ES3 Lecture 8

Location-based technologies and navigation

Location Awareness Technologies

- There are lots of location awareness technologies
  - Give a location relative to a reference frame
- We will consider Earth-relative positioning
  - Rather than room or object-relative positioning
- Most obvious technology is GPS
  - Satellite constellation gives location most places on Earth
- Other technologies like WiFi positioning or cell tower location use existing ground based infrastructure
  - Triangulate distances to get location estimate

Location awareness issues

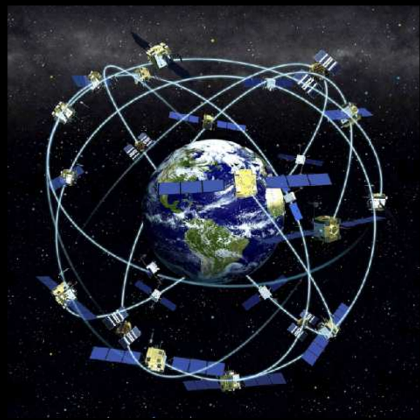
- Technical:
  - quality and accuracy of fix
    - how close is the given position to the true device location?
  - update time
    - how quickly do positions update, and how long does it take to get an inital fix?
  - how to navigate on the Earth
    - given a pair of positions, which way should you go? How far apart are two points on the Earth's surface?
  - routing
    - how can you quickly get from A to B given obstacles and constraints?

- Social issues:
  - privacy
    - Who is position information shared with, and what control do users have over this?

How GPS works

- There are 31 GPS satellites in *geosynchronous* (not *geostationary*!) orbit around the Earth
  - Each has an atomic clock knows its position relative to the earth at any given time
  - Time, *ephemeris* (accurate current location) and *almanac (*general information about all satellite orbits) is continously broadcast

- Receivers get the times and positions from the satellites
- By computing difference in received times, location can be deduced
  - Further away satellites have longer delays



GPS coverage

- GPS transmissions are effectively line-of-sight
  - If satellites are occluded by objects or are over the horizon, no signal will be recieved
  - This is why GPS doesn't work indoors or even under heavy foliage
- GPS satellites are not evenly distributed around the Earth
  - Fewer near the polar regions
  - The UK is in quite a poor coverage area

GPS Fix

- At least 3 satellites must be reliable received to get a fix
  - More satellites mean a faster and more reliable fix
- If a GPS unit has not been initialised in the current location recently, it needs to update all information about satellites before a fix can be made
  - This is slow, and can take several minutes
- The satellite orbital position data must be received from the satellites (*ephemeris*)
- GPS has slow transmission rates
  - 50 bits/second (encoded with CDMA)
  - It takes a long time to download all the relevant data
    - One "frame" takes 30 seconds (but only transmitted every 90 seconds)
    - Contains time and ephemeris, but only 1/25th of the almanac in each frame

GPS Noise

- GPS positions can be inaccurate
  - Too few satellites makes it hard to get an accurate fix
  - Reflections off objects can introduce errors (multipath errors)
  - Shadowing from buildings can interrupt signals
  - Ionosphere introduces unpredictable delays
  - Solar activity periodically disrupts GPS
    - (big solar storms occur occasionally, with a cycle of about 11 years or so)

- GPS reports how accurate it thinks values are
  - dilution of precision, or DOP
  - not always very good estimates of uncertainty, but better than nothing

