IN4391 - Smart Phone Sensing
Final Report

April 04, 2018
Group - CCS, Phone - OnePlus 5, Android Version - 8.0.0, API - 26
MATLAB, org.apache.commons.math3.analysis.function.Gaussian
https://github.com/BSJAIN92/SPS, https://github.com/BSJAIN92/Example6-SPS

Bhavya Jain (4745507)
Delft University of Technology
Delft, Netherlands
B.Jain@student.tudelft.nl

Konrad Kleeberger (4748476)
Delft University of Technology
Delft, Netherlands
K.kleeberger@student.tudelft.nl

ABSTRACT

1 BAYESIAN LOCALIZATION

1.1 Data Collection

- In each iteration, we collected 10 samples with refresh time of 5s (50s/iteration)
- The recording was always done in all directions
- Recording was done on a quiet Friday afternoon (March 9) and a busy Wednesday afternoon (March 14). More data was recorded to increase sample size on April 4.

1.2 Data Processing

- The collected data was filtered for stable SSIDs, mainly those which are not expected to change during the course of the project.
  - Observed SSIDs = 15+, Selected SSIDs = 4
  - SSIDs Selected = "eduroam", "tudelft-dastud", "TUvisitor", "TRIBLER"
- In collected data, it was observed that many BSSIDs were observed very few times in a particular cell. To clean the collected data, BSSIDs were filtered for BSSIDs observed in at least 80% samples. 7% to 13% of observed BSSIDs were selected
- The data has been further processed via Matlab to make the gaussian fit.
  - Used "fitdist" function with the RSS values of a BSSID per cell as input and the parameter "normal" for normal distribution
  - For calculating the y value of a normal distribution we use the gaussian function of the library math3 of apache commons.

1.3 Radio Map

Figure 1 shows one BSSID which has been scanned in four cells. The graph for C14 and C15 are overlapping as well as for C12 and C13. Both of these pairs of cells are neighbors and hence, it's logical for them to have similar map. C12 and C15 are on the opposite site of the corridor and are quite different, so they hardly have any overlap in the normal distribution.

Figure 2 shows again that C12 and C15 are quite different and nearly have no overlapping of values. Interestingly, the adjacent cell pairs (12, 13 and 14, 15) have similar mu values, with much different sigma values. In this case, even with similar mu values, the cell with narrow distribution will always have higher probability, and will have more chance to win. This shows the variety of data that was observed and used for training.

1.4 Evaluation

Evaluation was done on a Thursday morning (March 15). We used parallel and serial methods to determine the probability and filtered data for standard deviation values for less than 5, 4 and 3. Most accurate results were obtained using parallel method for standard deviation values less than 5. The following tables represent test results for parallel method to determine the probability with BSSIDs filtered for sigma values less than 3 and less than 5 respectively.

We decided to move ahead with parallel method to determine the probability and filtered the data to include BSSIDs with sigma less than 5.

1.5 Discussion

The main difficulty was encountered while determining suitable method for localization. Serial and parallel methods both had their merits. Hence, we implemented both methods and tested the application based on both. In order to account for complexity of localization for adjacent cells, the application was initially tested for cells 12, 13, 14 and 15.

Since cells 13 and 14 are adjacent and with some Access Points in the hallway to East side rooms, testing for these cells provides a good validation of the method used. Another difficulty was encountered in data collection. This was due to high difference in heights of team members. Our initial recorded data was much more scattered due to data recording at different heights. We decided a point for recording data for each person to arrive at same height. Also, during application testing, different method of holding the phone yielded different results.

Wi-fi is scanned after a certain period of time. In order to obtain correct test results, "Locate" button is disabled after every localization and is enabled after 5s. Also, we observed impact of including BSSIDs with various sigma values to understand their impact. We tested the localization accuracy using BSSIDs with different sigma values (<3, <4, <5). Collected data was filtered and cleaned not only

