Indoor Localisation, Localisation Algorithm

Sebastian Hojas, 014570704

December 2, 2015

1 Instructions

The program that computes the location is a Python (2.7) script and takes the following arguments:

```
Usage: python2.7 Localiser.py <input_file> [<database>]
       Input file is required, database optional.
```

2 Test results

```
python2.7 Localiser.py test.fingerprints train.fingerprints

>  x= 2  y= 2089  z= 474
```

3 Algorithm

1. Find n best matches

   Missing matches are penalised with a small factor

   On the other hand, intersecting coverage is being rewarded exponentially
2. Calculate average point based on all matches
3. Remove n/2 points furthest away from the average point (medianisation)
4. Calculate weighted average of points

I have been testing different values of n and went with $n = 10$ in the end.

4 Database format

I wanted to formly specifiy the fingerprints in *scans/db.fingerprints*. Each line represents one fingerprint and has the following format:

```
x;y;z;[MAC;RSSI]
  [] .. represents revision (no brackets used in the actual file)
```

5 Taking fingerprints

I have been taking fingerprints dynamically by walking from point to point and distributing it by timestamp. I have collected roughly 240 prints. One problem I have encountered is that my scans on OSX vary significantly. The closest matches are often somewhere completely different. I could not yet validate whether my algorithm supports this behaviour or the measurements vary so significantly. I have tried to work around this by removing potential matches far away from the average point in the matching algorithm.

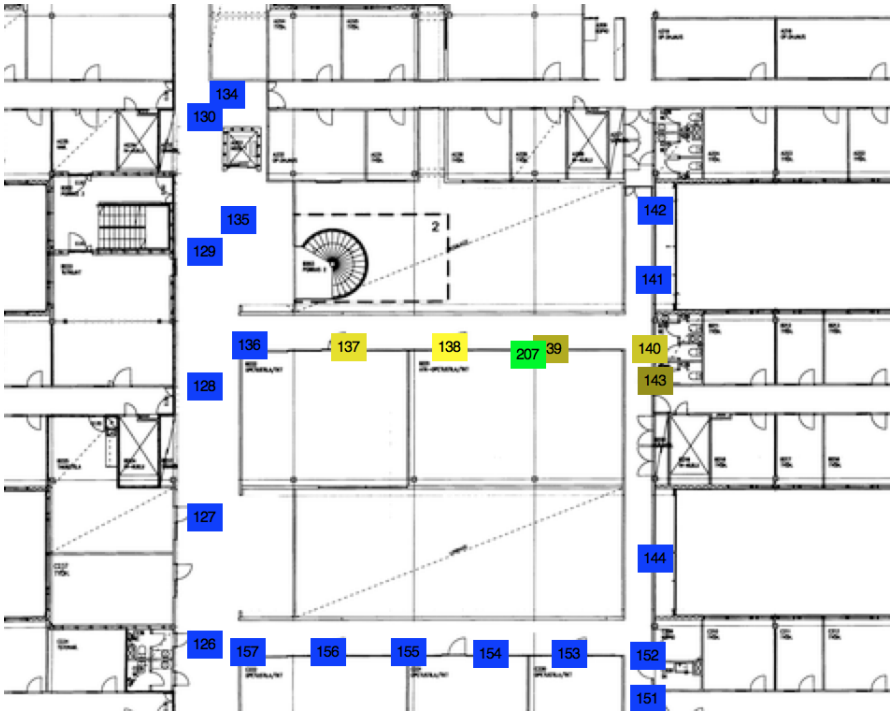

Figure 1: Prototype visualising measurements and localisation result. The green points represent the estiamted location. Blue points represent the non-matched fingerprints. Yellow-shaded fingerprints represent matched fingerprints (bright yellow = strong matching, dark yellow = weak matching)