>
<th>Range Estimates</th>
<th>Distance Estimates</th>
<th>Estimation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r11</td>
<td>r12</td>
<td>r21</td>
</tr>
<tr>
<td>0%</td>
<td>3.00</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>10%</td>
<td>3.30</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>20%</td>
<td>3.60</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>30%</td>
<td>3.90</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>10%</td>
<td>3.00</td>
<td>4.40</td>
<td>4.00</td>
</tr>
<tr>
<td>20%</td>
<td>3.00</td>
<td>4.80</td>
<td>4.00</td>
</tr>
<tr>
<td>30%</td>
<td>3.00</td>
<td>5.20</td>
<td>4.00</td>
</tr>
<tr>
<td>-10%</td>
<td>3.00</td>
<td>4.00</td>
<td>3.60</td>
</tr>
<tr>
<td>-20%</td>
<td>3.00</td>
<td>4.00</td>
<td>3.20</td>
</tr>
<tr>
<td>-30%</td>
<td>3.00</td>
<td>4.00</td>
<td>2.80</td>
</tr>
<tr>
<td>-10%</td>
<td>3.00</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>-20%</td>
<td>3.00</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>-30%</td>
<td>3.00</td>
<td>4.00</td>
<td>4.00</td>
</tr>
</tbody>
</table>

\[
\hat{d} = \sqrt{r_{1,1}^2 + r_{2,1}^2 + 2r_{1,1}r_{2,2}} \\
\hat{d} = \sqrt{r_{2,2}^2 + r_{1,2}^2 + 2r_{1,1}r_{2,2}}
\]
Linear Trajectory – Not Between Sensors

\[ \hat{d} = \sqrt{r_{1,1}^2 + r_{2,1}^2 - 2r_{1,1}r_{2,1}} \]

\[ \hat{d} = \sqrt{r_{2,2}^2 + r_{1,2}^2 - 2r_{1,1}r_{2,2}} \]
## Error Sensitivity – Trajectory Not Between Sensors

<table>
<thead>
<tr>
<th>Error</th>
<th>Range Estimates</th>
<th>Distance Estimates</th>
<th>Estimation Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r11  r12  r21  r22</td>
<td>d1   d2</td>
<td>d1   d2</td>
</tr>
<tr>
<td>0%</td>
<td>3.00  5.00  5.00  3.00</td>
<td>4.00  4.00</td>
<td>0.0%  0.0%</td>
</tr>
<tr>
<td>10%</td>
<td>3.30  5.00  5.00  3.00</td>
<td>4.01  3.77</td>
<td>0.3%  -5.8%</td>
</tr>
<tr>
<td>20%</td>
<td>3.60  5.00  5.00  3.00</td>
<td>4.04  3.52</td>
<td>1.1%  -12.0%</td>
</tr>
<tr>
<td>30%</td>
<td>3.90  5.00  5.00  3.00</td>
<td>4.10  3.26</td>
<td>2.5%  -18.6%</td>
</tr>
<tr>
<td>10%</td>
<td>3.00  5.50  5.00  3.00</td>
<td>4.00  4.61</td>
<td>0.0%  15.2%</td>
</tr>
<tr>
<td>20%</td>
<td>3.00  6.00  5.00  3.00</td>
<td>4.00  5.20</td>
<td>0.0%  29.9%</td>
</tr>
<tr>
<td>30%</td>
<td>3.00  6.50  5.00  3.00</td>
<td>4.00  5.77</td>
<td>0.0%  44.2%</td>
</tr>
<tr>
<td>-10%</td>
<td>3.00  5.00  4.50  3.00</td>
<td>3.35  4.00</td>
<td>-16.1% 0.0%</td>
</tr>
<tr>
<td>-20%</td>
<td>3.00  5.00  4.00  3.00</td>
<td>2.65  4.00</td>
<td>-33.9% 0.0%</td>
</tr>
<tr>
<td>-30%</td>
<td>3.00  5.00  3.50  3.00</td>
<td>1.80  4.00</td>
<td>-54.9% 0.0%</td>
</tr>
<tr>
<td>-10%</td>
<td>3.00  5.00  5.00  2.70</td>
<td>4.22  4.01</td>
<td>5.5%  0.3%</td>
</tr>
<tr>
<td>-20%</td>
<td>3.00  5.00  5.00  2.40</td>
<td>4.43  4.04</td>
<td>10.7% 1.1%</td>
</tr>
<tr>
<td>-30%</td>
<td>3.00  5.00  5.00  2.10</td>
<td>4.63  4.10</td>
<td>15.7% 2.5%</td>
</tr>
</tbody>
</table>

\[
\hat{d} = \sqrt{r_{1,1}^2 + r_{2,1}^2 - 2r_{1,1}r_{2,2}}
\]

\[
\hat{d} = \sqrt{r_{2,2}^2 + r_{1,2}^2 - 2r_{1,1}r_{2,2}}
\]
Ambiguity – Leads to Two Solutions