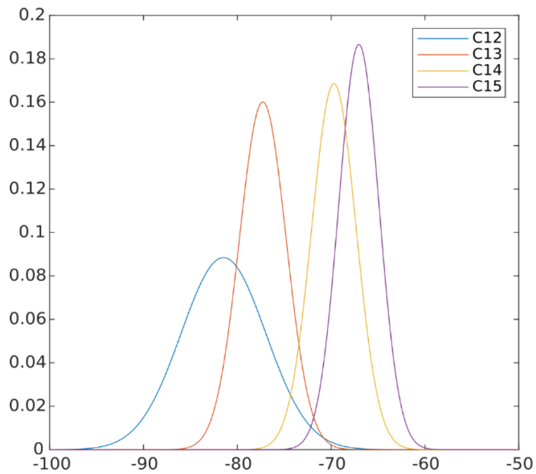

Figure 1: SSID with high difference for each cell

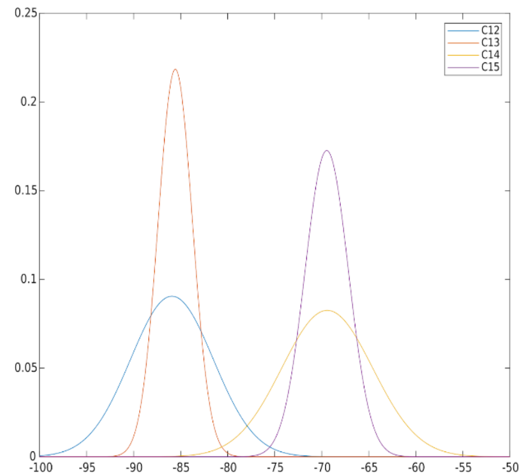

Figure 2: SSID with less difference for adjacent cell

Sigma < 3	Cell 15	Cell 14	Cell 13	Cell 12
Cell 15	10			
Cell 14	5	4	1	
Cell 13			10	
Cell 12			10	0

Sigma < 5	Cell 15	Cell 14	Cell 13	Cell 12
Cell 15	9	1		
Cell 14	1	9		
Cell 13			10	
Cell 12	1		2	7

Actual Location
Observed Location

Figure 3: Bayesian Localization Evaluation

using SSIDs, but also for BSSIDs that were observed in at least 80% of the samples collected.

The application was later tested for most cells using the same approach and the overall accuracy achieved was of ~85%. In the new version of application which has particle filter implemented as well, the application attempts localization using Bayesian after every 5s (Time taken by phone to scan wifi signal again) if a step has been taken. If no step has been taken, localization can still be performed using the locate button.

2 PARTICLE FILTERS LOCALIZATION

2.1 Motion and Step Detection

We used a hybrid approach for motion and step detection. It was observed that in some phones the step detection and step counter sensors were fast enough to start (~3 steps), while in some other phones it took a longer duration of movements (~10 steps) to detect the activity.

In the hybrid model, we detect the current activity, and once it is detected that the person is walking, we start to update particles location after every step. However, once the step detection sensor starts, we stop using our motion detection to improve accuracy and update particles on every step detected.

For our motion detection model, we are using the Accelerometer sensor. Using this, we get the acceleration values for individual axis. We calculate the magnitude of acceleration using the values for all axis. The values are recorded for a time window of 650 ms. This time window was decided after careful experimentation using long and short walks, and by measuring the time and steps taken during the walk. Once we have the magnitudes for the whole time window, we take their max, mean and standard deviation. We then subtract the mean from max to account for deviation due to gravity.

This updated magnitude is then checked to present in a specified range of 2.5 and 5. If yes, then a step is said to be taken. This value for magnitudes is determined after recording walking data and determining the most probable range of updated magnitude while walking.

2.2 Motion Model

(1) the yaw angle of the rotation matrix is used via rotation vector
(2) uniform distribution of the rotation and stride distance
(3) rotation offset of 280 degrees to adjust the north of the map to the north of the compass

(4) rotation variance is 30 degrees
(5) distance variance is 10 percent
(6) variance always in both directions
(7) stride length is calculated by height multiplied by a factor of 0.4x depending of the sex. For male we used the factor of 0.415[1]
(8) the particles are updated with every step

2.3 Floor Map and Particles

For floor map and particles, we used the Canvas and Bitmap in Android. The canvas was is scaled by using the width of current screen size and dividing it by 2700. The reasoning behind selecting the value 2700 is that the floor width is 2600 cm (eliminating the floor area where there would be no evaluation). we added 100 to it to make sure the walls are clearly visible on the screen. Now each 100 points in the bitmap represent 1m of actual distance.

Each individual shape is now scaled and the x and y coordinates are assigned to each room based on their length and width. Further, the walls are made 5 points thick (100 points = 1m) and each corner has overlap of walls' figures. Each wall is a rectangle shape object. The coordinates for walls were calculated in Excel and the code to initialize multiple walls was easily generated using Excel and then pasting the generated code lines in Android Studio.

