ENHANCING IBEACON BASED MICRO-LOCATION WITH PARTICLE FILTERING

Faheem Zafari - Ioannis Papapanagiotou
Computer & Information Technology
Purdue University
faheem0@purdue.edu
ipapapa@ncsu.edu
• **Indoor Localization**
  • Challenging due to inherent environment complexities.
  • Can be leveraged to provide a number of location based services.
**iBeacons**

- Apple’s protocol for Bluetooth Low Energy (BLE) Devices.
  - Proximity based services
- Utilizes 2.4 GHz.
- Device with iOS 7.0+ and Android 4.3+ pick beacon signals
- Message consists of
  - 16 bytes mandatory Universally Unique Identifier (UUID) field.
  - 2 byte optional major field.
  - 2 byte optional major field.
IBEACON BASED PROXIMITY SERVICES

- User device contacts a server for information about the message.
• Indoor localization requires high accuracy.
  → *Current systems are not accurate enough.*
• Beacons poised for wide-scale proliferation.
  → *In current state, only used for proximity based services.*
• Beacons have not been used for indoor localization.
  → *Not accurate without any filtering algorithm.*

**Need a filtered ibeacon based localization system**
• Design an ibeacon based indoor localization system using particle filtering:
  1. Formulate tracking problem as non-linear Bayesian tracking problem.
  2. Develop an iOS mobile application with particle filtering.
  3. Utilize ibeacons for indoor localization.
  4. Reduce the localization error.
\( y_i \), where \( i \in N \) is the target’s state sequence at time \( i \).

- \( y_i = g_i(y_{i-1}, m_{i-1}) \) where
  - \( g_i \) is a non-linear function
  - \( m_{i-1} \) is the i.i.d process noise

- \( z_i \) is the obtained measurement at time \( i \).
  - \( z_i = h_i(y_i, n_i) \) where
    - \( h_i \) is a non-linear function
    - \( n_i \) is i.i.d measurement noise.

**Working Principle:**
- Recursively calculate the belief in state \( y_i \) at time \( i \) using measurements \( z_{1:i} \) obtained up to time \( i \).
• **Prediction Stage**
  • Predict the next state of the user using Chapman-Kolmogorov equation.

\[
p(y_i|z_{1:i-1}) = \int p(y_i|y_{i-1})p(y_{i-1}|z_{1:i-1})dy_{i-1}
\]

• **Update Stage**
  • Update the prediction based on obtained measurements.

\[
p(y_i|z_{1:i}) = \frac{p(z_i|y_i)p(y_i|z_{1:i-1})}{p(z_i|z_{i-1})}
\]

where

\[
p(z_i|z_{i-1}) = \int p(z_i|y_i)p(y_i|z_{i-1})dy_i
\]
The prediction and update state are computationally exhaustive
  • Not feasible for real-time processing.

Particle filtering is a discrete approximation of the problem
  • Particles to represent the state (pmf)
  • Assign weights to particles based on probability.

\[ p(y_i | z_{1:i}) \approx \sum_{k=1}^{Ns} w_i^k \delta(y_i - y_i^k) \]
EXPERIMENTAL SETUP

• An iOS application developed.

• Two different scenarios
  • 1m x 1m
  • 11m x 6m

• Variables
  • Number of beacons
  • Number of particles
  • Tracking area
• Compare the estimated location of user with actual position

\[ < Error > = \frac{\sum_{i=1}^{n} \sqrt{(X_i - X_{est})^2 + (Y_i - Y_{est})^2}}{n} \]

• Experimental parameters

<table>
<thead>
<tr>
<th>Device</th>
<th>Apple iPhone 4s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wireless Interface</td>
<td>Bluetooth V4.0/ 2.GHz</td>
</tr>
<tr>
<td>Operating System</td>
<td>iOS 8.1</td>
</tr>
<tr>
<td>Beacons</td>
<td>Gimbal Series 10</td>
</tr>
<tr>
<td>Gimbal Range</td>
<td>50 meters</td>
</tr>
<tr>
<td>Transmission Frequency</td>
<td>100ms</td>
</tr>
<tr>
<td>Major Value</td>
<td>Yes</td>
</tr>
<tr>
<td>Minor Value</td>
<td>Yes</td>
</tr>
</tbody>
</table>
• 1mx1m scenario

