Position and Location

Dr Robert Harle
Mobile and Sensor Systems
Lent 2020

Why Location is Important

The shift to mobile devices mean they are used in a varity of different contexts.

Location is often the main signal for context.

For example, if you are in the kitchen, you are probably not going to be using Strava. At the swimming pool? Not the best time to offer recipes. In the toilet? Maybe we can auto-decline that video call...

Many Applications in its own Right...

- Safety – efficient evacuation, tracking children, monitoring high risk areas (building sites etc)
- Energy Reduction – enables smart buildings to optimise heating/lighting/etc
- Space usage – assess whether office layouts are optimal
- Security – auto-locking doors, computers, etc
- Navigation – unfamiliar buildings, resource finding
- Collaboration – where is X? Is Y in yet?
- Resource routing – nearest telephone
- Retail – find the right item, intelligent shopping
- Health – activity level monitoring, care of elderly
- Lots of new things we haven't thought of yet...

Proximity / Microlocation

Proximity Location

Simplest location system is to mark some area with a measurable signal with limited range.

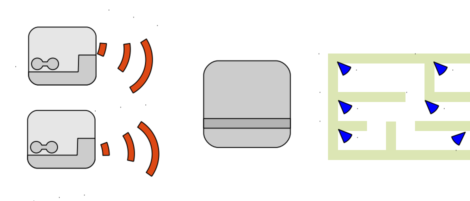
When the mobile device measures that signal, it knows where it is.

Typically very coarse location (area of building).

Active Badge

1990s: Badges that transmit an ID using infrared in a way analogous to your TV remote. IR Sensors installed in the environment listened for the IDs.

Provided room-scale proximity location



BLE Beacons (iBeacons)

Bluetooth Low Energy (BLE) has an advertising mechanism that can be used to place anchors, or 'beacons' for location

Handset sees beacon → looks up beacon position → assumes it is there. Beacon range 1-3m.

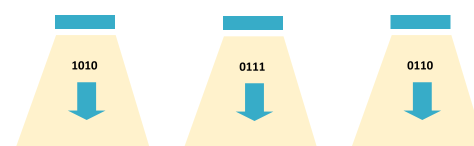
BLE compelling here because easy to deploy (batteries); minimal maintenance (beacons long lasting); small packages; cheap; widely supported

Philips Flashy Lighting

Lights are naturally constrained in space

Philips encodes an ID per light to 'label' a space

But it has to flash fast or we would perceive it



Cunning use of Camera

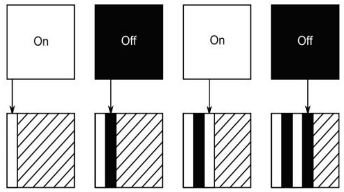


Fig. 3. Rolling shutter operation: The top squares indicate the state of the LED light. The bottom squares show the output of the CMOS camera as it enables one scanline at a time, forming the final image.

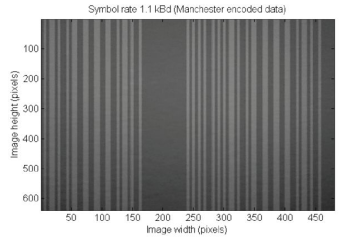
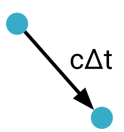


Fig. 4. Sample image of CMOS camera output – The result of the rolling shutter as taken from the preview mode of the Dell Streak mobile phone (at 1.1 kBd transmitted baud rate).

. Danakis, M. Afgani, G. Povey, I. Underwood and H. Haas, "Using a CMOS camera sensor for visible light communication," 2012 IEEE Globecom Workshops, Anaheim, CA, 2012, pp. 1244-1248. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6477759&isnumber=6477486

Time of Flight (ToF)

Time of Flight (ToF)

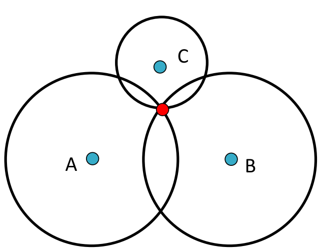


Measure the time it takes for some signal to propagate between two devices

Gives us a distance estimate ("ranges") if we know the speed of the signal, c.

