Overview on RSSI-based Positioning Algorithms for WPSs

Luca De Nardis and Giuseppe Caso

DIET Department, Sapienza University of Rome
Rome, Italy

Email: luca.denardis@uniroma1.it, caso@diet.uniroma1.it
Summary

- RSSI-based WPSs: Fingerprinting vs Propagation Channel Modeling
  - Fingerprinting approach for Positioning
  - Indoor Propagation Channel Modeling
  - Fingerprinting vs Propagation Channel Model
  - Preliminary experimental results @ DIET Department

- Fingerprinting-based positioning algorithms
  - Enhanced Weighted K-Nearest Neighbors (EWKNN)
  - Compressive Sensing-based
  - Introduction of inertial sensors, maps and mobility models for tracking

- Conclusions

- A EWKNN-based practical implementation @ DIET Department
Fingerprinting approach for Positioning

- The Fingerprinting technique aims at localizing a WiFi device by using a prebuilt radio map of the WiFi coverage over the area of interest.

- It involves two stages:
  1. **Offline Stage**: creation of the radio map (database) by collecting the RSSI readings from available surrounding WiFi Access Points (APs) within the area of interest in particular known and selected positions (Reference Points – RPs).
  2. **Online Stage**: device position estimation by comparing the online RSSI readings of the device with the offline RPs observations (fingerprints), forming the database. Several methods for position estimation:
     - Nearest Neighbor (NN)
     - K-Nearest Neighbors (KNN)
     - Weighted KNN (WKNN)
     - Enhanced WKNN (EWKNN)
     - Statistical Methods using Bayesian theory and kernel functions
     - Compressive Sensing

- Note that Fingerprinting does not need info on the APs position but it requires manual efforts for the database creation and management.
- Moreover, it requires a clear planning regarding the number of RPs and their distribution in the area.
Fingerprinting approach for Positioning
Indoor Propagation Channel Modeling

• Fingerprinting of an indoor venue can be expensive and time consuming. For this reason, another approach could be to using an indoor propagation model, in order to create the radio map and estimate RSSI values in the area.

• This approach can be less accurate than fingerprinting, if the propagation model does not take into account the dynamic and unpredictable nature of the indoor radio channel (shadowing, multipath, device orientation, and so on) but it is computationally simpler (no need of offline phase).

• The major constraint is that it requires info on the APs position but no RPs definition is needed in the area.

• Several models have been deployed for indoor propagation analysis. They can be divided in two different classes:
  1. **Statistical (Empirical) Models**: the signal propagation and its parameters are evaluated within generic areas in a statistical approach.
  2. **Deterministic Models**: they use information on the particular area of interest.
Multi-Wall (MW) Model

- The Multi-Wall (MW) model is a quite known empirical indoor propagation model. Its definition comes from the One-Slope (OS) model that assumes a linear dependence between the path loss and the logarithmic distance between the transmitter and the receiver.

\[ L_{OS}(d) = l_0 + 10\gamma \log(d) \quad (dB) \]

- MW adds a further attenuation term, due to the presence of walls and doors:

\[ L(d) = L_{OS}(d) + M_w \quad (dB) \]

- where

\[ M_w = l_c + \sum_{i=1}^{l} k_{wi} l_i + \sum_{n=1}^{N_d} X_n l_d + \sum_{n=1}^{N_fd} \lambda_n l_{fd} \quad (dB) \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l_c )</td>
<td>Constant (least square fitting procedure from measurements)</td>
</tr>
<tr>
<td>( k_{wi} )</td>
<td># of walls crossed by the signal of type i</td>
</tr>
<tr>
<td>( l_i )</td>
<td>Attenuation introduced by walls of type i (least square fitting procedure)</td>
</tr>
<tr>
<td>( X_n, \lambda_n )</td>
<td>Binary variables indicating if doors are open or closed</td>
</tr>
<tr>
<td>( N_d, N_{fd} )</td>
<td># of doors and fire proof doors crossed by the signal</td>
</tr>
</tbody>
</table>
Fingerprinting vs Propagation Model

\[
\{ \gamma, l_c, l_i, l_d, l_{fd} \}_{\text{opt}} = \arg \min_{\{ \gamma, l_c, l_i, l_d, l_{fd} \}} \left\{ \sum_{m=0}^{M-1} |P_m - (EIRP - L(d_m))|^2 \right\}
\]

