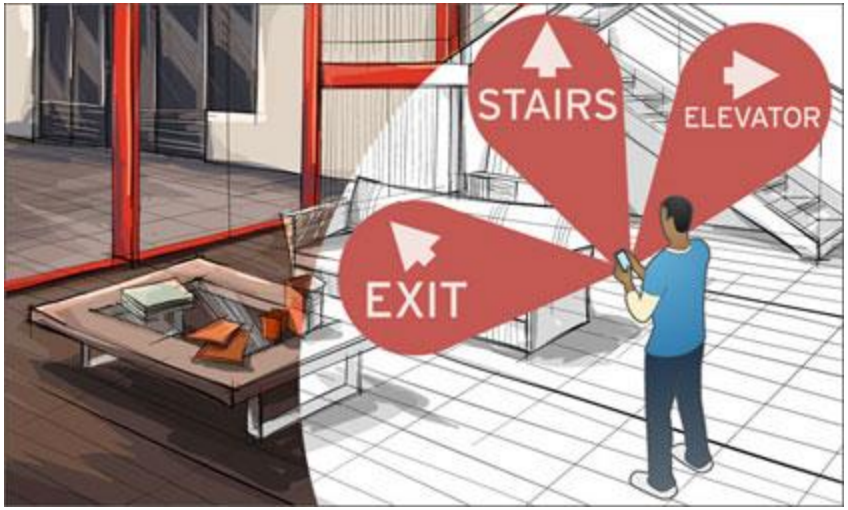
ENHANCING IBEACON BASED MICRO-LOCATION WITH PARTICLE FILTERING

Faheem Zafari- **Ioannis Papapanagiotou**
Computer & Information Technology
Purdue University
faheem0@purdue.edu
ipapapa@ncsu.edu

PURDUE UNIVERSITY

BACKGROUND

- **Indoor Localization**
  - Challenging due to inherent environment complexities.
  - Can be leveraged to provide a number of location based services.



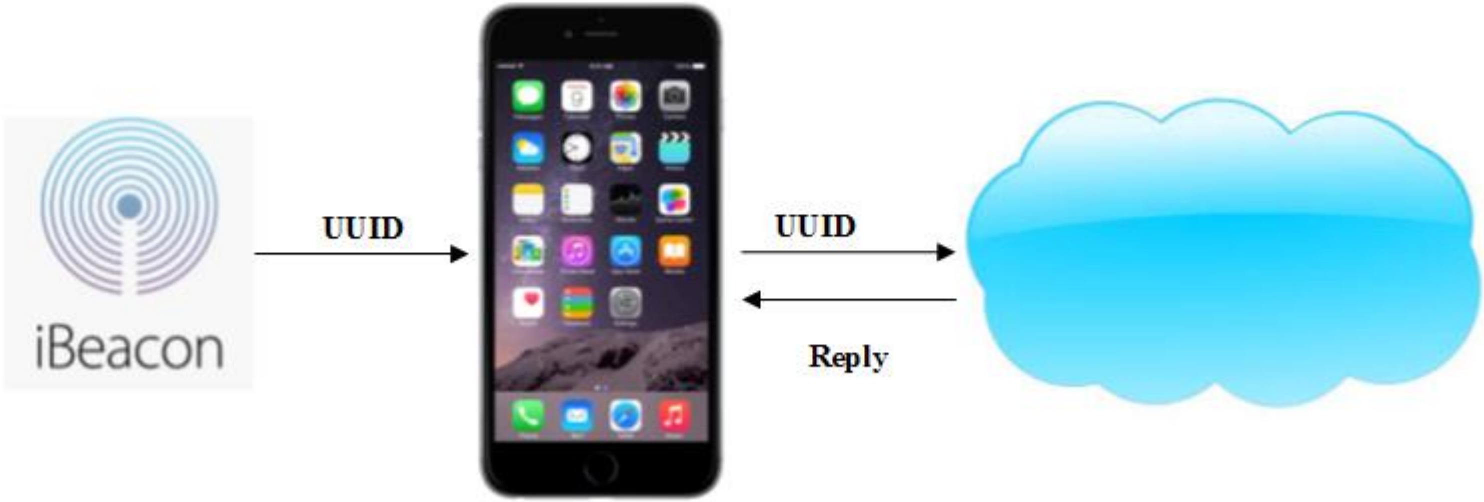
Taken from today.duke.edu

IBEACONS

- **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.