- GPS is much more accurate in latitude/longitude than it is in altitude

Typical (measured) GPS noise

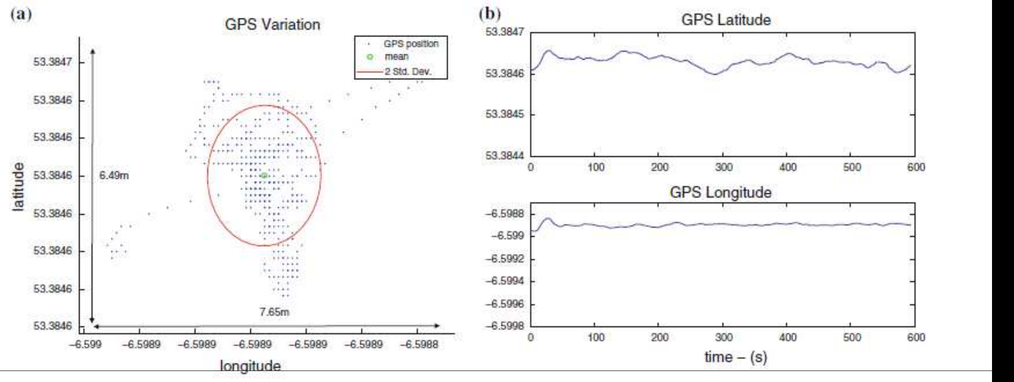
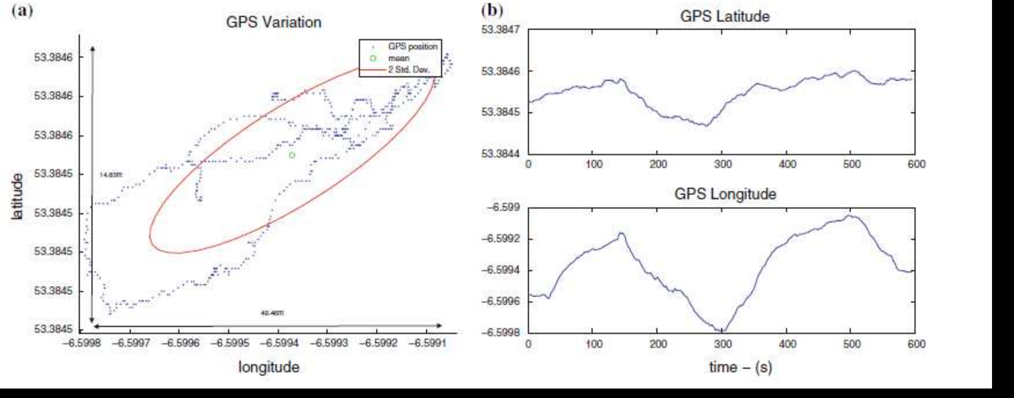


Fig. 5 *Left* GPS data recorded in a static position in very good conditions with a mean hdop value of 0.82. *Right* Corresponding time series for data over the 10 minute period. GPS device used was a Nokia LD-3W



- From Strachan and Murray-Smith, "Bearing-based selection in mobile spatial interaction", Pers. Ubiq. Comp. 2008

AGPS

- AGPS (assisted GPS) allows much faster fixes
  - Satellite almanac, current accurate time, and ephemeris information is sent via other networks
  - Usually via cell networks

- With AGPS, lock-on times can go from several minutes to a few seconds
  - Most mobile handsets support AGPS for faster fixes
  - Cell towers also allow crude positioning
    - Used to correct for ionospheric distortions

DGPS

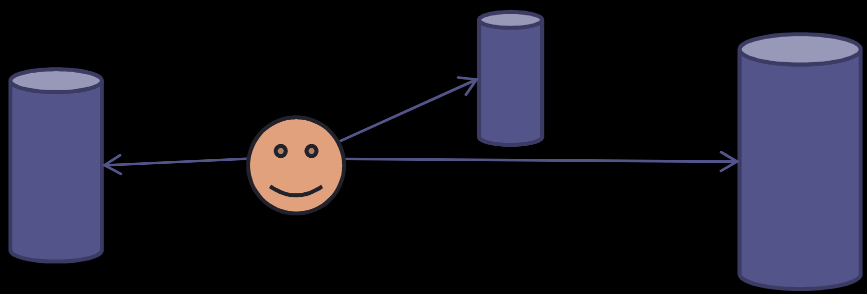
- Differential GPS (DGPS) is a technology for extremely accurate positioning using GPS
  - Often used for geological surveys, where shifts of the Earth crust in the order of a few tens of cm are involved

- Ground based references at known locations are used to correct errors in the GPS
  - Each ground station basically compares GPS estimate of where it is to true known location
  - This correction is broadcast to DGPS receivers
  - They obtain a GPS fix, then apply the correction the ground reference stations transmitted

- Requires significant infrastructure
  - Not commonly used for standard location tracking
  - but offers very high accuracy when needed

Wifi triangulation

- Location of nearby WiFi hotspots can be used to get position
  - Each has a worldwide unique MAC address
  - If multiple hotspots can be seen, signal strength can be used to improve fix
- Needs a database of WiFi hotspots
  - this data needs to be constantly collected
  - some companies offer money for GPS-fixed WiFi locations
- Relatively easy to implement, works even indoors
  - Needs no hardware beyond WiFi receiver
  - Signal strength does not vary smoothly with distance though (occlusions etc.)



