UCL Department of Computer Science
CS M038/GZ06: Mobile and Cloud Computing
Spring 2013
Kyle Jamieson and Brad Karp

Zee: Zero-Effort Crowdsourcing for Indoor Location (Rai *et al.*, ACM MobiCom 2012, pdf)

Background

- *What's the goal here?* Determine the location of a mobile device by RF fingerprinting of WiFi received signal strength.
  - Uses beacons transmitted by nearby access points (APs)
  - Mobile receives beacons; radio indicates received signal strength
  - Most prior fingerprinting-based work goes in two phases:
    1. Training (calibration) phase: record a vector $(s_1,\dots,s_k)$ containing signal strength from each of k APs (this is a **fingerprint**) at many known locations; store in a database (the **radio map**)
    2. Operation phase: mobile receives beacons from APs ; match vector of signal strengths in the radio map.
  - RADAR (pdf)
    * Training phase: Store average signal strengths for each location
    * Operation phase: Match measured fingerprint $(s'_1,\dots,s'_k)$ to the nearest neighbor in signal space using Euclidean distance metric: $\sqrt{\sum_{i=1}^{k}(s_i - s'_i)^2}$.
  - Horus (pdf): represent signal strength calibration measurements as a PDF and look at the distribution of signal strengths in the online phase.
  - Problems with calibration
    * Labor intensive
    * Must be redone if environment changes (often!)
- An alternative: RF propagation modeling
  - Use an RF propagation model to predict distance from AP d based on received signal power P_x
    $P_x = P_t - \gamma \log(d) + N$, where P_t is power measured near transmitter and γ is a parameter representing the rate at which signal power decreases with increasing distance (typically learned from data).
  - Triangulate distances from multiple APs for location.
  - Example systems: EZ (pdf), RADAR variant, and others.

Zee

- Main idea: Eliminate training phase—continuously "crowdsource" radio map based on measurements and a physical building map.
- Three data sources:
  1. Signal strengths from AP beacons received at smartphones
  2. Inertial sensors from smartphones
  3. Physical building map
- Combine signal strength and inertial sensor data with constraints imposed by the map (users can't walk through walls).
- Zee example scenario (slide)
  - Walk 1: from A to D
    * Initialize probability distribution of location to be uniform across the space.
    * Use accelerometer, compass, gyroscope to estimate motion

    * Update probability distribution by eliminating possibilities that would violate physical constraints imposed by floorplan
    * Only one possible path in shape ABCD
  - **Backwards belief propagation:** infer a more certain location at beginning of walk, based on knowledge accumulated
  - **Recording WiFi measurements:** Zee runs the WiFi fingerprinting training phase simultaneously, result: tuples of $\langle$signal strength, x, $y\rangle$
  - Walk 2: D to A
    * Initialize probability distribution of location based on WiFi fingerprinting: better initial estimate.
    * After crowdsourcing enough WiFi fingerprints, can rely mostly on those.

- Zee architecture
  - Placement-Independent Motion Estimator (PIME)
    * Goal: Estimate user's motion with accelerometer, compass, gyroscope sensor data. Generates an event each time a step occurs.
    * Heading offset: Angle between the orientation of the phone and the user's direction of motion.
    * Heading offset is also an unknown, so it gets incorporated into the augmented particle filter.
  - Augmented Particle Filter (APF)
    * In prior work on localization, particles are usually location (x,y) data.
    * Problem: often can't measure location directly (in the absence of WiFi fingerprinting data); can only detect steps.
    * Zee augments location with stride length and heading offset.

- Counting Steps
  - Two steps: first walk detection, then step counting
  - Mini study on where people carry phones (men: pockets; women: hands or handbags)
  - First try for walk detection: compute stddev(acceleration) over one-second periods, for ground-truth walking and idle periods
    * Less than .01g 99% of the time user is idle
    * More than .01g "almost 100%" of the time user is walking
    * Classifier would be threshold test on acceleration
    * Authors point out hand gestures could fool the test, but don't give data to support this—their data indicate the test is good enough!
  - So exploit *periodicity* in walks:
    * Autocorrelation: Multiply and sum a delayed version of signal with itself (with mean subtracted out)
    * Autocorrelation will spike at delays equal to period of a person's stride
    * Since we don't know the stride period, try delays within a range τ_{min} to τ_{max}.
    * Once stride period is found to be τ_{opt}, reduce window to a few samples around that number, and continuously update τ_{opt} (but authors don't tell us how).
  - *When does the stride period calculation happen?* Look at σ and ψ
  - *When does a step event happen?* Every $\tau_{opt}/2$ samples when in WALKING state.
  - Evaluation of step counting (slide)
    * Did not evaluate running, only walking with phone in various positions on user
    * How did the authors determine ground truth?
    * How robust is this approach to running, skipping, jumping, chatting in hallway, etc.?
    * Mistakes count steps, but what is the true negative column counting, and why is this an issue?

