Topical Issues: Location Fingerprinting

CST Part II

Dr Robert Harle

Indoor Location I

- For decades we have had GNSS (Global Navigation Space Systems) such as GPS providing us with great location info for outdoor spaces
- Indoors, however, they don't work
  - Signals don't penetrate directly – if you get them at all then they've usually bounced off buildings etc and are useless for accurate positioning
  - Even if they did, the location scale for indoors is not the same as outdoors.
    - Outdoor landmarks are separated by the order of tens of metres so 10m accuracy is great
    - Indoors a 10m accuracy is hopeless – it only locates you to a portion of the building.

Indoor Location II

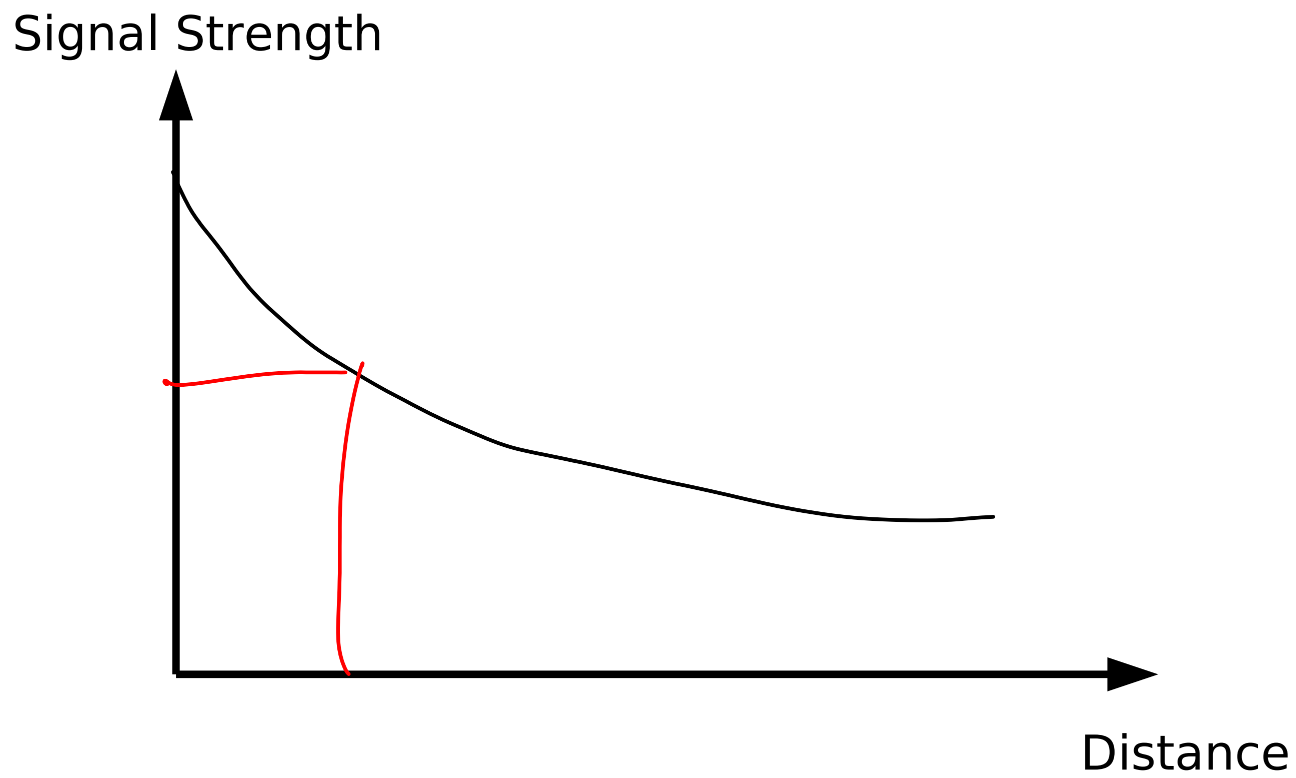
- So we need a different set of signals for indoor location
- Ideally we want something ubiquitous
  - Compatible signals in different buildings
  - Compatible tags/location devices
  - But getting whole building coverage usually means very high installation and maintenance costs

- Around 2000, researchers started to wonder whether they could use WiFi signals for positioning
  - Already deployed in buildings
  - Designed for total coverage
  - People have WiFi devices (laptops back then, phones now)