Cell tower location

- Cell towers can be used similarly
  - Mobile operators know exactly where all their towers are
  - Database already exists
- Currently connected tower gives position within several hundred metres
- If multiple towers are visible, relative signal strengths can give a better fix on position
  - Difference in time of arrival of signals can also be used (U-TDOA)
  - Angle of arrival can be measured and compared by base stations (they have multiple recievers at different angles)

Bluetooth Location

- Bluetooth is sometimes used to mark specific locations
  - If you can see a particular Bluetooth ID, you are within a few metres of it
- Unlike other services, can't practically be used for tracking over large areas
- Can be used to identify when near locations
  - For example, testing if you're near a given bus stop or shop
  - Give location specific information (timetable for local bus at this stop, for example).

Hybrid positioning systems

- Hybrid positioning systems combine multiple sources of location data
  - Usually some mix of WiFi, cell tower triangulation and GPS

- GPS is good outdoors in clear spaces
  - WiFi and cell towers are dense in urban areas where GPS fails

- Devices like the iPhone and recent Nokia smartphones have built in hybrid positioning services
  - Reliant on databases of WiFi and cell tower locations
    - Some systems are user generated (use GPS to locate fixed WiFi or cell tower points)
  - Mobile operators control cell tower data

- Gives pretty reasonable coverage throughout a variety of areas
  - Usually works okay even indoors if it's a densely populated area

Dead Reckoning

- *Dead reckoning* can be used for short term position updates when location services fail

- You need to know current direction (e.g. from a compass) and distance travelled
  - cars, for example, know roughly how far they have moved from the odometer
  - pedestrians can use number of footsteps (e.g. counted from accelerometer)
    - this is much more subject to error though

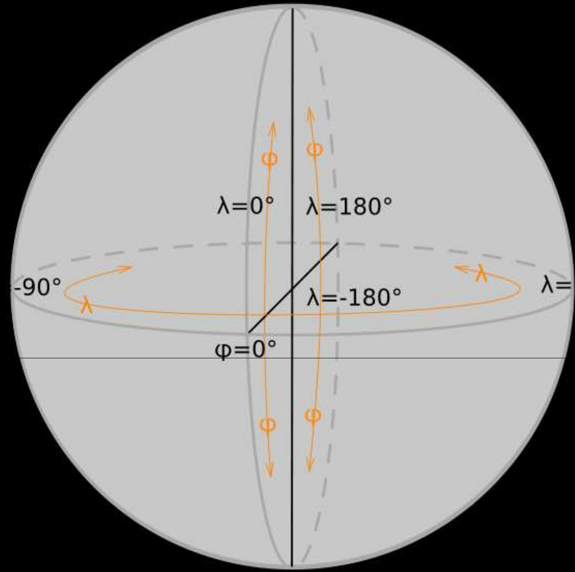
- Errors in dead reckoning usually accumulate quickly
  - Only really useful for filling in between very short location failures

Latitude, Longitude

- Earth coordinates are given as latitude, longitude
  - Latitude specifies how far north or south
    - 90 at North pole
    - -90 at South pole
  - Longitude specifies how far east or west
    - 0 at Greenwich
    - -180/180 at the international dateline

- Note that the ranges are different
  - Latitudes and longitudes are not equal divisions!

- 1 minute of latitude is always **1847m**
- 1 minute of longitude varies with latitude
  - **~1860km** at equator
  - **0m** at poles!