MOTIVATION

- 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 OBJECTIVES

- 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.

TRACKING PROBLEM FORMULATION

- 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.

TRACKING PROBLEM FORMULATION

OPTIMAL SOLUTION

- 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

$$p(z_i|z_{i-1}) = \int p(z_i|y_i)p(y_i|z_{i-1})dy_i$$

PURDUE UNIVERSITY

TRACKING PROBLEM FORMULATION

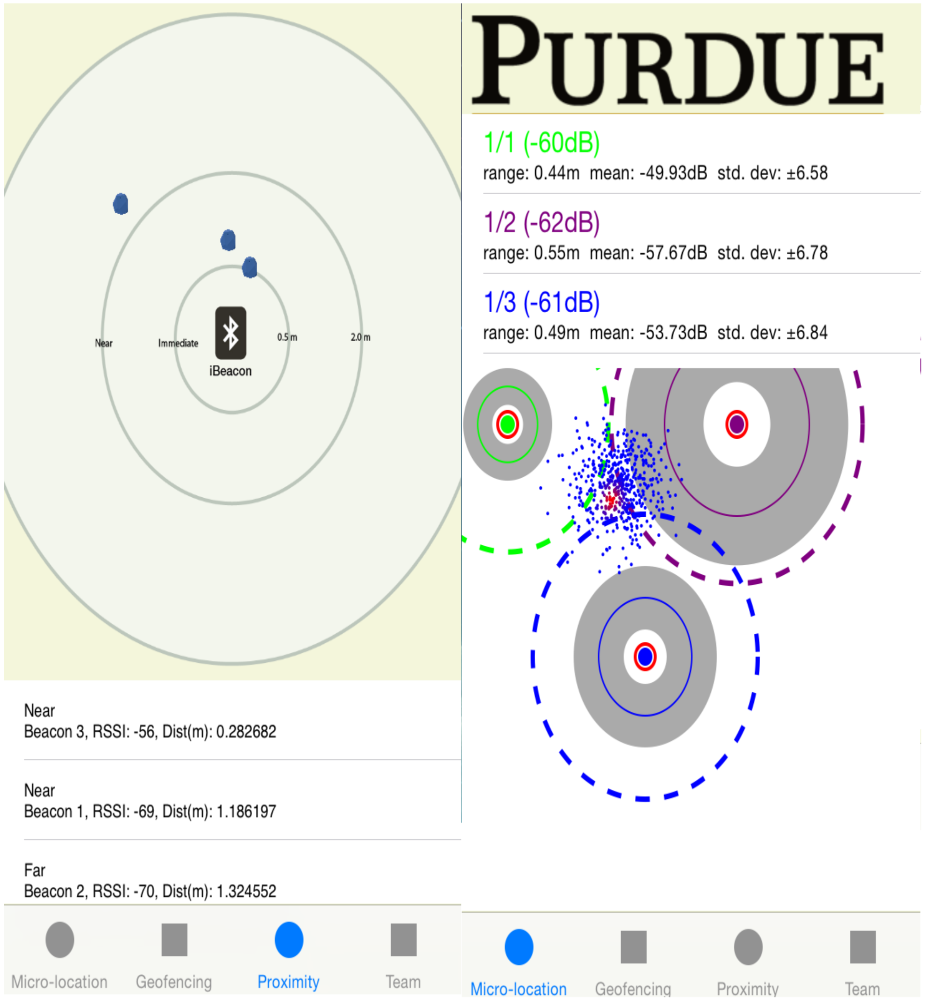
SUB-OPTIMAL SOLUTION: PARTICLE FILTERING