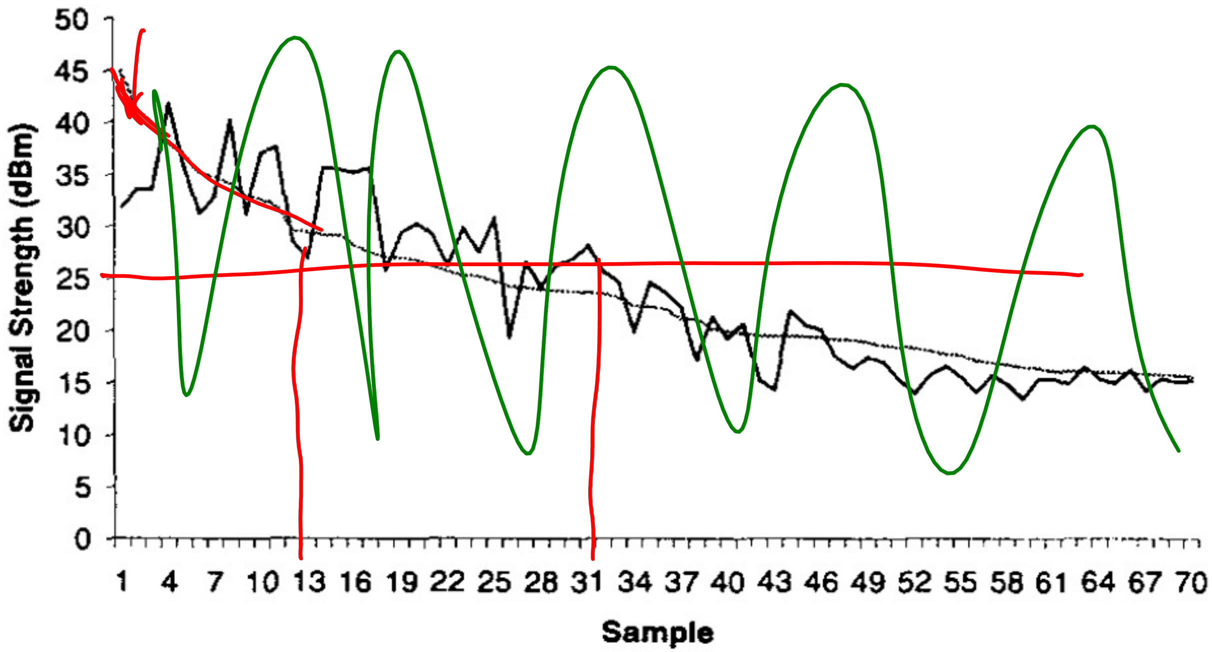
- Piggybacking positioning

Deterministic Approach

- The first attempts used a deterministic radio propagation model and ToA
- See *"RADAR: An In-Building RF-based User Location and Tracking System"* by Bahl and Padmanabhan



Results



- [Taken from RADAR paper]
- These results are suspiciously good! Most people can't get anything close to this because of:
  - Multipath interference
  - Building attenuation
  - Antenna orientation issues

RSSI

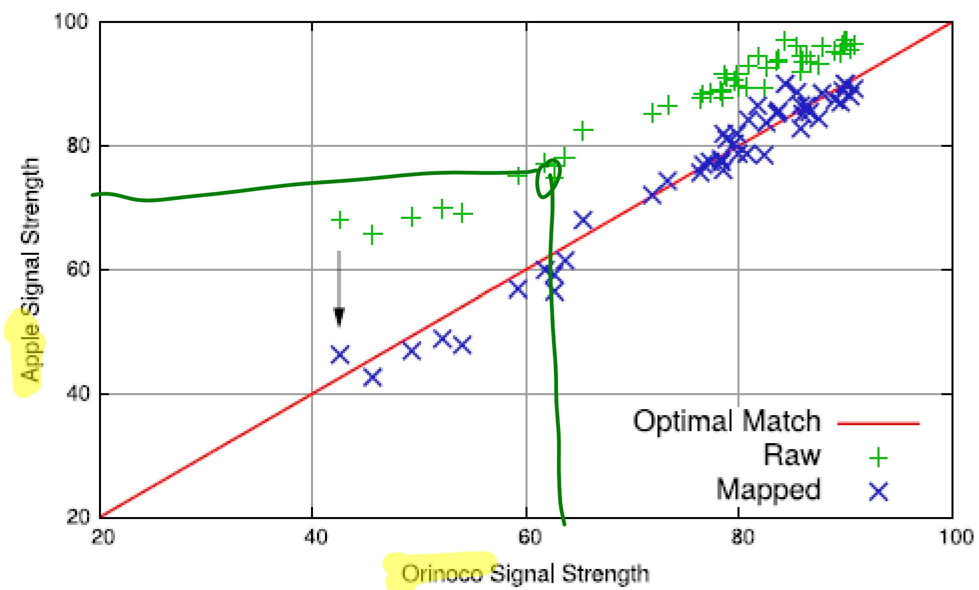
- Note that the previous graphs quoted signal strength in terms of dBm ($P_{dBm} = 10\log_{10} P_{watts} + 30$)
- These are absolute units of power. Usually, however, we just get given a Received Signal Strength Indicator (RSSI) that is an integer that maps to the actual power
- Unfortunately, the mapping is not standard and different manufacturers use different formulae :-(
- When using multiple devices, either calibrate their RSSIs or look up the mapping in use (assuming the manufacturer publishes it – most do somewhere)
- For many systems, more negative RSSIs mean weaker signals

Wifi RSSI

802.11 Spec

14.2.3.2 RXVECTOR RSSI

The receive signal strength indicator (RSSI) is an optional parameter that has a value of 0 through RSSI Max. This parameter is a measure by the PHY sublayer of the energy observed at the antenna used to receive the current PPDU. RSSI shall be measured between the beginning of the start frame delimiter (SFD) and the end of the PLCP header error check (HEC). RSSI is intended to be used in a relative manner. Absolute accuracy of the RSSI reading is not specified.



- This is taken from *"Indoor location fingerprinting with heterogeneous clients"* by Kjaergaard
- Wifi reports a number 0-255 but the spec doesn't say how to assign the numbers!

- Kjaergaard had to add in mapping of one device's output to every other in order to be able to use heterogeneous clients (blue crosses)

Fingerprinting

- Bahl and Padmanabhan had another solution
- Change the problem to one of pattern matching

  - **Offline Phase**
    - Make a map of the radio environment by measuring the signal strength (RSSI?) at many known locations spanning the area of interest (might need to use multiple devices and mapping of RSSI values)

  - **Online Phase**
    - Sample your local radio environment and lookup a position for it in your map
    - Question is how to store the map and how to do the matching?