Given multiple ranges from different base stations, and the base stattion locations, we can derive a location for the mobile device.

This is **trilateration**. If we use more than three distance (adding redundancy) it's **multilateration**.



Common Problems

- Syncing the devices accurately enough. If the signal is radio, **Each microsecond second of sync error is a 300m distance error**!
- The signal must be direct ("Line of sight"). Multipathed signals (i.e. signals that have bounced around to get to us) result in overestimates of distance.

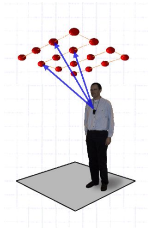
Sync by Speed Differential

If you choose two signals with **very** different propagation speeds, you can treat the faster one as propagating instantly.

For example, radio waves travel at 300,000,000 m/s while sound waves move at 343 m/s. So if we **simultaneously** emit an audio signal and a radio pulse and assume the latter is instantaneous, we introduce a timing error in the audio of 343d/3e8 = 3e-6d for a range d.

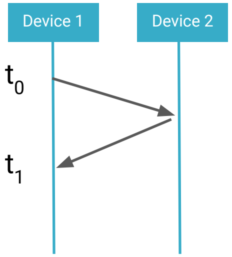
Example: the Bat System

- Developed in Cambridge (CUED/AT&T Research Cambridge)
- System starts clock and emits 433 MHz radio pulse
- Bat receives pulse and emits 50 Hz ultrasonic pulse
- Ceiling receivers measure pulse reception times
- **Achieves 3cm accuracy 95% of the time in 3D!**



Avoiding Sync with RTT

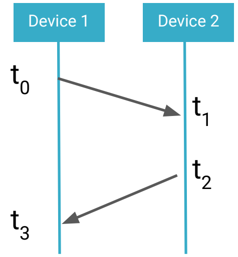
One option is to look at Round Trip Time (RTT). A device sends a signal to another device and starts a clock. That device retransmits the signal and the original device stops its clock when the reply is sensed.



$$d(A, B) = c\,(t_1 - t_0) / 2$$

Avoiding Sync with RTT

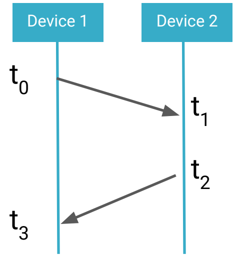
Unfortunately we can't just instantly bounce a signal back. It takes time to process and respond. And that time is not usually known.



$$d(A, B) = c\,((t_3 - t_0) - (t_2 - t_1)) / 2$$

Avoiding Sync with RTT

However, we can measure the time taken on each device if the hardware supports it. And because it's a time **difference**, it doesn't matter if device A and B have different clock times! I.e. no sync needed.



Independent of global time

$$d(A, B) = c\,((t_3 - t_0) - (t_2 - t_1)) / 2$$

Independent of global time

Example: WiFi Fine Time Measurement (FTM)

(Very) recent chipsets support the necessary timing circuitry (must be in h/w) to get the two time differences.

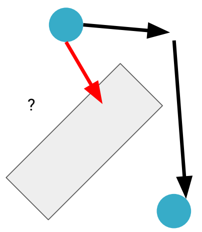
Fires hundreds of FTM messages, which get returned along with the measured time difference

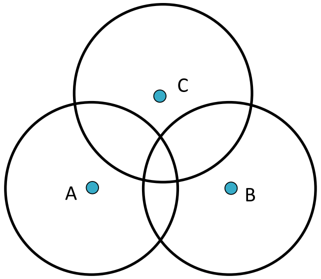
Pixel phones support it. But not many routers do yet.

Non-Line of Sight: Multipath

Signals typically bounce off something in the environment

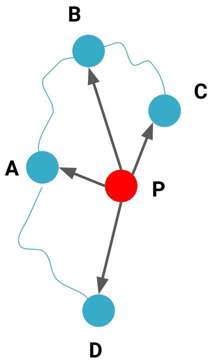
If we don't get line of sight (LoS), we overestimate the distance and our ranging circles are inconsistent.





