Location-Based Interaction

Euan Freeman

euan.freeman@glasgow.ac.uk

University of Glasgow | School of Computing Science

Feedback for L3 Class Reps

Your L3 class reps are looking for feedback on semester 2 courses

Click here or scan the QR code to access their feedback form

CS Level 3 Semester 2 Feedback Form

Outline

What aspects of location can we sense?

- Capabilities, strengths, limitations

Designing location-based interactions

- What sensing is appropriate for a given application?
- How should location data be processed?
- What is the relationship between location and functionality?

Privacy concerns

Intended Learning Outcomes

ILO1: Explain problems associated with human-computer interaction in mobile and ubiquitous usage contexts;

ILO2: Critically analyse a proposed mobile interactive system considering its intended usage context;

ILO5: Discuss cutting edge developments in mobile human-computer interaction, such as context-aware systems, sensor-based interaction, location-based interaction, and mixed reality;

Location, Location, Location

Device location changes often – context!

Location sensing supports interaction

- Engagement
  - e.g., experiences linked to place
- Exploration
  - e.g., finding what's nearby
- Navigation
  - e.g., how to get to a restaurant
- Relevant suggestions
  - e.g., nearest coffee shop
- Location sensitive behaviour
  - e.g., turn the lights off when you leave home

What *is* Location?

Your position on the planet

- e.g., GNSS (GPS, Galileo, BeiDou, etc)
- e.g., access point trilateration

Your position in a room/building

- e.g., you are 3m from a BTLE beacon
- e.g., you have moved 1m in this direction

Your proximity to a known device

- e.g., you are ‘at home’ because your smart meter is within range

Sensing Methods

Many approaches:

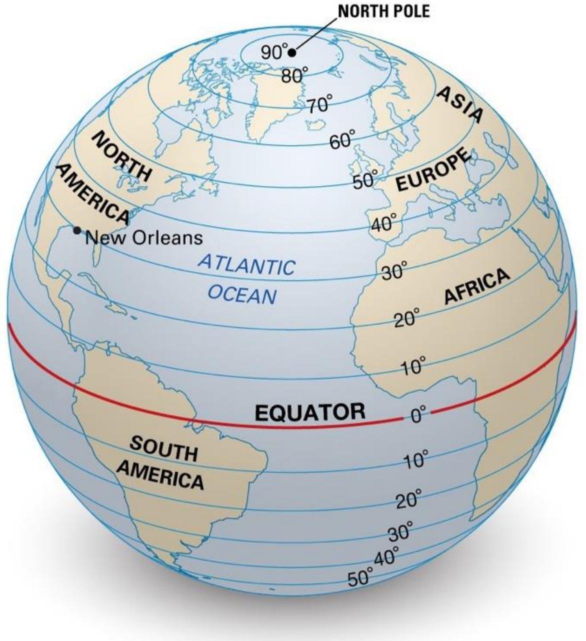
- GNSS: GPS ⊂ GNSS
- Access Points: e.g., cell and Wi-Fi trilateration
- Beacons / Tags: e.g., Bluetooth and UWB beacons
- Optical Sensing: e.g., fiducial markers, visual-inertial SLAM
- Multimodal: e.g., Google Fused Location, Apple Core Location

Need to know strengths & weaknesses

- So you can make informed choices about which to use
- So you understand the constraints of the options you’re working with

Images via Encyclopaedia Britannica

Global Navigation Satellite System

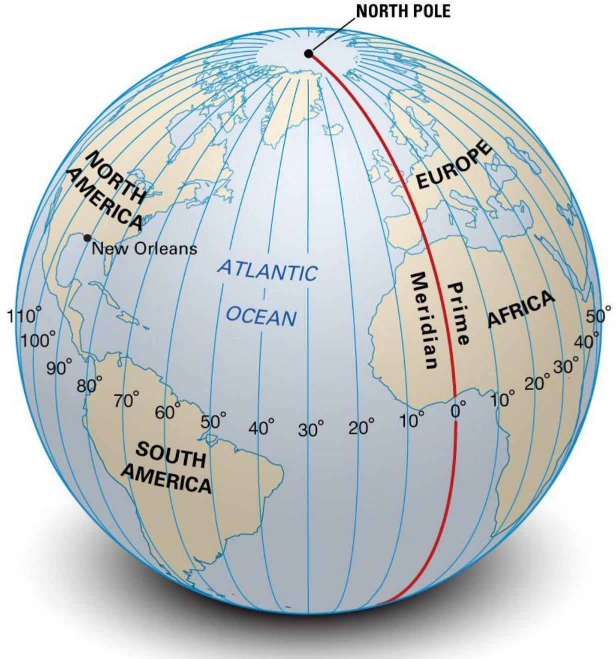


Geographic coordinates: (latitude, longitude)