- Increasing the number of beacons, up to a threshold, improves performance.
- After reaching certain number of beacons (5), the performance degrades
  - Due to self-interference
1mx1m scenario: Localization error in meters

<table>
<thead>
<tr>
<th>Particles</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>0.308</td>
<td>0.290</td>
<td>0.303</td>
<td>0.301</td>
</tr>
<tr>
<td>600</td>
<td>0.356</td>
<td>0.308</td>
<td>0.312</td>
<td>0.302</td>
</tr>
<tr>
<td>800</td>
<td>0.396</td>
<td>0.301</td>
<td>0.302</td>
<td>0.31</td>
</tr>
<tr>
<td>1000</td>
<td>0.384</td>
<td><strong>0.276</strong></td>
<td>0.298</td>
<td>0.316</td>
</tr>
<tr>
<td>1200</td>
<td>0.400</td>
<td>0.299</td>
<td><strong>0.293</strong></td>
<td>0.318</td>
</tr>
<tr>
<td>1400</td>
<td>0.403</td>
<td>0.289</td>
<td>0.307</td>
<td>0.315</td>
</tr>
<tr>
<td>1600</td>
<td>0.385</td>
<td>0.314</td>
<td>0.306</td>
<td>0.316</td>
</tr>
<tr>
<td>1800</td>
<td>0.407</td>
<td>0.298</td>
<td>0.299</td>
<td><strong>0.291</strong></td>
</tr>
<tr>
<td>2000</td>
<td>0.411</td>
<td>0.312</td>
<td>0.300</td>
<td>0.304</td>
</tr>
</tbody>
</table>

- Highest accuracy attained with 4 beacons and 1000 particles.
- A certain set of beacons works optimally for a specific number of particles.
  - e.g. 5 beacons work best with 1200 particles
• 11mx6m scenario

• Increase in number of beacons, to a threshold, improves performance.

• After reaching certain number of beacons (7), the performance degrades
  • Due to self-interference
### Average Error (Results)

- **11mx6m scenario**: Localization error in meters

<table>
<thead>
<tr>
<th>Particles</th>
<th>Beacons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>400</td>
<td>2.195</td>
</tr>
<tr>
<td>600</td>
<td>2.167</td>
</tr>
<tr>
<td>800</td>
<td>2.152</td>
</tr>
<tr>
<td>1000</td>
<td>1.736</td>
</tr>
<tr>
<td>1200</td>
<td>1.843</td>
</tr>
<tr>
<td>1400</td>
<td>2.262</td>
</tr>
<tr>
<td>1600</td>
<td>2.049</td>
</tr>
<tr>
<td>1800</td>
<td>1.774</td>
</tr>
<tr>
<td>2000</td>
<td><strong>1.668</strong></td>
</tr>
</tbody>
</table>

- Highest accuracy attained with 5 beacons and 1000 particles.
- A certain set of beacons works optimally for a specific number of particles.
  - e.g. 4 beacons work best with 600 particles.
DEPLOYMENT SUGGESTIONS

• Increasing number of beacons to a certain number ‘n’ will improve performance.
  • ‘n’ depends on space and environment.
  • Going beyond ‘n’ causes self-interference hence degrading performance.
• For every deployment environment, there is an optimal number of beacons and particles.
• Beacons should be placed at higher altitudes and in areas where there it is less prone to interference.
CONCLUSION

• We developed an iBeacon based indoor localization system.
  • Provided mathematical formulation.
  • Utilized particle filtering algorithm for better accuracy.

• The proposed system results in an accuracy of
  • 0.27 meters in 1m x 1m environment
  • 0.97 meters in 11m x 6m environment.

• We provided deployment suggestions based on real-time experiments.
CURRENT WORK

- Implement particle filtering on a server side
- Utilize different filtering algorithms for improving the accuracy of
  - Beacon based indoor localization
  - Beacon based proximity services
- Open sourced the code for public use.
  - [https://github.com/ipapapa/IoT-MicroLocation](https://github.com/ipapapa/IoT-MicroLocation)