Nearest Neighbour (Deterministic)

Nearest Neighbour in Signal Space (NNSS)

- Offline
  - At each survey point, $\mathbf{p}^i$, take a series of measurements and (usually) combine them to give one vector, $\mathbf{s}^i$, for that point (e.g. form a mean vector)

- Online
  - Measure a signal vector $\mathbf{m}$
  - Identify the nearest $\mathbf{s}^i$ to $\mathbf{m}$
    - *Nearest* requires some notion of distance: obvious choice is euclidean distance but other options are possible

$$D^i_{euclidean} = \sqrt{\sum_{j=0}^{A} |m_j - \hat{s}^i_j|^2}$$

  - Return the position associated with $min(D_{euclidean})$

kNN

- Can obviously extend to kNN i.e. identify the k nearest neighbours and then estimate the position using a weighted average

$$\hat{\mathbf{x}} = \frac{\sum_{i=0}^{k} w_i \mathbf{p}^i}{\sum_{i=0}^{k} w_i} \qquad w_i = \frac{1}{D^i_{euclidean}}$$

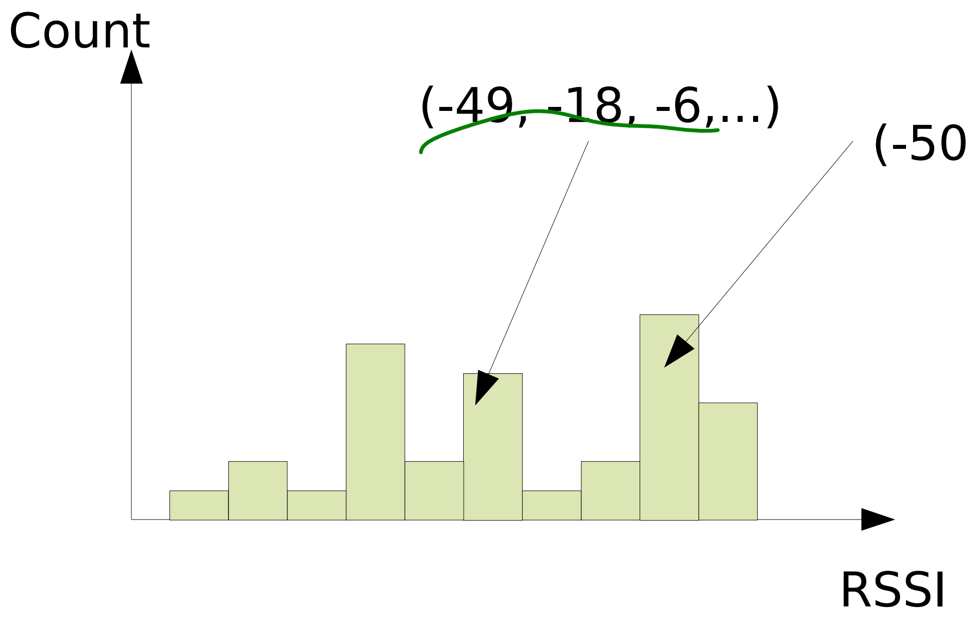
- Most results have found k=3 or 4 optimal for WiFi
- But if you have a high density of survey points, k=1 works fine.

Probabilistic Framework

(Non-deterministic)

Probabilistic Approach: Offline I

Survey the RSSIs multiple times at each survey point, but now keep a histogram of the vector occurrences. E.g. for $\mathbf{p}^j$



- This allows us to approximate the joint probability

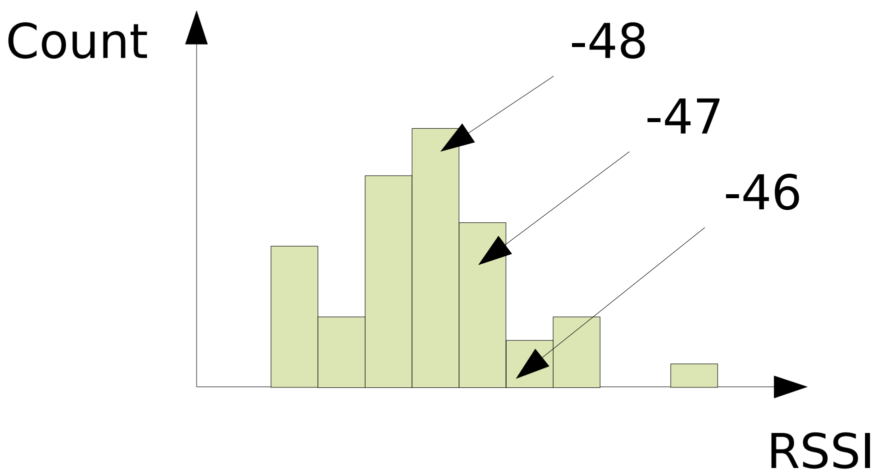
$$P(AP_1 = s_1, AP_2 = s_2, \ldots | \mathbf{p}^j) = \frac{count(AP_1 = s_1, \ldots)}{num}$$

Probabilistic Approach: Offline II

- **Problem**: Getting a statistically significant number of occurrences of every possible signal vector isn't remotely practical (i.e. *count(...)* is not stat. sig.)
- So we make a sensible assumption: that the RSSIs from different APs are independent:

$$P(AP_1 = s_1, AP_2 = s_2, \ldots | \mathbf{p}^j) = \prod_{i=0}^{A} P(AP_i = s_j | \mathbf{p}^j)$$

- Now just collect a histogram per AP



Probabilistic Approach: Online I

- We want to compute:

$$P(\mathbf{p}^j|\mathbf{m})$$

- Apply Bayes' theorem:

$$P(\mathbf{p}^j|\mathbf{m}) = \frac{P(\mathbf{m}|\mathbf{p}^j)P(\mathbf{p}^j)}{P(\mathbf{m})}$$

Probabilistic Approach: Online II

- Because we only care about the most probable position, that normalising factor is just a constant that we can ignore since we're really trying to find:

$$argmax(P(\mathbf{m}|\mathbf{p}^j)P(\mathbf{p}^j)) = argmax(P(\mathbf{m}|\mathbf{p}^j))$$

Alternative Likelihood Estimates

- Although we used a simple normalised histogram approach, there are other ways to model the likelihood
  - Gaussian
    - Fit a gaussian to the histogram and store params
    - Good: simpler to store or transmit; 'fills' in gaps in the histogram
    - Bad: histogram might not be gaussian!