- Latitude: ° above/below equator (top right)
- Longitude: ° east/west from prime meridian (bottom right)

We are currently at (55.871421,-4.292324)

- 55.8° above the equator
- 4.3° west of Greenwich
  - Greenwich Meridian Line is used as Prime Meridian (i.e., 0°)
- (I simplified these, but you need all digits – Earth is pretty big)



GNSS Accuracy

Accuracy varies

- Best case: 10cm (BeiDou), 20cm (Galileo), 30cm (GPS)
- Average use case: 1-3m

Device location is determined via multiple satellite signals

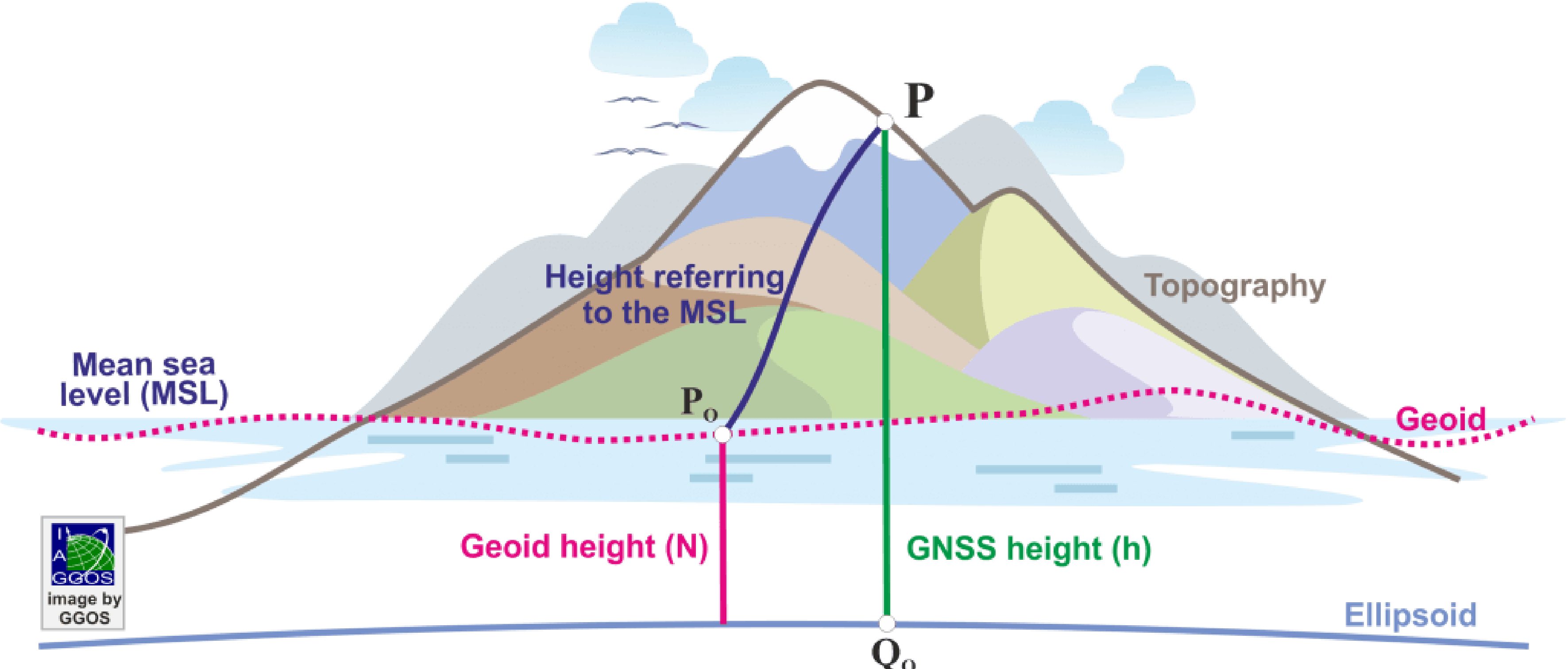
- With ‘line of sight’
- Performance varies based on position, surroundings, atmosphere, etc
  - Generally worse in urban areas with tall buildings – signals get blocked or reflected

Can accurately estimate outdoor ‘location’ for *most* applications

GNSS Elevation

Height above surface

- Often referred to ‘above sea level’ (but the sea isn’t level)
- GNSS uses a reference ellipsoid and geoid



Other Elevation Sources

Many mobile devices have barometric pressure sensors

- Directly measure elevation via ambient air pressure
- Especially useful when GNSS is imprecise
  - e.g., climbing, skiing, mountain biking – where moving a few metres in any direction changes elevation significantly

Elevation basemaps

- Estimates elevation from topographical data using geographic coords
- e.g., Ordnance Survey, USGS
- e.g., Strava crowd-sources from user uploads with barometric pressure