- 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



PURDUE
UNIVERSITY

EXPERIMENTAL SETUP

- Compare the estimated location of user with actual position

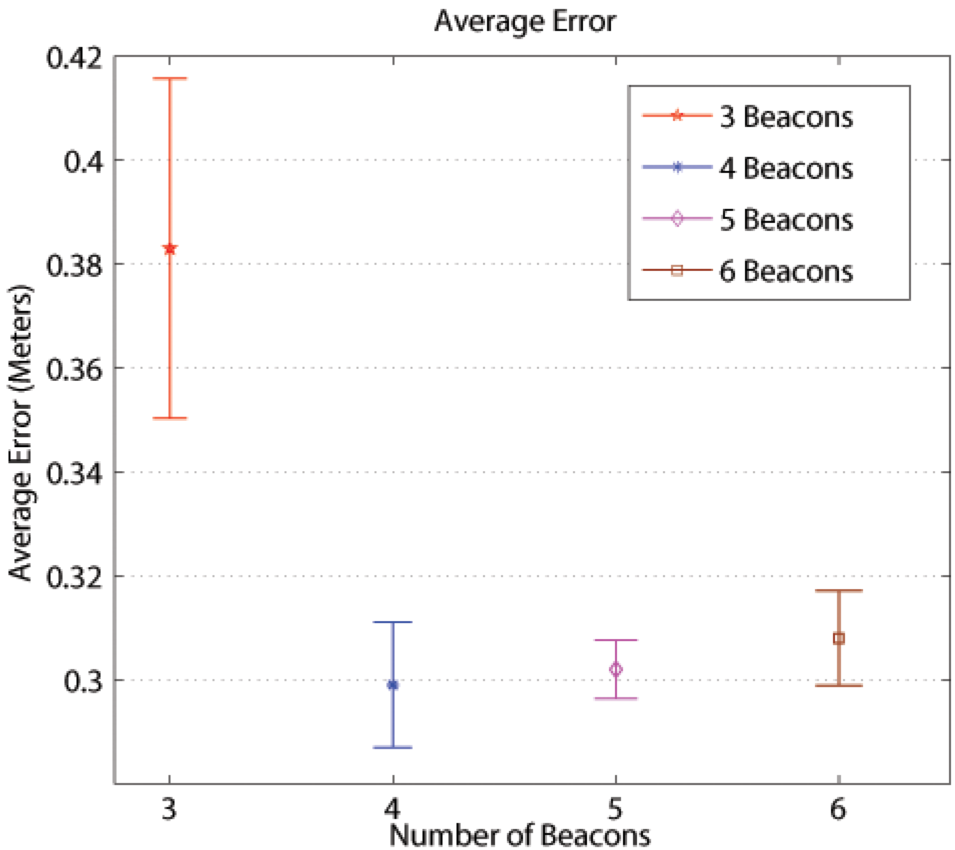
$$< Error > = \frac{\sum_{i=1}^{n} \sqrt{(X_i - X_{<est>})^2 + (Y_i - Y_{<est>})^2}}{n}$$

- Experimental parameters

Device	Apple iPhone 4s
Wireless Interface	Bluetooth V4.0/ 2.GHz
Operating System	iOS 8.1
Beacons	Gimbal Series 10
Gimbal Range	50 meters
Transmission Frequency	100ms
Major Value	Yes
Minor Value	Yes

AVERAGE ERROR (RESULTS)

- 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

AVERAGE ERROR(RESULTS)