- Estimating heading offset (slide)
  - See slide for definitions.
  - Want to determine direction of user relative to true north based on phone's compass reading, but there are two additional variables, magnetic offset and placement offset.
  - The quantity that is measured is the compass reading θ, so heading offset (HO) is the difference bwtween the direction of the user relative to true north and the compass reading.
  - Approach: first coarse estimate based on accelerometer then fine estimate using particle filter.
  - Coarse estimate (accelerometer)

    * Look at spectrum of the accelerometer signal
    * Peaks at multiples ("harmonics") of a fundamental frequency
    * Fundamental frequency corresponds to two step "sway"
    * Empirical fact: The **second harmonic** is **very weak** in directions perpendicular to motion, but **dominant** in direction parallel to motion.
    * Why? Parallel to walk, each step registers (this is the second harmonic). Perpendicular to walk, only hip sway registers (this is the fundamental).
    * What is the algorithm here? Not explicitly stated, but look at magnitude of the second harmonic of the spectrum in each accelerometer direction.
    * Phone measures acceleration (force) parallel and perpendicular to magnetic north (N'), allowing computation of $\alpha + \theta = \arctan(F_x/F_y)$ **[Personal communication with authors: the equation $\alpha + \theta + \gamma = \arctan(F_x/F_y)$ in the paper is incorrect]**

- Tracking using agumented particle filter (APF)
  - Simultaneously estimating user's stride length, placement offset (α), and location.
  - Stride length estimation example (Figure 13): start with a small set of initial positions, grows because of different stride lengths. When user turns a corner, incorrect stride lengths' particles collide with walls and get eliminated.
  - Update the user's location based on $\alpha + \theta$, which is HO without the magnetic offset.
  - Add noise β_i at each step to try to adapt to magnetic offset.
  - Particles that collide with walls get eliminated and a new particle randomly chosen from previous step and updated.
  - Backward belief propagation traces surviving particles back in time to estimate location at beginning of a walk.
- WiFi beacon based localization
  - Zee periodically scans for beacons transmitted from nearby WiFi APs, records received signal strength (RSS).
  - Series of readings: (location, AP identifier, RSS)
  - Two localization approaches: Horus (map-based) and EZ (modelling-based)
  - Initialize location probability distribution of particle filter with WiFi based location information
  - The authors don't discuss continuously feeding in WiFi localization information in to the particle filter
- Evaluation
  - Methodology
    * One user carrying a phone continuously for 15 hours
    * Only process accelerometer data while walking detected in the past ten seconds
    * Collect WiFi measurements continuously. Why?
    * Stop and start Zee to generate different walks
  - Questions to answer:
    * How well is Zee able to track users? (slide)
      · Run Zee with particle filter in forward direction
      · High initial error (initial location uniform across floor), turns improve localization error to less than one meter
      · Take nine checkpoints (predetermined locations where ground truth location is recorded and compared with Zee's reported location)
      · This graph shows just walk #1
      · Somewhere between checkpoints at steps 80 and 100, a turn eliminated spurious location possibilities in particle filter and error dropped sharply.
      · What data could have substantiated this claim?
    * Does lack of knowledge of initial location or heading offset problem impair system performance?
      · Knowing HO alone, Zee converged much faster (particle filter has less uncertainty)
      · Knowing HO and initial location, Zee converged even faster after 60 steps, but error started low, then increased (stride length estimation error, then decreased again). **Authors claim incorrect stride length estimation caused this.**
    * How well does stride length estimation/HO estimation work?
      · What could strengthen our confidence in this result even fruther?
    * Initial location data from previous walks improves performance

    - Used EZ (model based approach) to provide WiFi location data at **starting location**
    - What source of information did the authors not incorporate?
    - Not quite as accurate as backprop; why?
  - How well do WiFi-based localization schemes perform when using location maps that Zee builds?
    - Baseline: collect WiFi beacon measurements at 117 locations, 1000 beacons per location