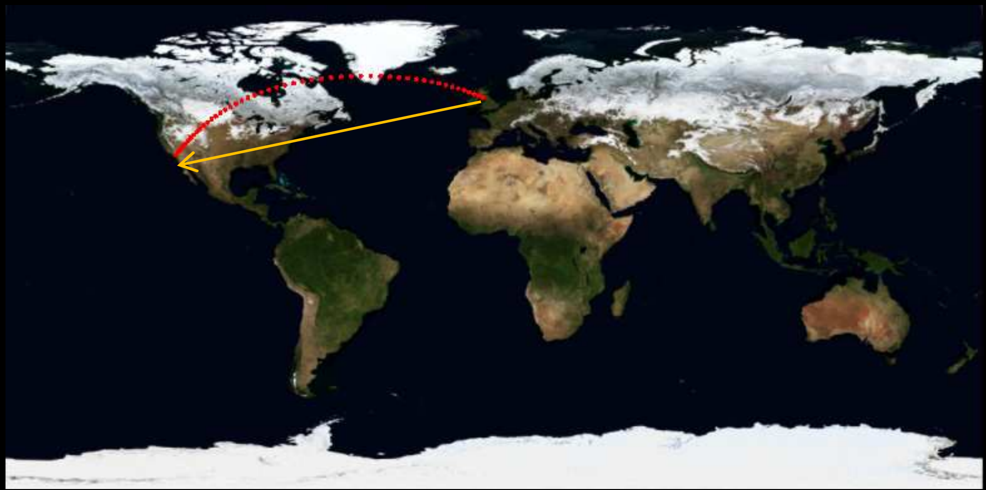


Great Circles

- The Earth is nearly, but not quite spherical
  - Slight bulge at equator

- The shortest path between two points on a sphere is not a straight line, but a **great circle**

- Flying from Glasgow to LA, the shortest route is over Iceland and Greenland, not due west-southwest!



Decimal versus minutes, seconds

- Latitude and longitude are either specified as:
  - Decimal degrees x.yy
  - Decimal minutes x'yy.zz'
  - Decimal seconds x'yy'zz.ww"

- To do computations, you must convert to decimal degrees
  - if in minutes d_decimal = degrees + minutes/60
  - if in seconds d_decimal = degrees + minutes/60 + seconds/3600
  - and vice versa
    - Decimal seconds is conventional for display

- Must also convert sign:
  - Latitude N = +ve, S = -ve
  - Longitude E = +ve, W = -ve

Decimal versus minutes, seconds

- The entrance to the department is located at 55º52'26.02"N 4º17'31.78"W
- This is in decimal seconds
- In signed decimal degrees this is:
  +55.873894, -4.292165

Distances and headings

- You can't just add and subtract latitudes and longitudes!
- There are basic formulas for calculating headings and distances from one position to another
  - http://**williams.best.vwh.net/avform.htm** lists simple algorithms used by pilots
- Note that lat, lon are given in degrees. Most implementations of mathematical functions work in radians!
  - Remember to do the conversions before computations

Distances and headings (II)

- Distance and heading between two points at lat1, lon1 -> lat2, lon2
- Assuming a spherical earth (haversine algorithm)
    - convert lat, lon from degrees to radians first!

```
distance= 2 * asin(sqrt((sin((lat1-lat2) / 2))**2 + cos(lat1) * cos(lat2) *
(sin((lon1-lon2) / 2))**2))
```

Value in radians

- Multiply by 6371000 to get distance in m (6371 km = radius of Earth)

```
heading = to_degrees(atan2(sin(lon1-lon2)*cos(lat2), cos(lat1)*sin(lat2)-
sin(lat1)*cos(lat2)*cos(lon1-lon2)))
```

- Value in radians
- This is the (initial) great circle heading
- For long distances, great circle heading changes during course!

Destination given bearing and distance

- To compute a destination point, given a starting position, a heading (radians) and a distance (km):
  - again lat, lon must be converted to radians!

```
R = 6371.0 // km (radius of the earth)
lat2 = asin(sin(lat1)*cos(distance/R) + cos(lat1)*sin(distance/R)*cos(heading))
lon2 = lon1 + atan2(sin(heading)*sin(distance/R)*cos(lat1),
cos(distance/R)-sin(lat1)*sin(lat2))
```

Intermediate points

- Another useful value is the position of a point some fraction between two destinations
  - Two-thirds of the way from LA to London
- Compute distance **d** as before (converted to radians!)
- given lat1, lon1, lat2, lon2 (in radians)
- And **f,** a fraction from 0.0--1.0 representing how far along the path

```
A=sin((1-f)*d)/sin(d)
B=sin(f*d)/sin(d)
x = A*cos(lat1)*cos(lon1) +  B*cos(lat2)*cos(lon2)
y = A*cos(lat1)*sin(lon1) +  B*cos(lat2)*sin(lon2)
z = A*sin(lat1)           +  B*sin(lat2)
lat = to_degrees(atan2(z,sqrt(x**2+y**2)))
lon = to_degrees(atan2(y,x))
```

Vincenty's Algorithm

- If you need real accuracy in measuring distances given latitude, longitude, use Vincenty's algorithm
  - Accurate to **0.5mm (!)**
  - Compared to several **metres** for the standard ("haversine") algorithm

- If you're measuring and summing lots of small distances (e.g. steps) the errors can add up, so Vincenty's algorithm becomes important
  - Or if you're guiding missiles...