- 1mx1m scenario: Localization error in meters

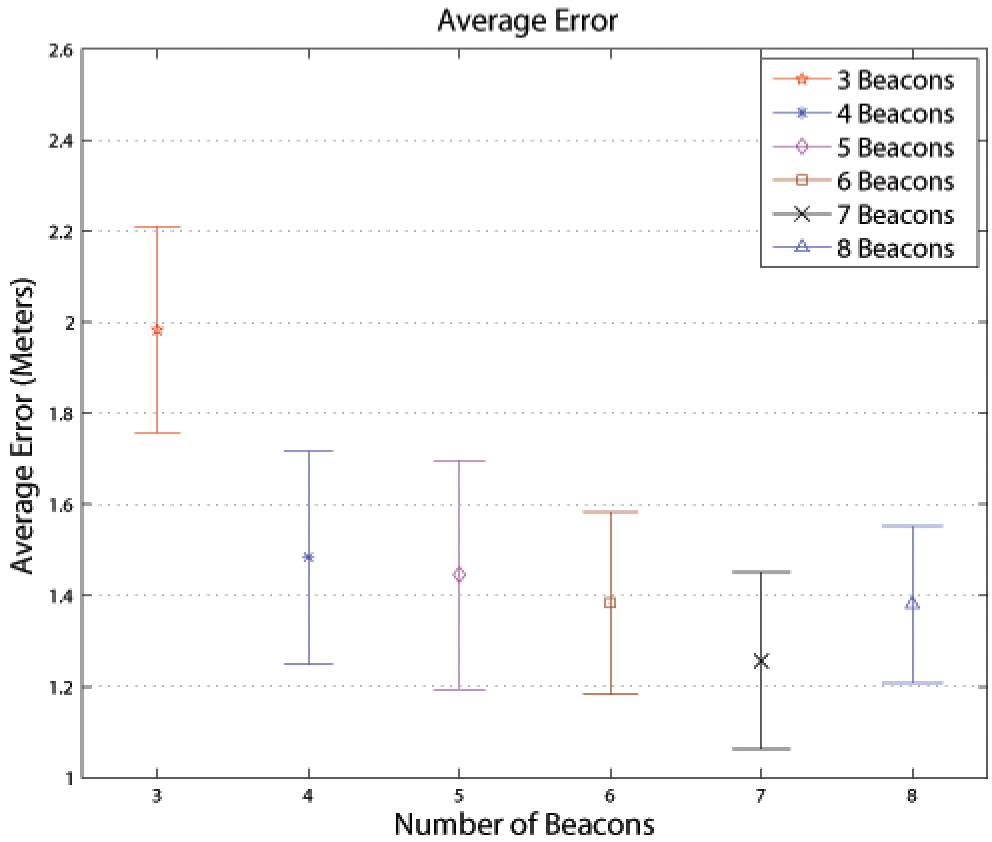
Particles	Beacons			
	3	4	5	6
400	**0.308**	0.290	0.303	0.301
600	0.356	0.308	0.312	0.302
800	0.396	0.301	0.302	0.31
1000	0.384	**0.276**	0.298	0.316
1200	0.400	0.299	**0.293**	0.318
1400	0.403	0.289	0.307	0.315
1600	0.385	0.314	0.306	0.316
1800	0.407	0.298	0.299	**0.291**
2000	0.411	0.312	0.300	0.304

- 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

PURDUE UNIVERSITY

AVERAGE ERROR(RESULTS)

- 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

PURDUE
UNIVERSITY

AVERAGE ERROR(RESULTS)

- 11mx6m scenario: Localization error in meters

Particles	Beacons					
	3	4	5	6	7	8
400	2.195	1.486	1.720	1.590	1.385	1.492
600	2.167	**1.074**	1.159	1.422	1.200	1.595
800	2.152	1.729	1.721	1.598	1.345	1.623
1000	1.736	1.802	**0.975**	1.284	1.220	1.432
1200	1.843	1.678	1.531	1.126	**1.008**	1.275
1400	2.262	1.507	1.575	1.639	1.442	**1.180**
1600	2.049	1.251	1.455	**1.149**	1.339	1.245
1800	1.774	1.368	1.540	1.208	1.362	1.168
2000	**1.668**	1.451	1.328	1.430	1.017	1.411

- 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

PURDUE UNIVERSITY

Questions?

faheem0@purdue.edu
ipapapa@ncsu.edu