We have used 5.000 particles in the application. We were easily able to support over 10.000 particles, but we reduced them to support frequent localization using Bayesian. This is now supported in almost all modern smart phones. Smart phones with better configuration (including OnePlus 5) were able to support over 50.000 particles along with Bayesian. Each particle is of radius 5 points.

Particles distribution is done based on the area of each cell. Bigger cells are given more particles in accordance with the area covered. This accounts for the fact that bigger cells have more area to move around and hence can easily support more particles. Cells with smaller area means that particles collide with walls too often, hence have to be re-sampled frequently.

Figures 4 and 5 show the complete results for 2 smart phones with huge difference in configurations and how many particles were they able to support in the application. Two different scenarios of with and without continuous Bayesian Localization were tested.

Figures 6 and 7 show the time taken by each sensor and model to start detecting the activity and the accuracy of each of them for the two smart phones.

For 3rd floor, since no cells were provided in the floor plan, we went ahead and made similar cells as floor 4 for the same area. This allowed us to distribute particles evenly across the rooms. In case of re-sampling on floor switch, particles can also be easily re-sampled in adjacent cells with this concept.

[1] http://livehealthy.chron.com/determine-stride-pedometer-height-weight-4518.html

2.4 Collision Detection

As the movement of the particles is not step by step within the thickness of the walls a normal intersection detection is not enough as particles can jump over the walls. Our second approach was to split the distance into multiple parts which are within the wall thickness to prevent the jump. This method still had some issues and the computation was way to intensive. Finally, we settled down on a very simplistic approach which connects the previous and new values of a point via a rectangle. This trick allows us to test for an intersection of the newly created rectangle and the walls. The computation only needs to check with all walls.

2.5 Re-sampling

The first approach was to resample the particles within the room with the highest probability and therefore most particles. As our initial belief spawns particles based on the area, this would lead that the resampling favours cells which already have many particles. Especially if we move in the direction of the corridors. We achieved way better accuracy when we resampled to all surviving particles. This method allows cells with fewer particles to survive longer in case they were on the right spot.

For floor switch, neither of the smart phones available with us had barometer. As a result, we had to either build a motion model for climbing or going down the stairs, or come up with something innovative.

Since we are continuously using Bayesian Localization, we figured we will use this for floor switch. The concept is that if the last cell detected by Bayesian was Cell 17 (3rd floor cell for stairs) and now it detects Cell 16 (4th floor floor cell for stairs), it means that we have now climbed up from 3rd floor to 4th floor. Similarly, if last cell detected was 16 and current cell detected is 17, it means we came down to 3rd floor. This implemented in similar way. Particles are re-sampled in the cell for stairs of new floor (16 or 17) and in adjacent cell(s).

2.6 Localization Experiments

Initial experiments for motion detection, motion model and particle re-sampling were done in Aula from 30th March to 2nd April. During this time after experiments and testing, most confidence was on functionality of motion detection, however motion model and particle filter were assumed to have a large scope of improvement when they were to be tested at the actual location.

Experiments for localization using Particle Filters, continuous Localization using Bayesian and communication between devices were conducted on 3rd and 4th April in the given floor plan and the application was improved based on the results of these experiments. Minor changes had to be made to the working of motion model and particle filter. Furthermore we switched the resampling method.

Figure 8 shows the synchronized location of other devices. Every other device is shown as a red dot in the centre of the cell. The location is determined by the particle filter. The particle filter shows inaccurate data as for demonstration purpose the corridor between

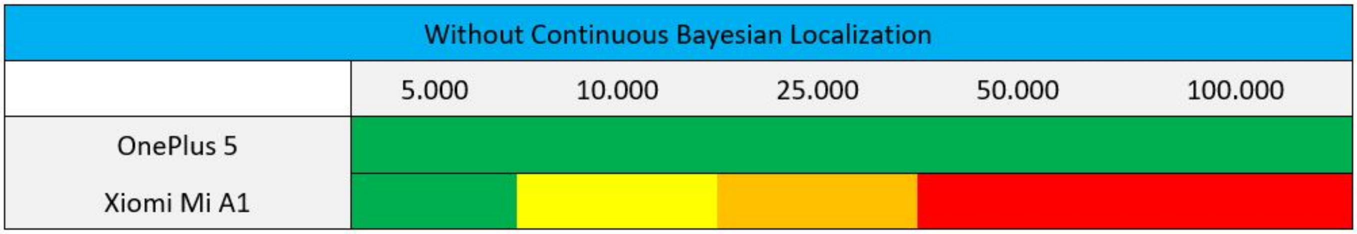

Figure 4: Particles Supported by each phone without continuous Bayesian Localization