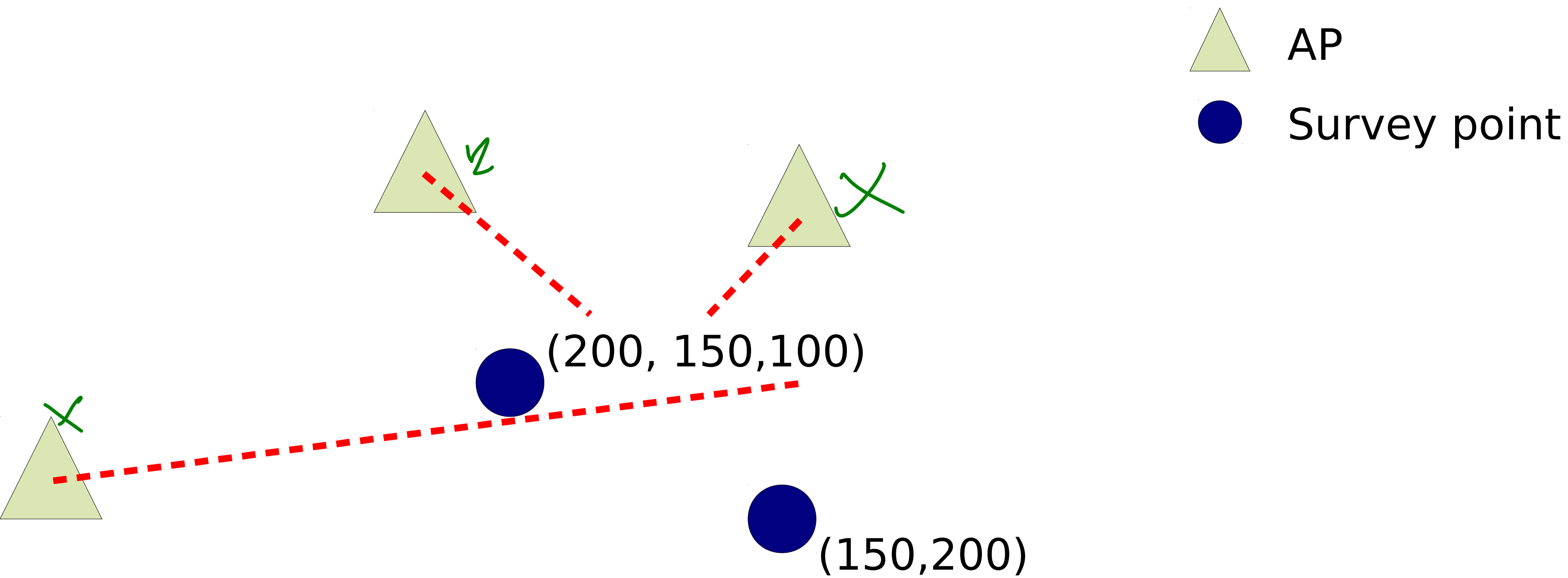
Alternative Likelihood Estimates

- Parametric
  - Fit a general function and store params
  - Good: simpler to store or transmit; 'fills' in gaps in the histogram
  - Bad: How do you choose a function suitable for all histograms?
- Kernel
  - Non-parametric approach
  - Good: more general representation; 'fills' gaps
  - Bad: more complex to work with