Comparison between optimal MW classic model (continuous line) and measurements (dots) – office environment

<table>
<thead>
<tr>
<th>Model</th>
<th>Office (dB)</th>
<th>Classroom (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>αd</td>
<td>7.8</td>
<td>5.1</td>
</tr>
<tr>
<td>OS</td>
<td>7.4</td>
<td>5.1</td>
</tr>
<tr>
<td>MWC</td>
<td>4.3</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Comparison between optimal MW classic model (continuous line) and measurements (dots) – classroom environment

Preliminary experimental results @ DIET

Access Point 1

Access Point 2
Fingerprinting-based positioning algorithms: EWKNN

- The KNN algorithm selects and combines the nearest K neighbors (RPs' fingerprints) around a device to determine its position.

- Using a fixed number (K) of fingerprints may decrease positioning accuracy: if K is not changed during the positioning process, sometimes, RPs far from the device might be included in the KNN algorithm. Therefore, eliminating some RPs before applying the positioning algorithm appears necessary.

- Furthermore, by computing proper weightings, the WKNN algorithm can provide improved accuracy. However, WKNN with a fixed number of RPs cannot always achieve the required accuracy, for the same reason as simple KNN.

- EWKNN introduce simple filtering procedures in order to select the optimal number of RPs dependently from the situation, improving the position accuracy of the WPS.
The EWKNN algorithm

The device receives RSSIs from the APs, compares them with the RPs fingerprints and calculate $D_i$ for each RP$i$

$$D_i = \sum_{j=1}^{N} |A_j - R_{i,j}|, \quad i = 1, 2, 3,...,L \quad (1)$$

where $A_j$ is the RSSI from the $j^{th}$ AP, $R_{i,j}$ is the RSSI of the $j^{th}$ AP at the $i^{th}$ RP stored in the database. $N$ is the number of APs, $L$ is the number of RPs.

1. I RPs filtering: after sorting the $D_i$ in ascending order ($D_1$ will be the minimum), and denoting with $R_T$ a properly chosen threshold, RPs whose $D_i$ is larger than the threshold are removed from the list of useful RPs.

2. II RPs filtering: let be $G$ the number of remaining RPs after the I filtering and $S_i$ the difference between $D_1$ and the remaining $D_i$; after evaluating $E(S)$ as in (2), the RPs having a larger $S_i$ than $E(S)$ are removed from the list.

$$E(S) = \frac{S_2 + S_3 + \cdots + S_G}{G-1} \quad (2)$$

The position of the device is estimated as shown in (3), where $L(RP_j)$ denotes the location of RP$j$.

$$P = \frac{\frac{1}{D_1}L(RP_1)+\frac{1}{D_2}L(RP_2)+\cdots+\frac{1}{D_K}L(RP_K)}{\frac{1}{D_1}+\frac{1}{D_2}+\cdots+\frac{1}{D_K}}, \quad (3)$$
EWKNN: experimental results

Compressive Sensing (CS) provides a framework for recovering sparse signals with far fewer noisy measurements than that needed by the Nyquist sampling theorem. The sparse signal can be reconstructed exactly with high probability by solving an $l_1$-minimization problem.

The localization problem can be modeled as a sparse problem since at each time instant the user is located at a specific point in space.

The system consists of two phases:

- **Offline phase**
  - Creation of the fingerprinting database on a grid of reference points (RPs).
  - Fingerprints are then decomposed into multiple clusters using the affinity propagation algorithm.

- **Online phase**
  - Coarse localizer to find the right cluster.
  - Fine location estimation using CS.
During the offline phase, the time samples of RSS readings are collected at RPs, by pointing the mobile device to different orientations. Then, the average of the RSS time samples is computed and stored in a database.

The collected RPs are then divided into a number of clusters. Since the database at different orientations has a different set of RSS readings, the clustering is performed for each orientation. The affinity propagation algorithm is used to generate the clusters, as it does not require initialization of exemplars in the traditional K-means clustering algorithm.