- Algorithm is complex -- don't try and implement it yourself
  - Example (LGPL) Javascript implementation

```
/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  */
/*  Vincenty Inverse Solution of Geodesics on the Ellipsoid (c) Chris Veness 2002-2009            */
/* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  */

/*
 * Calculate geodesic distance (in m) between two points specified by latitude/longitude
 * (in numeric degrees) using Vincenty inverse formula for ellipsoids
 */
function distVincenty(lat1, lon1, lat2, lon2) {
  var a = 6378137, b = 6356752.3142,  f = 1/298.257223563;  // WGS-84 ellipsiod
  var L = (lon2-lon1).toRad();
  var U1 = Math.atan((1-f) * Math.tan(lat1.toRad()));
  var U2 = Math.atan((1-f) * Math.tan(lat2.toRad()));
  var sinU1 = Math.sin(U1), cosU1 = Math.cos(U1);
  var sinU2 = Math.sin(U2), cosU2 = Math.cos(U2);

  var lambda = L, lambdaP, iterLimit = 100;
  do {
    var sinLambda = Math.sin(lambda), cosLambda = Math.cos(lambda);
    var sinSigma = Math.sqrt((cosU2*sinLambda) * (cosU2*sinLambda) +
      (cosU1*sinU2-sinU1*cosU2*cosLambda) * (cosU1*sinU2-sinU1*cosU2*cosLambda));
    if (sinSigma==0) return 0;  // co-incident points
    var cosSigma = sinU1*sinU2 + cosU1*cosU2*cosLambda;
    var sigma = Math.atan2(sinSigma, cosSigma);
    var sinAlpha = cosU1 * cosU2 * sinLambda / sinSigma;
    var cosSqAlpha = 1 - sinAlpha*sinAlpha;
    var cos2SigmaM = cosSigma - 2*sinU1*sinU2/cosSqAlpha;
    if (isNaN(cos2SigmaM)) cos2SigmaM = 0;  // equatorial line: cosSqAlpha=0 (§6)
    var C = f/16*cosSqAlpha*(4+f*(4-3*cosSqAlpha));
    lambdaP = lambda;
    lambda = L + (1-C) * f * sinAlpha *
      (sigma + C*sinSigma*(cos2SigmaM+C*cosSigma*(-1+2*cos2SigmaM*cos2SigmaM)));
  } while (Math.abs(lambda-lambdaP) > 1e-12 && --iterLimit>0);

  if (iterLimit==0) return NaN  // formula failed to converge

  var uSq = cosSqAlpha * (a*a - b*b) / (b*b);
  var A = 1 + uSq/16384*(4096+uSq*(-768+uSq*(320-175*uSq)));
  var B = uSq/1024 * (256+uSq*(-128+uSq*(74-47*uSq)));
  var deltaSigma = B*sinSigma*(cos2SigmaM+B/4*(cosSigma*(-1+2*cos2SigmaM*cos2SigmaM)-
    B/6*cos2SigmaM*(-3+4*sinSigma*sinSigma)*(-3+4*cos2SigmaM*cos2SigmaM)));
  var s = b*A*(sigma-deltaSigma);

  s = s.toFixed(3); // round to 1mm precision
  return s;
}
```

Pedestrian Navigation Issues

- Slow moving receivers are much more affected by multipath (reflection) effects
  - unfortunately, in cities, where most pedestrian navigation takes place, these are especially bad
- Noise effects are particularly severe
  - a few hundred metres doesn't matter much in a car...
    - but it's a lot if you are walking
- Making user aware of uncertainty is important
  - show uncertainty circle on the map (a la Google Maps)
  - or show point cloud estimates...