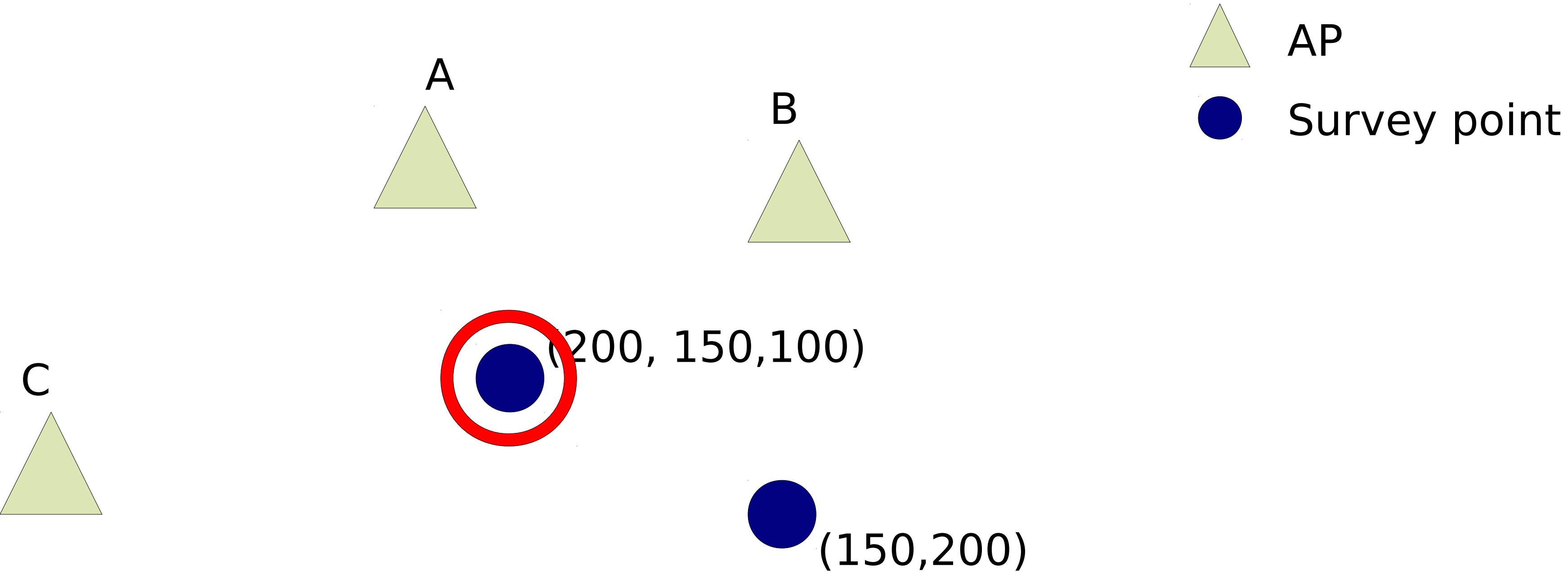
Missing Signals I

- So what happens when the measured vector doesn't contain readings for all A APs at a site?
- E.g. survey has AP_1 with {-70,-69,-70} at location **p** but **m** does not contain AP_1 at all
  - kNN approach not so bad because it just adds in a big penalty for that AP – relative to other APs the true location should still win out

Example (NN)



Example (NN)

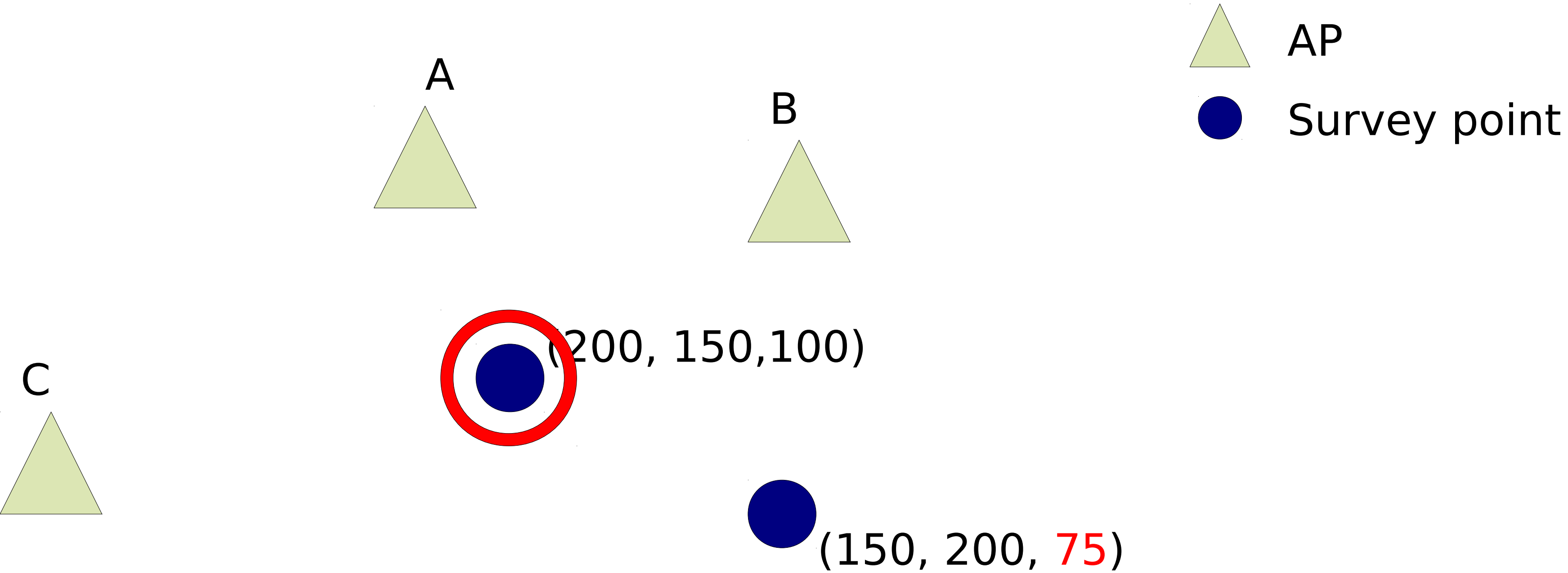


Imagine C dies. We stand in the red circle and measure (200,150)

Dist 1: sqrt(0+0+100*100) = 100
Dist 2: sqrt(50*50+50*50) = 70.71

Oops!

Example (NN)



Imagine C dies. We stand in the red circle and measure (200,150)

Dist 1: sqrt(0+0+100*100) = 100
Dist 2: sqrt(50*50+50*50 + ~~70*70~~) = 103.1

75x75

Missing Signals I

- So what happens when the measured vector doesn't contain readings for all A APs at a site?
- E.g. survey has AP_1 with {-70,-69,-70} at location **p** but **m** does not contain AP_1 at all
  - kNN approach not so bad because it just adds in a big penalty for that AP – relative to other APs the true location should still win out
  - Probabilistic approach has $P(AP_1=0|\mathbf{p})=0$ and so the likelihood becomes zero. This is fine if **p** is the wrong answer but a problem if, say, AP_1 is temporarily broken...

Missing Signals II

- Probabilistic Solution 1
  - Only compare using those APs in **both** the survey vector and **m**
  - This becomes problematic if there is only a small matching subset.
    - E.g. Only one AP in the joint set and it just so happens that the signal strength matches. Then we would compute a high probability that this is the correct location when all the *(A-1)* other APs say otherwise...
  - Probably need to enforce some minimum set overlap

Missing Signals III

- Probabilistic Solution 2

Give all APs a small, uniform probability to start with so that $P(AP_i=s \mid p^j) > 0$ for all possible s, j

  - Now the probability will always be non-zero wherever we test, but it should be negligibly small compared to the 'true' location
  - If an AP dies the probability of being at the true location will be reduced by the same proportion as the other locations so it is still the most likely location.