The first step of the online phase is the coarse localization, having the main goal to reduce the region of interest from the whole database to a subset of it, reducing the computational complexity of the fine localization stage, as fewer RPs are considered. Furthermore, it confines the maximum localization error to the size of this subset, whereas this error can be much larger when no coarse localization is implemented.

Different coarse localization metrics are investigated to reduce the maximum error of the positioning system. In the fine localization stage, AP selection schemes are studied to further improve the accuracy of the estimation.
CS-based: experimental results

TABLE 2
Position Error Statistics

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean[m]</th>
<th>90th[m]</th>
<th>Max[m]</th>
<th>Var[m²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>kNN(κ = 3)</td>
<td>1.8</td>
<td>3.7</td>
<td>7.9</td>
<td>2.81</td>
</tr>
<tr>
<td>Kernel-based</td>
<td>1.6</td>
<td>3.6</td>
<td>7.1</td>
<td>2.28</td>
</tr>
<tr>
<td>CS-based</td>
<td>1.5</td>
<td>2.7</td>
<td>6.2</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Inertial sensors, maps, mobility models

- The past few years have seen mobile phones being equipped with inertial sensors (such as accelerometers and gyroscopes) and a magnetometer.
- Together, these sensors can provide useful information about the motion of the user that can augment WiFi based positioning.
- If the device is held in a steady position, the sensors can reliably detect user motion and improve position estimates. Addition of map constraints may help alleviate some of the fundamental limitations associated with a sensor-based tracking.
Inertial Sensors

- Sensors such as accelerometer, gyroscope and magnetometer can be used to determine device relative motion. For example, in the absence of motion, the accelerometer reading is constant (corresponding to gravity).

- However, the use of this kind of sensors is different from that in inertial navigation systems (INS) on planes. These sensors are of much lower quality and techniques such as double integration of acceleration data to get relative motion will be inaccurate.

- An alternate approach is to derive pedestrian odometry from two components:
  1. **Distance estimate** obtained from step count. The step count can be obtained from the accelerometer and combined with a step length model to estimate relative distance traveled.
  2. **Heading estimate** obtained by combining a 3D orientation estimate from the accelerometer, gyroscope and compass.

  **Sensor odometry alone is generally insufficient for positioning**
Maps

• Map constraints provide additional information on the position.
• The following information can be derived from a map:
  – Certain areas of the map are not feasible for the location (for example, the area within a large solid column or wall).
  – Certain paths on the map are not possible for the (for example, movement across walls).
  – Some parts of the map may be more likely to be frequented by users (corridor areas).
  – Furthermore, the map can be used to predict a better model of WiFi signals, mitigating the need for fingerprinting.
  – Map constraints can also be used correct slow accumulating errors and drifts in sensor estimates.
Particle Filtering

- It is important to incorporate all available information pertaining to the location of the device.
- Given its representative flexibility, particle filter is more capable of incorporating various available sources of information, compared to various derivatives of Kalman filters.

Algorithm 1: Particle filter with \( m \) particles

1. Select a collection of \( m \) particles to represent the initial cloud, from a priori distribution.
2. For each estimation interval:
   3. For each particle:
      4. Propagate the particle using the mobility model and other indirect information (mobility model using sensors, map, etc.)
      5. Calculate an importance weight associated with particle from the direct measurements (WiFi measurements, GNSS, etc.)
   6. End for
   7. Resample \( m \) times from the propagated particle pool, according to the importance weights.
   8. Report the mean or centroid of the particle cloud as the position estimate.
   9. End for
WiFi + Inertial Sensors: experimental results

Performance Comparison: a) WiFi + Maps and b) WiFi + Maps + Sensors

Cumulative distribution of positioning accuracy in a typical office building with 14 access points

Conclusions

- Fingerprinting technique is an effective method for implementing an accurate WPS but can be expensive and time consuming.
- If APs location is known, a good model of WiFi signal propagation mitigates the need of fingerprinting.
- As both EWKNN and CS-based algorithms demonstrate, it is possible to improve position estimation accuracy by adaptively adjusting parameters and measurements useful for the estimate (RPs and APs selection, weightings). This leads also to address complexity and cost reduction.
- A scalable and accurate indoor positioning system can be also achieved by combining (Kalman or particle filters) complementary information from WiFi access points, sensors on the mobile device and building maps, to estimate position.