Time Difference of Arrival (TDoA)

Sync the Base Stations (Only)



Base stations A, B, C, D synchronised either by cable or possibly using GPS.

A, B, C, D measure the time of pulse reception **in the same timeframe** as each other.

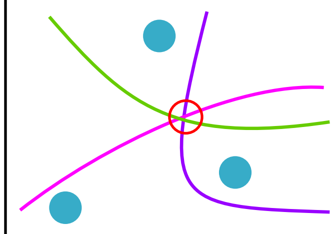
$$c(t_A - t_P) = d_{AP} \qquad (1)$$

$$c(t_C - t_P) = d_{CP} \qquad (2)$$

$$c(t_A - t_C) = (d_{AP} - d_{CP}) \qquad (1) - (2)$$

Does not involve P's unsynced clock!

Sync the Base Stations (Only)



$$c(t_A - t_C) = (d_{AP} - d_{CP})$$

$$c(t_B - t_C) = (d_{BP} - d_{CP})$$

$$c(t_A - t_B) = (d_{AP} - d_{BP})$$

Hyperbola equations
Intersection gives position

Example: U-TDoA

Phone networks apply TDoA to locate phones

Sync their base stations using GPS (see later)

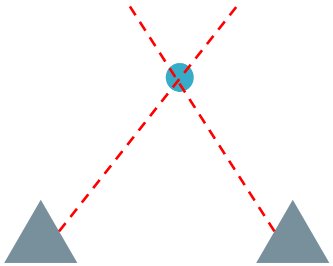
Often auto-enabled duing emergency calls

Accuracy typically 30-400m, depending on how much multipath there is

(Note you can't stop them doing it)

Angle of Arrival (AoA)

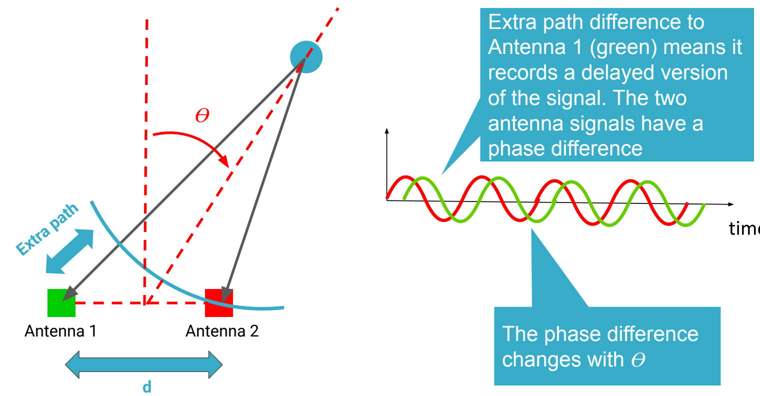
Angulation



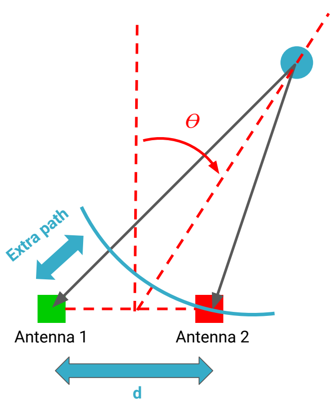
Can estimate the position of a transmitter if you can get bearings to the signal from multiple sites (at known locations).

But how can you get the directions?

Antenna Arrays



Antenna Arrays



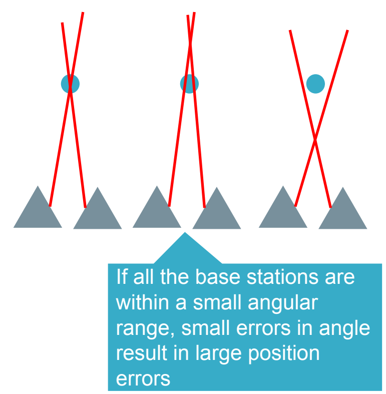
When $\theta=0$ the paths are the same (no phase difference)

When $\theta=\pi/2$ the path difference is the element separation, d

We set $d=\lambda/2$ (half a wavelength).

If we make d longer, there will be multiple θ for certain phase differences. We can make it shorter, but then we need to measure the phase difference more accurately.