More General Position

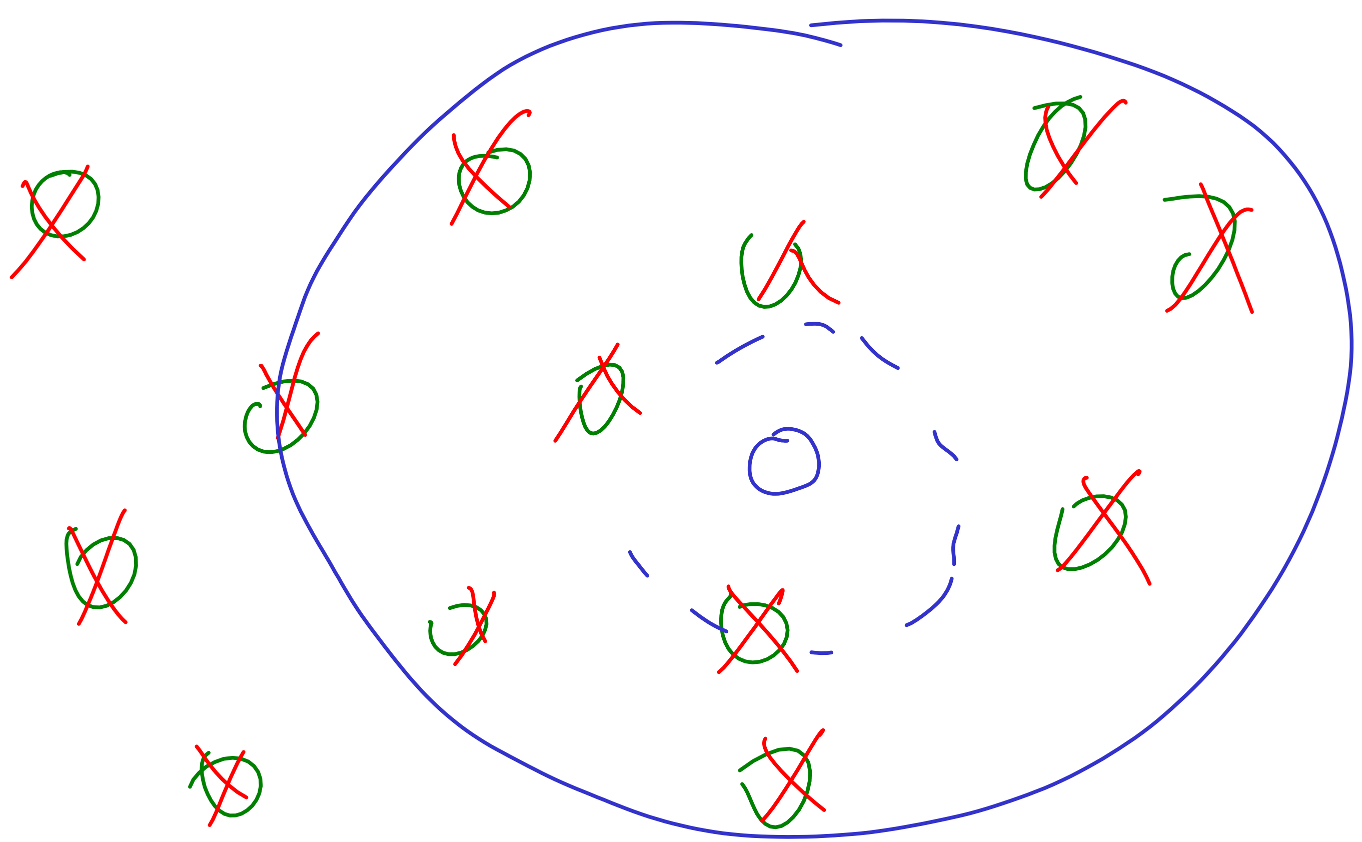
- As with the kNN approach, we can give more general locations by incorporating the top *k* probabilities into a weighted average

$$\hat{\mathbf{x}} = \frac{\sum_{i=0}^{k} w_i \mathbf{p}^i}{\sum_{i=0}^{k} w_i} \qquad w_i = P(\mathbf{p}^j | \mathbf{m})$$

Scalability

The Problem

- Ideally we survey at many, many points.
- Need to efficiently identify which subset of points to compute the probabilities for if we use prob methods (most do).



