Indoor localization system

With application to FOG episodes in Parkinsonians

Florenc Demrozi [1]
Vladislav Bragoi [1]
Federico Tramarin [2]
Graziano Pravadelli [1]

[1]Univeristy of Verona

[2]University of Padova

Table of contents

Introduction

Introduction

People affected by the Parkinson's disease are often subject to episodes of Freezing of Gait (FoG) near specific areas within their environment.

We present a low–cost indoor localization system specifically designed to identify these critical areas, which triggers the generation of a rhythmic stimuli (e.g. through a wearable device) anticipating or preventing the FoG episode.

Received Signal Strength Indicator (RSSI), is a particular measurement, used often in modern Radio Frequency (RF)–based communication systems, that is related to the power perceived by a receiver.

Received Signal Strength Indicator (RSSI), is a particular measurement, used often in modern Radio Frequency (RF)–based communication systems, that is related to the power perceived by a receiver.

Fingerprinting, is the state-of-the-art technique for indoor localization, which consist in dividing the available area in Reference Points (RPs) and carrying out a matrix of RSSI values recorded from the center of each RP.

Preliminaries

Received Signal Strength Indicator (RSSI), is a particular measurement, used often in modern Radio Frequency (RF)–based communication systems, that is related to the power perceived by a receiver.

Fingerprinting, is the state-of-the-art technique for indoor localization, which consist in dividing the available area in Reference Points (RPs) and carrying out a matrix of RSSI values recorded from the center of each RP.

Classification, is a two–step process, in the context of machine learning, which consists of a *learning phase* and a *classification phase*.

Related Works

One of the first fingerprinting systems for indoor localization was based on triangulation of WiFi signal strength and propagation model, that have an accuracy of roughly $2 \div 3$ meters.

Other solutions, based on combining sensor fusion with fingerprinting (e.g. WiFi/Bluetooth signals), require significant computational power and present problems related to the signal nature and the movement pattern.

An example of this second approach is InDoorAtlas, that is designed to work on mobile devices by exploiting RSSI measurement from *WiFi* and *Bluetooth* radio signal emitters, which are then combined with data coming from *magnetometer*, *accelerometer* and *gyroscope* sensors. Furthermore, during the learning phase, it makes also use of the *GPS*.

Localization system

System description

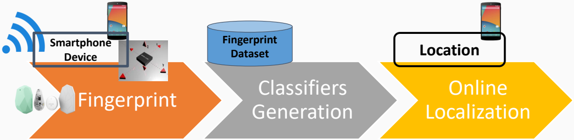

Figure 1: Localization system description

Fingerprinting

Fingerprinting

The first goal is to create a dataset of RSSI measurements and MV values, based on the following three steps:

Fingerprinting

The first goal is to create a dataset of RSSI measurements and MV values, based on the following three steps:

- subdivide the environment in RPs, mapping all risky areas into a set of points (not necessarily contiguous)

Fingerprinting

The first goal is to create a dataset of RSSI measurements and MV values, based on the following three steps:

- subdivide the environment in RPs, mapping all risky areas into a set of points (not necessarily contiguous)
- set RP information (x, y position and the existence of borders in any cardinal direction)

Fingerprinting

The first goal is to create a dataset of RSSI measurements and MV values, based on the following three steps:

- subdivide the environment in RPs, mapping all risky areas into a set of points (not necessarily contiguous)
- set RP information (x, y position and the existence of borders in any cardinal direction)
- collect data

Fingerprinting algorithm

Algorithm 1 Data collection and fingerprinting dataset creation

Output: fingerprinting dataset f

```
f = ∅
for RP center c_xy do
    for direction d ∈ {N, S, E, W} do
        for timestamp t ∈ [1 ... 30] seconds do
            rssi = collect_RSSI_Measurement(c_xy, d, t)
            mv = collect_MV_Values(c_xy, d, t)
            f = f ∪ add(c_xy, d, t, rssi, mv)
        end for
    end for
end for
return f
```

Classifier Generation

Classifier Generation

The classifier generation is characterized by a sequence of operations:

Classifier Generation

The classifier generation is characterized by a sequence of operations:

- **Data normalization**
  RSSI and the MV datasets are normalized through a classical min-max feature scaling technique, in order to make the training less sensitive to the scale of features.