Geometry Matters



Example: Bluetooth 5.1

The latest Bluetooth introduces support for antenna arrays working exactly as we described.

Likely to feature in smartphones, providing direction finding to help find things.

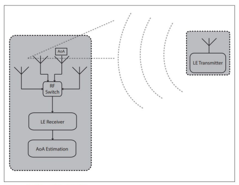


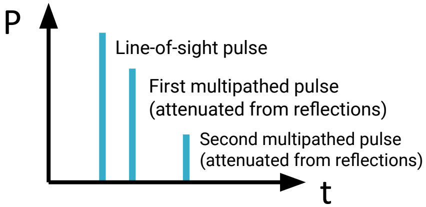
Figure 8.1: Angle of Arrival method

Case Study: UWB and Ubisense

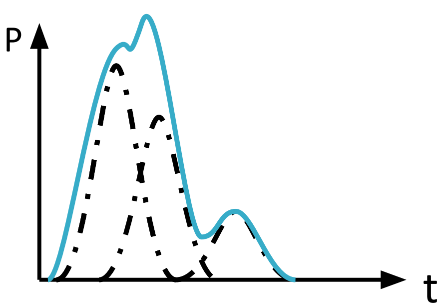
Radio signals penetrate (most) walls.

This is attractive for positioning since it means fewer base stations are needed to cover an area.

However, as well as passing through walls, RF signals are also reflected (multipath).



Multipath with Standard RF

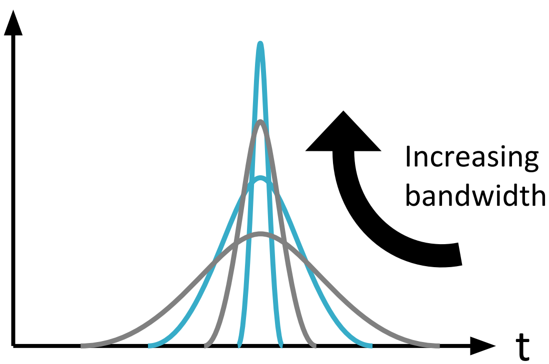


In reality, we don't have super-fast ulses

Instead every pulse has a finite width.

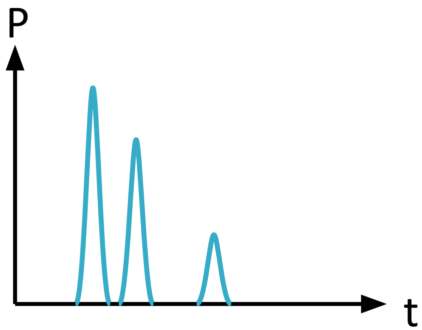
Unfortunately, the width is typically of the order of the time between reflected pulses and they all mere together to make it hard to get the first (direct) pulse

Bandwidth and Pulse Width



The more frequencies we have in a signal (i.e. bigger bandwidth), the sharper the pulse

UWB

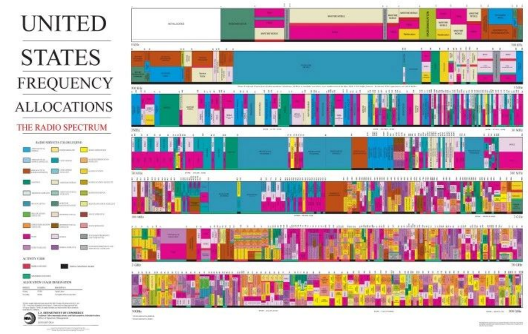


Ultrawideband (UWB) radio has a huge bandwidth (5GHz+) so can produce much more narrow pulses.

This allows us to identify the first (direct) pulse in the data for typical indoor environments.

Hence UWB well suited to positioning.

But doesn't UWB Interfere?



UWB is an Underlay System

Any 'standard' radio system has to be able to cope with a certain amount of noise. To do this we set the transmission power to be well above the 'noise floor'.

A UWB signal uses roughtly the same amount of power, but spreads it over 5GHz (not 20MHz). At any given frequency the UWB power is below the noise floor for a narrowbnd radio working there.

Called "underlay".