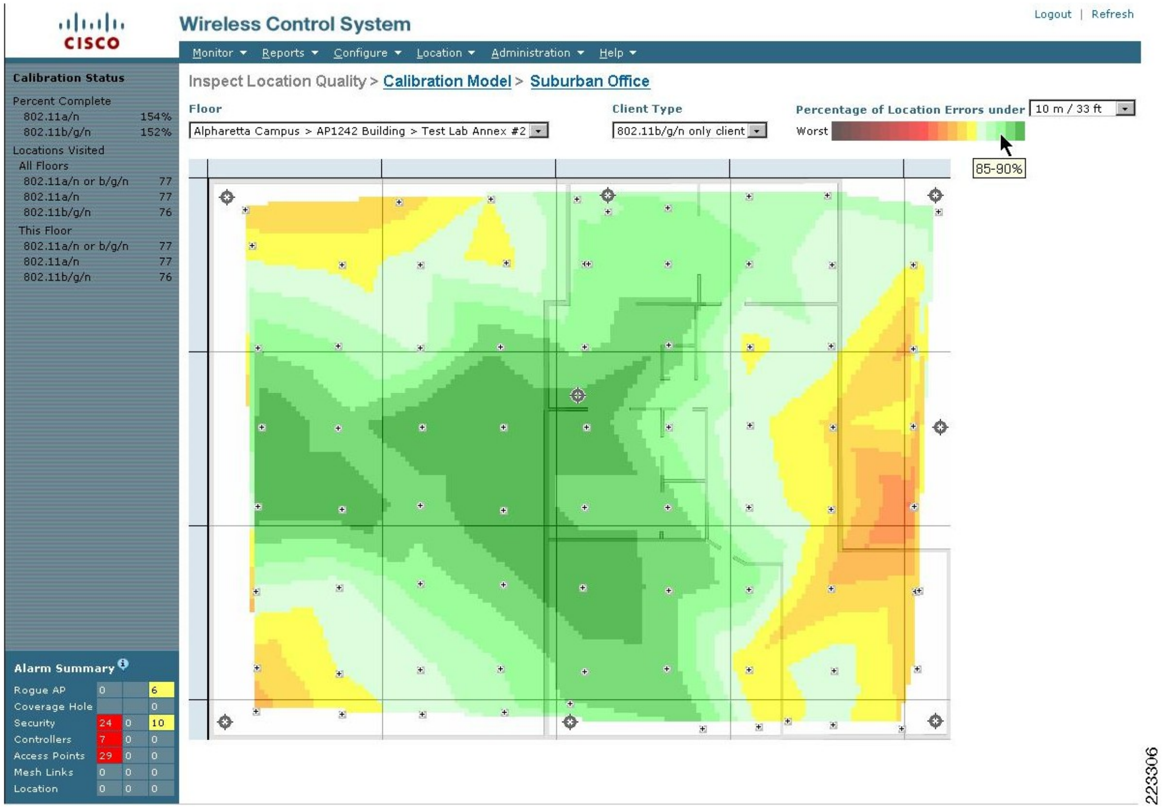
Implementations

Research Systems

- There have been a lot of attempts at fingerprint-based location tracking
- Unfortunately it's inherently difficult to pinpoint just how accurate they are. Accuracy depends on:
  - Building materials
  - Building layout and object mobility (inc. humans!)
  - Radio interference
  - Device orientation, height, and RSSI consistency
- Researchers tend to test their systems in areas of limited extent and under unrealistic conditions (it can be especially difficult to know the ground truth location!)
  - Take quoted numbers with a pinch of salt!
  - Generally accepted that wifi accuracy can get to:
    - 1m 60% of the time
    - 3m 90% of the time

Commercial Offerings I

- Cisco LBS
  - Built into some of their routers
  - Deployment tools but they advise professional installation if you want good accuracy

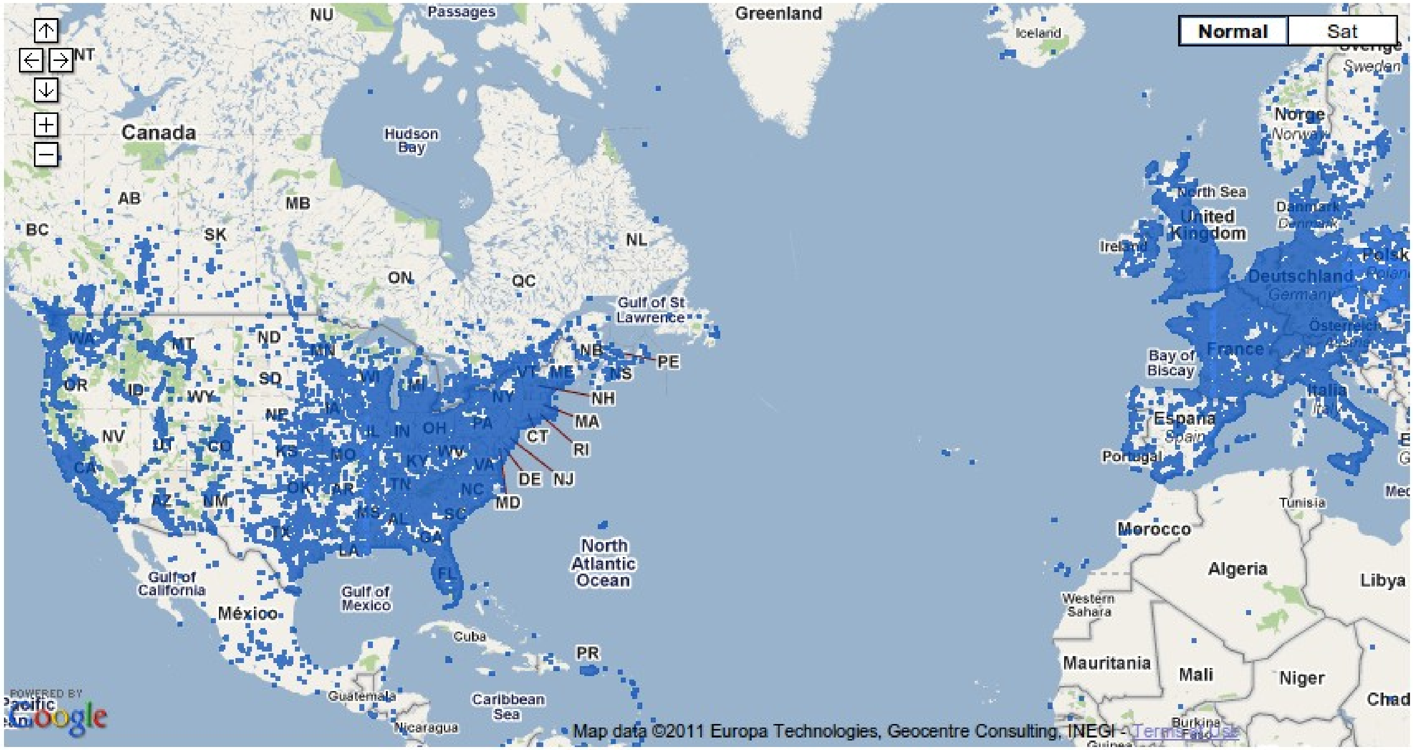


Commercial Offerings II

- Ekahau
  - Retrofit to any wifi system, but supply custom wifi tags
  - Claim “over a decade” of research into positioning algorithms
  - Make some very bold claims about accuracy and performance
  - But probably the market leader for this sort of indoor tracking