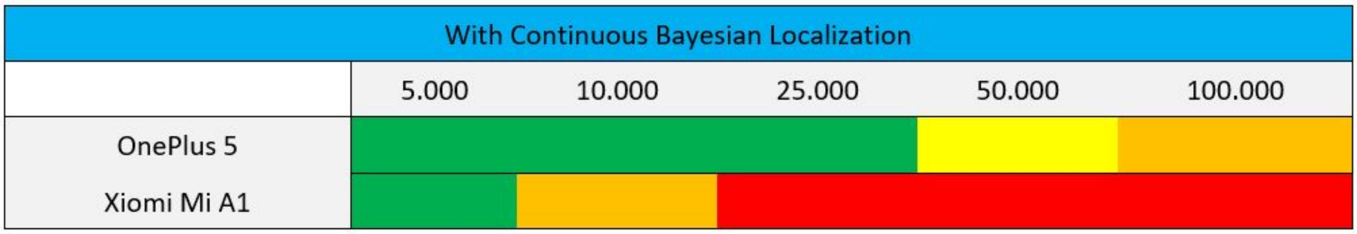

Figure 5: Particles Supported by each phone with continuous Bayesian Localization

Steps Taken to Detect Motion			
	Step Detector	Step Counter	Custom
OnePlus 5	10 Steps	10 Steps	1-2 Steps
Xiomi Mi A1	3 Steps	8-10 Steps	1-2 Steps

Figure 6: Steps Required to Start Detecting Motion

Accuracy			
	Step Detector	Step Counter	Custom
OnePlus 5	95%	95%	80-85%
Xiomi Mi A1	95%	95%	80-85%

Figure 7: Accuracy of Sensors and Models

cell two and fourteen had been used which is not supported by our map. Converging particles can be seen in Figure 9. The black line shows the walked path and both the visual image and the shown probability match with the actual position which is room four. The Bayesian method does not show any value as the screenshot is from a different device.

3 CHALLENGES AND INNOVATION

- Challenges
  - Cleaning recorded data for Bayesian Localization
  - Developing Motion Model and having offsets for noise
  - Step Counter Sensor not working good enough
  - Lack of Barometer
  - Recording Data
- Innovation
  - Continuous Bayesian Localization
  - Hybrid Motion Model (Switch between developed model and existing Step Counter/Detector)
  - Communication between smart phones about location of each other
  - Filter for SSIDs based on stable SSIDs and BSSIDs based on presence in most samples.
  - Used Excel to generate lines of code

4 INDIVIDUAL WORKLOAD

- Bhavya
  - Coding - Bayesian Data Recording
  - Data Recording - Bayesian
  - Data Cleaning - Bayesian
  - Experiments and Testing - Bayesian
  - Coding - Floor and Particles
  - Coding - Continuous Bayesian Localization
  - Report - Sections 1(1.1-1.4), 2(2.1, 2.3) ,3 ,4
  - Report - Figures
- Konrad
  - Coding - Bayesian Probability
  - Data Recording - Bayesian
  - Data Processing - Bayesian (Mu and Sigma)
  - Coding - Collision Detection
  - Coding - Communication Between Devices
  - Experiments and Testing - Particle Filter
  - Data - Particle Filter Motion Model
  - Report - Sections 1(1.5), 2(2.2, 2.4, 2.5, 2.6), 5, 6

5 POSSIBLE FUTURE DIRECTIONS

Following are the future directions we expect based on our implementation of application and novelty feature.