Ubisense

Ubisense spun out of the lab. UWB receivers placed in environment and wired together so accurately time synced

Tags emit UWB pulses that the receivers can position

Receivers use:

- TDoA based on UWB ranging
- AoA based on an antenna array in the receivers

Accuracy around 15cm in 3D @ 100Hz+!

Global Navigation Satellite Systems (GNSS)

GNSS or GPS?

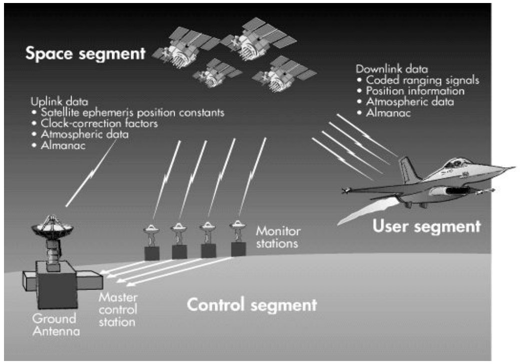
GPS (Global Positioning System) was the first satellite navigation system, owned by the USA.

Today there are many different (and mostly compatible) GNSS systems:

- NAVSTAR GPS (USA)
- Galileo (EU)
- Beidou (China)
- Glonass (Russia)

GPS will be our case study since they are all so similar anyway

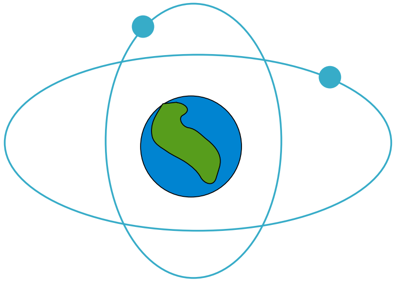
Segments



There are three ‘segments’ to GNSS:

- Space
- Control
- User

Space Segment



Satellites deployed in space around 12,500 miles away

NB. **Not** geostationary

Need a minimum of 24 satellites at any one time to get full coverage

Currently 31 operational satellites, 9 reserves, 2 being tested, and 30 retired but still up there (!)

Power

Each satellite feeds ~30W signal into its antenna.

After antenna gain and travel to Earth we get something around 10^{-16} W!! To pull this out of the noise, we need a larger bandwidth and a slow bit rate. GPS has bandwidths around 15 MHz and bit rates of just 50 bps.

Shannon-Hartley: $C = B \log_2(1 + S/N)$

Capacity, bits per second

Bandwidth (Hz)

Signal to Noise ratio (SNR)

For GPS we get S/N around 0.000002!

GPS is a TDoA system!

It's inverted from the model we had before, where the mobile device transmitted to a series of synced receiver base stations.

Our previous TDoA approach required:

- All base stations to listen to the same transmission
- All base stations to be synchronised
- All base station positions to be known accurately

Let's see how GPS reproduces these

1. Base Stations Listen to the Same Transmission

Each satellite transmits its own signal. Say satellite S1 transmits at time t_{S1} in their clock. The mobile device records when it receives a signal in its clock, ™ .

The clocks are not synchronised but we **pretend** they are to create a **pseudorange:**

$$d_{pseudo,S1} = c(t_m - t_{s1})$$

In reality, the clock offset means this isn't the real distance, which must account for the clock offset between S1 and the mobile device, $t_{offset,S1}$:

$$d_{real,S1} = d_{pseudo,S1} + ct_{offset,S1}$$

2. The Base Stations are Synchronised

The satellites are all synchronised by giving each an atomic clock (!). So:

$$t_{offset,S1} = t_{offset,S2} = t_{offset,S3} = \ldots = t_{offset}$$

Note: Since we need to solve for x, y, z and t_{offset} we need to see at least four satellites

3. All Base Station Positions to be Known Accurately

That means all satellite positions to be known accurately… and they're moving…

Each satellite in space has an **ephemeris**: a trajectory it should follow. But small errors add up.

So a series of ground stations around the Earth monitor the satellites

They have precisely known locations and atomic clocks so they know what range to expect and can compute corrections to the ephemeris

Control Segment

The control segment is the set of ground stations. It's tasks include:

- Monitor the health of the satellites, bringing up spares if needed
- Compute corrections to each satellite's ephemeris
- Upload the ephemeris corrections
- Upload any clock corrections (accounting for relativity etc).