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

where x is the original value and x' is the normalized value.

- Dataset enrichment
  The original fingerprinting dataset is enriched as follows:

[1] k-Means clustering algorithm

Classifier Generation

- Dataset enrichment

  The original fingerprinting dataset is enriched as follows:

    - given the whole set of RSSI values, k-Means [1] groups them in k clusters, forgetting that the set of values $RSSI_{[1...n]}$ corresponding to timestamp t, were measured all together when the smartphone was in the same (x, y) position at time t;

[1] k-Means clustering algorithm

Classifier Generation

- **Dataset enrichment**

  The original fingerprinting dataset is enriched as follows:

  - given the whole set of RSSI values, k-Means [1] groups them in k clusters, forgetting that the set of values $RSSI_{[1...n]}$ corresponding to timestamp t, were measured all together when the smartphone was in the same (x, y) position at time t;
  - extract the minimum and maximum RSSI values, min_p and max_p , for each cluster p;

[1] k-Means clustering algorithm

Classifier Generation

- **Dataset enrichment**

  The original fingerprinting dataset is enriched as follows:

  - given the whole set of RSSI values, k-Means [1] groups them in k clusters, forgetting that the set of values $RSSI_{[1...n]}$ corresponding to timestamp t, were measured all together when the smartphone was in the same (x, y) position at time t;
  - extract the minimum and maximum RSSI values, min_p and max_p , for each cluster p;
  - associate each value $RSSI_i$, with $i \in [1...n]$, per each timestamp t, to one of the k clusters according to the following rule: if $min_p \leq RSSI_i \leq max_p$, then $RSSI_i$ is associated to cluster p.

[1] k-Means clustering algorithm

- Classifier set up

  The proposed localization systems is composed on five classifiers, all of them based on the k-NN[2] machine learning algorithm:

[2] k-nearest neighbors algorithm

Classifier Generation

- Classifier set up

  The proposed localization systems is composed on five classifiers, all of them based on the k-NN[2] machine learning algorithm:

  - C1: its role is filtering the RPs, associating RSSI and MV (magnetic vector related to the Magnetic Field) values with one specific target class, so it filters the known RPs by restricting the possible location of the person among the RPs belonging to the recognized class;

[2] k-nearest neighbors algorithm

Classifier Generation (Classifier set up)

- C2: which represents the raw RSSI vectors;

Classifier Generation (Classifier set up)

- C2: which represents the raw RSSI vectors;
- C3: representing the new categorized RSSI vectors;

Classifier Generation (Classifier set up)

- C2: which represents the raw RSSI vectors;
- C3: representing the new categorized RSSI vectors;
- C4: raw MV values;

Classifier Generation (Classifier set up)

- C2: which represents the raw RSSI vectors;
- C3: representing the new categorized RSSI vectors;
- C4: raw MV values;
- C5: the categorized MV values obtained from the previous classifier.

Classifier Generation

- Probabilistic graph model
  Purpose: increase the localization accuracy and preserve the environment knowledge.

Classifier Generation (Graph model)

Consider a graph $M = (V, E, P, B, T)$, where:

Classifier Generation (Graph model)

Consider a graph $\mathsf{M} = (\mathsf{V}, \mathsf{E}, \mathsf{P}, \mathsf{B}, \mathsf{T})$, where:

- $\mathsf{V} \in \{\mathsf{V_r} \cup \mathsf{V_{ar}}\}$ is the set of RPs composed of:
  - V_r: RPs identifying risky areas as for example doorways;
  - V_{ar}: identifying RPs adjacent to risky areas;

Classifier Generation (Graph model)

Consider a graph $M = (V, E, P, B, T)$, where:

- $V \in \{V_r \cup V_{ar}\}$ is the set of RPs composed of:
  - V_r: RPs identifying risky areas as for example doorways;
  - V_{ar}: identifying RPs adjacent to risky areas;
- $E = \{(v_i, v_j) | v_i, v_j \in V \wedge v_i, v_j \textit{ are neighbors}\}$ is the set of edges between RPs;

Classifier Generation (Graph model)

Consider a graph $\mathsf{M} = (\mathsf{V}, \mathsf{E}, \mathsf{P}, \mathsf{B}, \mathsf{T})$, where:

- $\mathsf{V} \in \{\mathsf{V_r} \cup \mathsf{V_{ar}}\}$ is the set of RPs composed of:
    - V_r: RPs identifying risky areas as for example doorways;
    - V_{ar}: identifying RPs adjacent to risky areas;
- $\mathsf{E} = \{(\mathsf{v_i}, \mathsf{v_j}) | \mathsf{v_i}, \mathsf{v_j} \in \mathsf{V} \land \mathsf{v_i}, \mathsf{v_j} \textit{ are neighbors}\}$ is the set of edges between RPs;
- $\mathsf{P} : \mathsf{E} \rightarrow [0, 1]$ defines, for each edge $e \in E$, the probability of crossing e;

Classifier Generation (Graph model)

Consider a graph $\mathsf{M} = (\mathsf{V}, \mathsf{E}, \mathsf{P}, \mathsf{B}, \mathsf{T})$, where:

- $\mathsf{V} \in \{\mathsf{V_r} \cup \mathsf{V_{ar}}\}$ is the set of RPs composed of:
  - V_r: RPs identifying risky areas as for example doorways;
  - V_{ar}: identifying RPs adjacent to risky areas;
- $\mathsf{E} = \{(\mathsf{v_i}, \mathsf{v_j}) | \mathsf{v_i}, \mathsf{v_j} \in \mathsf{V} \wedge \mathsf{v_i}, \mathsf{v_j}$ *are neighbors*$\}$ is the set of edges between RPs;
- $\mathsf{P} : \mathsf{E} \rightarrow [0, 1]$ defines, for each edge $e \in E$, the probability of crossing e;
- $\mathsf{B} : \mathsf{V} \rightarrow \mathcal{P}(\{\mathsf{N}, \mathsf{S}, \mathsf{W}, \mathsf{E}\})$ associates each RP to its set of adjacent borders (e.g., walls) represented as cardinal points.

Classifier Generation (Graph model)

Consider a graph $\mathsf{M} = (\mathsf{V}, \mathsf{E}, \mathsf{P}, \mathsf{B}, \mathsf{T})$, where:

- $\mathsf{V} \in \{\mathsf{V_r} \cup \mathsf{V_{ar}}\}$ is the set of RPs composed of:
  - V_r: RPs identifying risky areas as for example doorways;
  - V_{ar}: identifying RPs adjacent to risky areas;
- $\mathsf{E} = \{(\mathsf{v_i}, \mathsf{v_j}) | \mathsf{v_i}, \mathsf{v_j} \in \mathsf{V} \wedge \mathsf{v_i}, \mathsf{v_j}\ \textit{are neighbors}\}$ is the set of edges between RPs;
- $\mathsf{P} : \mathsf{E} \rightarrow [0, 1]$ defines, for each edge $e \in E$, the probability of crossing e;
- $\mathsf{B} : \mathsf{V} \rightarrow \mathcal{P}(\{\mathsf{N}, \mathsf{S}, \mathsf{W}, \mathsf{E}\})$ associates each RP to its set of adjacent borders (e.g., walls) represented as cardinal points.
- $\mathsf{T} : \mathsf{V} \rightarrow \mathsf{D}$, where $D =${pattern change, stairs, doorway, ...}, defines per each RP the characteristic of the environment where the RP is located.

Classifier Generation (Graph model)

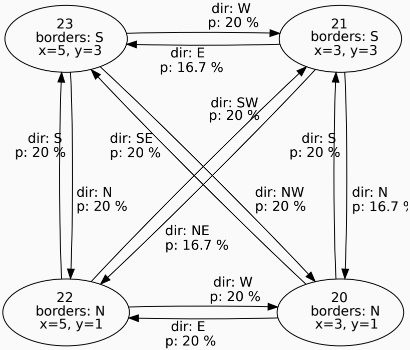

Figure 2: A graph example proposed in the previous slide

Table 1: Training data for the proposed classifiers.