Commercial Offerings III

- Skyhook wireless
  - Special mention, even though they don't do indoors (yet)
  - Skyhook have a huge database of APs for localising WiFi devices. They obtained it through a combination of wardriving and customer manual entry



Skyhook

In April, Apple Ditched Google And Skyhook In Favor Of Its Own Location Databases

MG Siegler
Jul 29, 2010

Send | 3 | Digg
78 Comments

As you're undoubtedly aware, location is one of the hottest fields out there right now. Startups, services, devices, and advertisers are all hovering around it. As you're also likely well aware, Apple likes to be in control of their own devices. So it should come as no surprise to hear that Apple is moving to be in complete control of their own location database.

Back in June, Apple changed its privacy policy to reflect some of the newer things they were doing with regard to location. This worried some people — including two U.S. Congressmen who sent a letter to Apple asking about the change. A couple weeks ago, Apple responded to that inquiry with a letter from Apple's general counsel, Bruce Sewell. The overall main points of that letter have **already been covered** quite a bit (basically, none of the data Apple collects is linked to a specific user or device and no data is shared without consent). But buried on page 5 of the 13-page letter is a bit of information that's rather interesting.

Here's the passage (highlights are mine):

> To provide the high quality products and services that its customers demand, Apple must have access to comprehensive location-based information. **For devices running iPhone OS versions 1.1.3 to 3.1, Apple relied on (and still relies on) databases maintained by Google and Skyhook Wireless ("Skyhook") to provide location-based services. Beginning with the iPhone OS version 3.2 released in April 2010, Apple relies on its**

Got a tip? Building a startup? Tell us

Most Popular

Now | Commented | Facebook

- TechCrunch Interview: Marissa Mayer Reveals The Two Pillars Of Google's Local Strategy
- AOL Launches "Incredibly Easy" Video Chat Service Internally; We Launch It Externally
- Android To Surpass Apple's App Store In Size By August 2011: Report (Exclusive)
- A Few Key People Really Can Make a Huge Difference
- Brazil's Peixe Urbano Takes "Significant Growth" Round From General Atlantic and Tiger Global

Commercial Offerings IV

- Skyhook wireless
  - They power Apple's location engine for iphones etc, claiming 10m accuracy 99.8% of the time.
  - We know that they fingerprint, but not the details of the algorithm they use (there are a series of patents in their name, but they're not all that revealing).

System Issues

AP Density

- Generally speaking, APs are deployed to give ubiquitous *comms* coverage
  - So overlap at the edges; stripe the radio channels to prevent interference there

- But fingerprinting is going to be generally better the more APs we can hear at a given position
  - Therefore there is a commercial disadvantage here:
    - More APs must be deployed
    - Might *degrade* the comms features (more interference)!

Short-Term Changes

- To a radio, people are big bags of water
  - At ~2GHz, water will absorb radio power
  - So the fingerprint will change as people move around you
  - You will detune your antenna different amounts according to where it is on you

Survey Realities

- In order to survey, you need to know where your transmitter is
  - Which needs an indoor positioning system...
  - (OK outside because you can drive around)
  - Lots of work just surveys 'roughly' – finger on a map etc

Survey Adaptation I

- Over time surveys get out of date
  - Environments change
  - If it's a single big change (e.g. new APs deployed) then all bets are off
  - Thankfully most changes are incremental and there's an opportunity for us to adapt to them autonomously
  - For example, Skyhook have a self-healing database
    - If a measurement comes in with a new AP in it, they compute a position without that AP and then add in the AP at that position
    - If an AP moves, they try to spot the odd-one-out and treat it similarly
    - Works quite well, except that attackers have shown this makes it very easy to break (spoof your AP, jam others, etc)
  - Not really a solved problem!

Survey Adaptation II

- Ekahau FAQ: **How often and when do I need to re-calibrate the mapped area with ESS?**
- "The simple answer in most cases is: never. However, reconstruction occurs where walls or doorways are sometimes moved. In these instances, you would have to re-calibrate the impacted area only. You would have to conduct a site survey of the Wi-Fi anyway to verify that your Wi-Fi is still good for its original use and would have to get a new map showing the new layout of the floor plan."

Hmmm.

Using Other Signals

- Fingerprinting works for any type of signal that is expected to have locally constant power levels
  - WiFi, ZigBee, 2G, 3G, 4G, Bluetooth
- It's also an easy way to fuse together different types of signal
  - But remember we ideally need a multitude of signals at each location and a survey that's dense enough to provide the desired accuracy and capture any possible trends
  - E.g. Wifi indoors often has small null zones caused by destructive interference of multipathed signals. The size of the null zones is $O(\text{wavelength})=O(12\text{cm})$. So a signal can vary from strong to null in just a few cm...
  - Outdoors we also have to consider practicalities: environment changes fast (vehicles, people); large survey area; each AP needs a power source...

Conclusions

- Location fingerprinting has been remarkably successful and looks here to stay
- However, fine-grained location estimates from them re still very much a research topic – there are lots of unanswered questions as to how you deal with changing fingerprints
- Moral: choose the technique according to the application