User Segment

The user segment means the GPS receiver that needs positioning

- Downloads the **almanac**. This provides the list of satellites and approximate orbital information.
- Downloads the ephemeris information for each observable satellite.
- Computes pseudoranges and (if 4+ satellites) a GPS fix.

Revisiting the Almanac

The almanac is repeated regularly by each satellite. However, at 50 bits per second, the information takes ~12.5 minutes to transmit!

- **Cold start**. If you turn on a brand new GPS receiever, you have to search for all satellites, have untrusted time, and must download the whole almanac. Takes 15 minutes or so.

- **Warm start**. Receivers cache the rough location, time and almanac (which is valid for 180 days). If you then ask for a position, it just needs to get the ephemeris (valid for ~4 hours).

- **Hot start**. If you had a location fix within the last four hours, you can acquire a position without waiting for any satellite data.

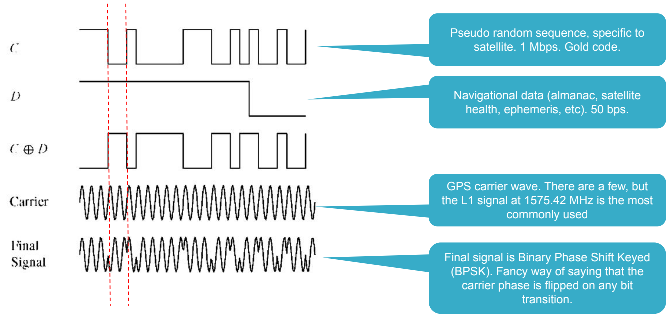
Assisted GPS (A-GPS)

How come your phone always acquires a fix so quickly?

Most phones have A-GPS, where the network sends the almanac and ephemeris data over the network data channel, allowing a hot start scenario.

GPS Signals

Remember GPS need to use a wide bandwidth (~15 MHz) to get the required SNR.



http://what-when-how.com/a-software-defined-gps-and-galileo-receiver/gps-signal-gps-and-galileo-receiver-part-2/

GPS Signals

In order to retrieve the data, we correlate the signal with the pseudo-random sequence associated with each satellite. If we are currently hearing that satellite, the correlation spikes and we are 'locked on' to that satellite.

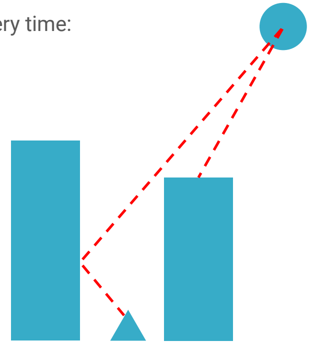
Note for this to work, the satellite pseuso-random sequences must resist correlating with each other. **Gold codes** are used - mathematically defined to give almost no correlation between each other.

https://en.wikipedia.org/wiki/Gold_code

Accuracy

Lots of sources of error, not all of which apply every time:

- Signal acquisition errors (3m)
- Ionospheric delays (~5m)
- Ephemeris errors (2.5m)
- Satellite clock errors (2m)
- Multipath (1m+)



"Urban canyons"

Improving Accuracy

Signal diversity. Two new civilian signals available: L2C (1227 MHz) and L5 (1176 MHz). Each will be affected by e.g. the ionosphere differently. If we listen to all of them together we can ~remove the atmospheric effects. Expected to get us down of 0.5 m accuracy without any tricks!

Differential GPS. Place reference stations at precisely known locations. These can then estimate corrections to the current errors and broadcast them to nearby receivers. If you're close to a DGPS station, you can get a few cm of accuracy, but it degrades the further away you are.

GPS for Time

We have been solving for (x,y,z) but implicitly that meant we measured the time offset between the device's clock and the satellites' atomic clocks.

In fact, we end up knowing global time to around 1 ns in theory!

(In practice, electronics delays get in the way and we get anything from 100 ns to 1 us. But that's still pretty good!)