Access Point Trilateration

Cross-referencing with network access point locations

- e.g., trilateration from known position of cellular signal towers
  - In conjunction with other peoples' known positions with similar signal strengths
- e.g., estimated location of Wi-Fi access points
  - Android and Apple periodically report GNSS, visible devices and signal strengths
    - A significant 'Wi-Fi Positioning System' dataset
- Typical performance 5-15m accuracy

Every time you go somewhere, you're contributing to this dataset

- When people pass your home, the location of your AP gets updated too

Indoor Positioning

Geographic positioning systems do not work well indoors

- Limited line of sight to satellites, signal occlusion and interference, etc

Access point trilateration may be feasible

- Similar resolution to outdoors, though
- Signal strength is affected by posture, people, furniture, other devices, etc
  - e.g., I measured -33 dBm to -48 dBm sitting on my couch, just moving my phone

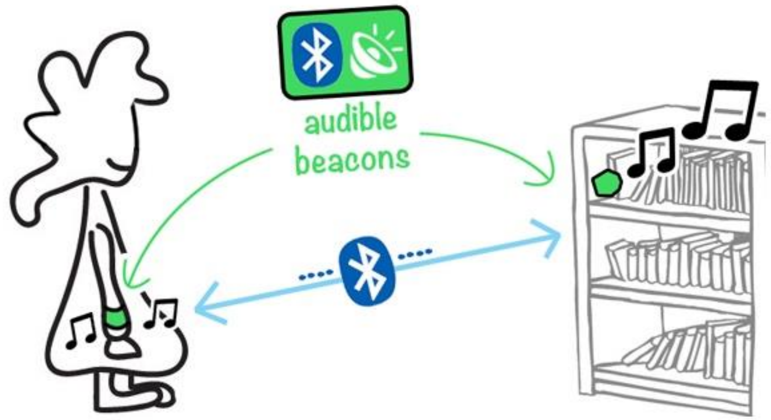
Indoor Positioning via Proximity

Absolution positioning is difficult

- i.e., your exact location with respect to a frame of reference

Indoor position can be estimated via proximity to other devices

- i.e., relative position in a building, not necessarily absolute position



Indoor Positioning Beacons/Tags

Small emitters that use Bluetooth or UWB radio

- Unaware of their own location
- Mainly intended for proximity-based interactions
  - e.g., *this* functionality only triggers near *this* beacon
- Whilst not intended for *precise* positioning…
  - e.g., Apple iBeacon: “immediate”, “near”, ”far”
- … can be used for localisation
  - e.g., if I had an emitter on my office door, you could navigate to me using proximity measurements
  - e.g., if everyone had an emitter on their door, you could potentially navigate turn-by-turn using trilateration

Indoor Beacon Measurements – RSSI

Relative Signal Strength Indicator (RSSI)

- Measures the power of a received radio signal
- Assumes a consistent attenuation
  - i.e., the further a signal travels, the weaker it becomes
- Can be used to determine proximity to an emitter
  - Based on established relationship between RSSI and distance

Time of Flight (ToF)

Angle of Arrival (AoA)

Indoor Beacon Measurements – ToF

Relative Signal Strength Indicator (RSSI)

Time of Flight (ToF)

- Measures duration between signal emission and reception
- Reliable relationship between distance, speed and time
  - Radio waves travel at a known speed
- Can be used to determine proximity to an emitter
  - Based on the above assumption

Angle of Arrival (AoA)

Indoor Beacon Measurements – AoA

Relative Signal Strength Indicator (RSSI)

Time of Flight (ToF)

Angle of Arrival (AoA)

- Measures signal reception at the distinct elements in the antenna
- Can be used to determine angle relative to a known emitter
  - i.e., based on the direction of the received wave
- In conjunction with approximate distance, can infer position

Bluetooth Performance

Signals are occluded and reflected by people + objects

Most devices use RSSI

- Noisy measurement of proximity – less accurate localisation
- Approximately 2-5m

Updated standards support AoA

- Support for improved accuracy but still gaining adoption
- Approximately 10-50cm

UWB Performance

Signals can pass through solid objects (including walls)

Uses ToF for accurate proximity (and reasonable localisation)

- Generally more precise than Bluetooth
- Approximately 10-50cm

Limited hardware support and adoption

- All iPhones have dedicated UWB chips (U1 from 2019, U2 from 2023)
- Limited number of high-end Android devices support UWB