RP ID	Pos. X	Pos. Y	Borders	Description	Timestamp	Direction	$RSSI_{[1...n]}$	$MV_{[xyz]}$	$CRSSI_{[1...n]}$	$CMV_{[xyz]}$	Group
1	1	1	North	doorway	t_1	North	[-72 -93 -44 ...-56]	[13 9 8]	[3 2 3 ...3]	[2 3 1]	l_1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	7	4	East	pattern	t_j	East	[-75 -88 -47 ...-59]	[13 8 7]	[3 2 3 ...2]	[2 3 2]	l_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
m	23	15	West	narrowing	t_h	West	[-45 -58 -60 ...-76]	[5 42 28]	[1 3 3 ...2]	[1 3 2]	l_k

Online Localization

Localization

Consider this example:

Localization

Consider this example:

- at time t, the person is located at RP v_i. During walking, the mobile phone perceives radio signals, computes magnetic field vectors and gets movement directions that are periodically provided to the localization system;

Localization

Consider this example:

- at time t, the person is located at RP v_i. During walking, the mobile phone perceives radio signals, computes magnetic field vectors and gets movement directions that are periodically provided to the localization system;
- at time $t + 1$, the perceived RSSI and MV values, after being normalized, are provided to the C1 classifier. C1 filters the RPs in the graph model of the environment and then, survived RPs are then concurrently considered by classifiers C2, C3, C4 and C5, which finally return at maximum 4 different candidates, one for each classifier, to predict the new position of the person.

The position is calculated as follows:

Localization

The position is calculated as follows:

1. if at least three classifiers (C2, C3, C4 and C5) agree on the same RP v_j, the person is located at v_j. Then, the probabilistic graph model is used only to decide if generating the rhythmic stimuli or not.

Localization

The position is calculated as follows:

1. if at least three classifiers (C2, C3, C4 and C5) agree on the same RP v_j, the person is located at v_j. Then, the probabilistic graph model is used only to decide if generating the rhythmic stimuli or not.
2. when there is no agreement among the classifiers, the system computes the shortest paths from v_i to each of the RP returned by classifiers C2, C3, C4 and C5. The RP belonging to the highest probability path is finally selected as new location of the person.

Localization

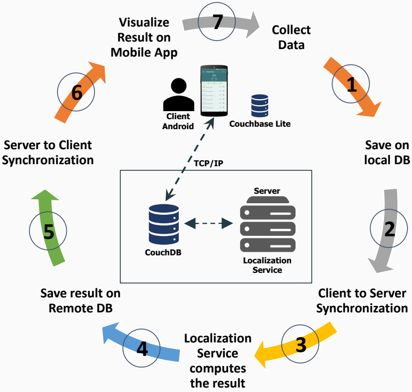

Figure 3: A simplified flow graph of the localization process

Conclusion

Conclusion

Experimental campaigns, carried out in three different private environments, with different types of smartphones and different users, have shown that this approach was able to detect users in the vicinity of a risky area with an absolute accuracy of 1.5 meters.

Conclusion

A comparison with *InDoorAtlas* shows that this approach provide better minimum absolute accuracy (1 meter), but it is very sensitive to particular movement. This resulted in a higher number of false positives, which might become very disturbing.

Energy consumption: tests revealed that our approach present a battery consumption of an *average 12% of total battery charge* for one hour of usage on the used smartphones, compared with a battery consumption of an *average of 15%* for *InDoorAtlas*.

Furthermore, with our approach the localization can be executed directly on the smartphone without the risk of running out of memory, and avoiding any communication latency.

Conclusion

The table shows the comparison between the two approaches, in terms of accuracy:

Environment	Number RPs	Our Approach (min,max)	InDoorAtlas (min,max)
Indoor (H_1)	14	$1.5m, 1.5m$	$1m, 8m$
Indoor (H_2)	12	$1.5m, 1.5m$	$1m, 8m$
Indoor (H_3)	17	$1.5m, 1.5m$	$1m, 8m$

Table 2: Accuracy of the proposed localization approach.

Thank you!

Indoor localization system

What about tests?

The figure shows the software architecture structure, pointing out that the system is composed by the following parts:

- Android application
- CouchDB (remote) database
- Localization Service

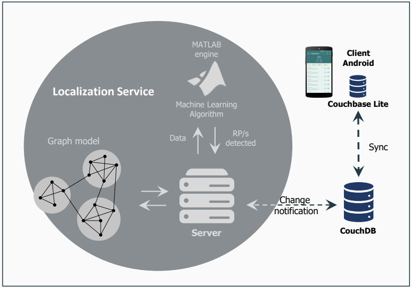

Figure 4: Software architecture