Our telecommunications and banking systems and more *depend* on GNSS-provided time sync.

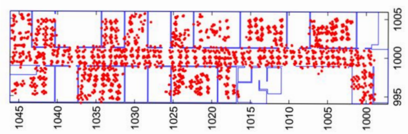
Location Fingerprinting

Signal Fingerprinitng

Developed for radio systems, esp. WiFi

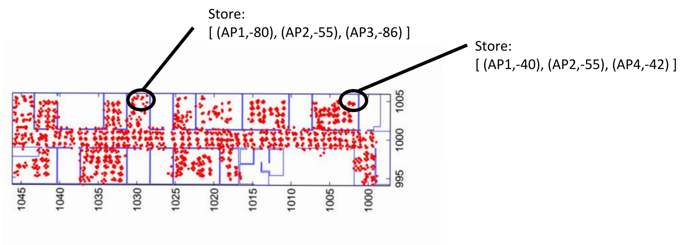
You perform an 'offline' survey, manually measuring signal properties (usually strength) at a range of spatial positions

→ "Signal map" or "fingerprint map"



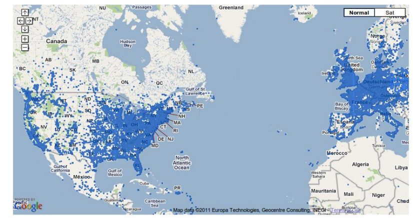
Signal Fingerprinting

- **Offline ("survey") stage**
  - Identify known locations that cover the space comprehensively (how?)
  - At each point take 10 or so WiFi RSSI measurements
  - Average and store



Signal Fingerprinting

- In the online phase, we scan the local signals and perform some form of pattern matching to the survey points
- This is essentially indoor wardriving. The outdoor equivalent is well established



Nearest Neighbour in Signal Space (NNSS)

- **Online**
  - Mobile device scans WiFi and builds a vector of observations **m** (e.g. [(Ap1,-40), (AP2,-60)]
  - Now consider every point that we surveyed and find the one "nearest" to the measurement in signal space
    - *Nearest* requires some notion of distance: obvious choice is euclidean distance but other options are possible

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

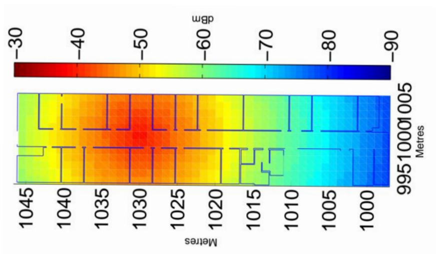
  - Return the position associated with $min(D_{\text{euclidean}})$
  - Can easily upgrade to kNN (usually k=3 or 4)

Missing values

- How do you compare:
  [(AP1,-30), (AP2,-50), (AP3,-95)] and
  [(AP1,-30), (AP2,-50)] ?
- Maybe AP3 died or you happened not to hear it on the scan (-95 is very weak)
- Naive distance metric has AP3 producing an overall distance of 95!
- Easiest trick is to assign all unseen APs the reader sensitivity (lowest value a reader can report). Usually around -100dBm.
- Then [(AP1,-30), (AP2,-50)] becomes [(AP1,-30), (AP2,-50), (AP3,-100)] and the distance is 5 – more reasonable
- Actually a much harder problem when an AP that you should be hearing strongly dies. Plenty of different approaches but all have nasty corner cases.

Regression Maps

- We can optionally use regression algorithms to create a continuous map for each base station



The good...

- You're re-using signals that already exist so no need to deploy any hardware!
  - "Opportunistic positioning"
- But the reality is...

...the bad

- APs are deployed for comms, not positioning (→ lower density, poor geometry)
- Body shadowing (we're bags of water that absorb 2.4GHz quite well)
- Environments change so radio paths change
- How do you know where you are when you survey? (you need an indoor positioning system..!)
- Device heterogeneity
- Scanning costs are high on the mobile end – disrupts normal behaviour
- Room ambiguity

Summary

- Location is an important contextual hint
- We have lots of tools to get location, but indoors there are many challenges
- We will also look at a sensor fusion approach for location using wearables in the next lecture