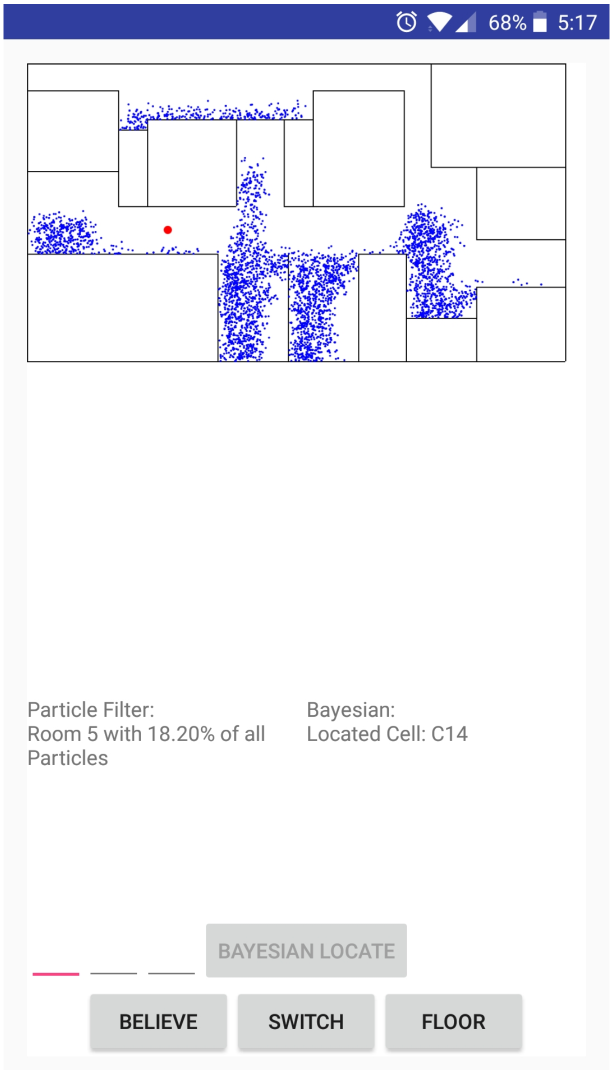

Figure 8: Screenshot of the application without converging showing the location of one other device

- More number of particles can be supported to increase accuracy.
- Based on novelty feature of communication between devices
  - Constant location Updates can be sent
  - A monitoring service could be built to keep track of every device
  - We propose it's use in building security where security personnel can receive notification about another each other if any unusual activity is detected

6 REFERENCES

- http://livehealthy.chron.com/determine-stride-pedometer-height-weight-4518.html
- https://developer.android.com/guide/topics/sensors/sensors_motion.html

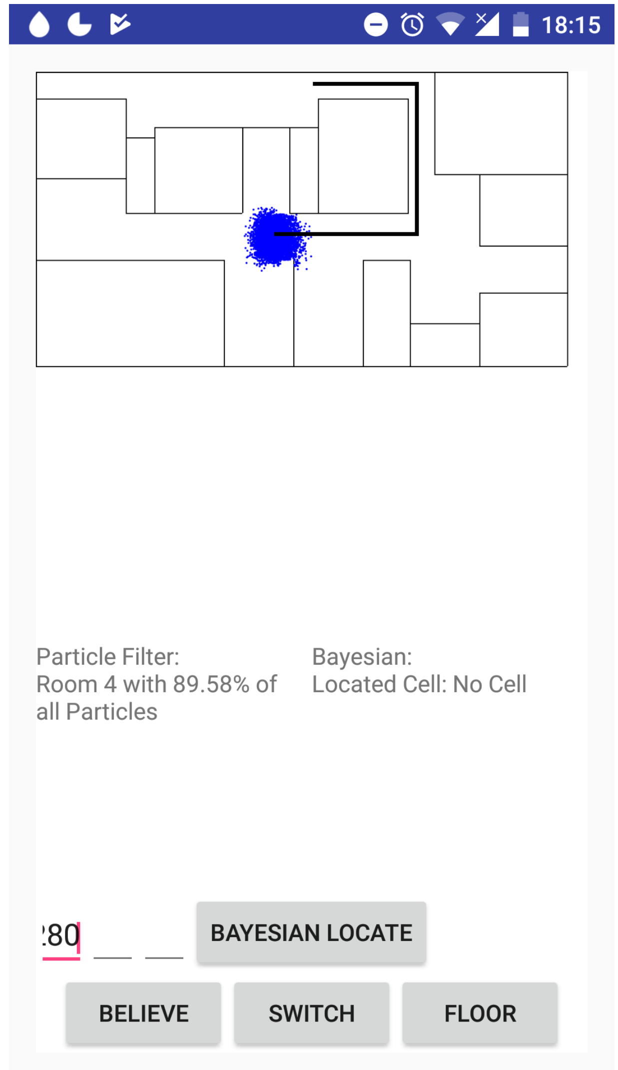

Figure 9: Screenshot of the application with a unique path and convergence

- https://github.com/SmartPhoneSensingDelft