- How to distinguish between these two cases?
  - Short answer: You can’t
  - Long answer: Take the long view
Algorithm – Assumptions

- Assumptions
  - Target trajectory linear
  - Target velocity constant
  - Sensing radius >> node separation
- Let
  - $t_0 =$ time of first presence
  - $t_1 =$ time of last presence
  - $r_{\text{CPA}} =$ range at CPA
  - $t_{\text{CPA}} =$ time of CPA
- Call the node active between $t_0$ and $t_1$
Algorithm

- Save samples from \([t_0, t_1]\)
- Wait until not active \((t > t_1)\)
- Broadcast message\((t_0, t_1, r_{CPA}, t_{CPA}, \text{linearity})\)
- Upon receiving a message from \(q\), a node \(p\) checks to see if it is active
- If active, then \(p\) waits until it is not active
- Node \(p\) checks the following predicate \(P\) to determine an overlap

\[
P = (p.t_0 \leq q.t_{CPA} \leq p.t_1) \land (q.t_0 \leq p.t_{CPA} \leq q.t_1)
\]

- If \(P\) is true and both trajectories are linear, then \(p\) queries \(q\) for \(q\)'s range to the target at \(p\)'s time of CPA. Node \(p\) also includes in this query \(p\)'s range to the target at both \(p\) and \(q\)'s time of CPA
- Nodes \(p\) and \(q\) both compute their distance \(d\) using the four different equations and save these estimates in a windowed buffer
- Estimate \(d\), the mode of buffer, as: \(d = 2 \times \text{median} - \text{mean}\)
- So what if the trajectory is not linear?
Non-Linear Trajectory – Between Sensors

\[ r^+(t) = r_1(t) + r_2(t) \]

Jan 15, 2004  MobiLoc  17
Non-Linear Trajectory – Not Between Sensors

\[ r^-(t) = |r_1(t) - r_2(t)| \]
Algorithm

- Save samples from \([t_0, t_1]\)
- Wait until not active \((t > t_1)\)
- Broadcast message \((t_0, t_1, r_{CPA}, t_{CPA},\) linearity\)
- Upon receiving a message from \(q\), a node \(p\) checks to see if it is active
- If active, then \(p\) waits until it is not active
- Node \(p\) checks the following predicate \(P\) to determine an overlap
  \[ P = (p.t_0 \leq q.t_{CPA} \leq p.t_1) \land (q.t_0 \leq p.t_{CPA} \leq q.t_1) \]

- If \(P\) is true and either trajectory is not linear, then \(p\) queries \(q\) for its entire set of samples between \([t_0, t_1]\).
- Node \(p\) computes the minimum sum and maximum difference, saves these estimates in a windowed buffer, and transmits to \(q\)
- Estimate \(d\), the mode of buffer, as: \(d = 2 \times \text{median} - \text{mean}\)
- Looking for a better algorithm…
Experimental Setup

- Mica2Dot PEG
- 250 ms “chirp” (effective $f_s$)
- Unrealistic but convenient test platform
Results – Overall

Error over multiple Trajectories

Mean Squared Error

Jan 15, 2004  MobiLoc
Results – Sum and Difference vs. Actual Distance

[Graphs showing distance variations over sample numbers for different pairs of nodes.]
Conclusion

• We have shown how to:

  – Avoid using special hardware for localization only

  – Average range readings over time *and* space, allowing better diversity and reducing effects of multi-path and occlusions

  – How to indirectly range neighbors by ranging a target and then combining readings?
• **Definition**: The *range graph* is the planar geometric structure whose six vertices are defined by the location of three unknown nodes $N_1$, $N_2$, $N_3$ and of a mobile object at three times $t_1$, $t_2$, $t_3$, and whose edges are the nine ranges $r_{i,j}$.

• **Theorem 1**: The range graph is rigid.

• **Theorem 2**: The range graph is unique if $T_1$, $T_2$, and $T_3$ are non-collinear, $N_1$, $N_2$, and $N_3$ are non-collinear, and $r_{i,j}$, $\forall i,j \in \{1,2,3\}$, are error-free.

---

Future Work – Range Graph
Discussion