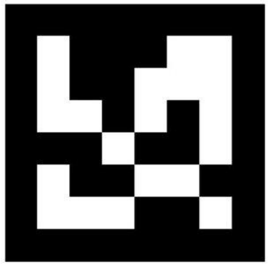
Fiducial Marker Tracking

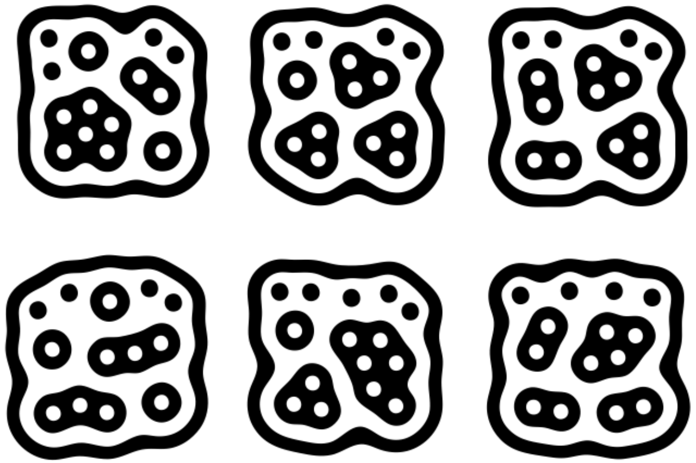
Optical detection of structured markers

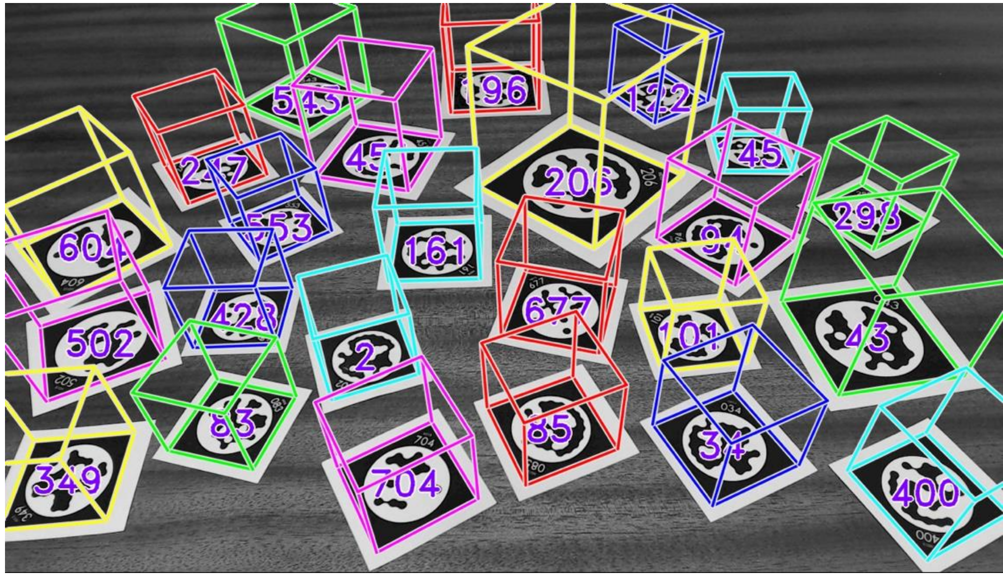
- Known size and lack of symmetry ⇒ relative distance and orientation
- Unique designs that are robust against scale and orientation variation
  - i.e., can be uniquely identified from many directions even when image clarity is limited

Examples include…

- ARTag (top left), reacTIVision (top right), STag (bottom), QR codes…







Fiducial Marker Performance

Accuracy and reliability varies with platform

- Affected by lens, sensor, image processing, sample rate
- Image quality affected by lighting conditions
- Marker design affects accuracy – some better than others
  - e.g., QR codes less suitable for localisation – too far away to decipher

Requires direct line of sight

- Fine if you're sensing in the right direction…
- Accuracy varies with distance and angle
  - Counter-intuitively, looking 'straight on' is not always the best

Activity (in your own time)

What approaches could you use for these applications (and why)?

- Helping students locate their supervisor's office
- Giving directions to departure gates in an airport
- Locating electronic components in a workshop
- Tracking number of laps completed by a runner on an outdoor track

Relative Movement

We’ve considered location as:

- Absolute position on the planet
  - e.g., (Latitude, Longitude, Elevation) via GNSS or WPS
- Relative position to known devices or markers
  - e.g., using RSSI, ToF or AoA
  - e.g., optical sensing of visual markers

We can also sense change in location without knowing location

- e.g., virtual reality headsets with inside-out tracking
  - We’ll see more of this in the Mixed Reality units of the course

Movement Sensing

Motion sensors are not suitable for localisation

- Inertial measurement units detect acceleration, orientation, etc
- But cannot reliably determine change in position due to drift
  - Continuous change in sensitivity and measurement characteristics
  - e.g., accelerometers are still impacted by manufacturing stress for several months

Optical sensors can more reliably estimate relative movement

- Simultaneous Localisation and Mapping (SLAM)
  - Localisation: figuring out where ‘I’ am
  - Mapping: figuring out the environment ‘I’ am in

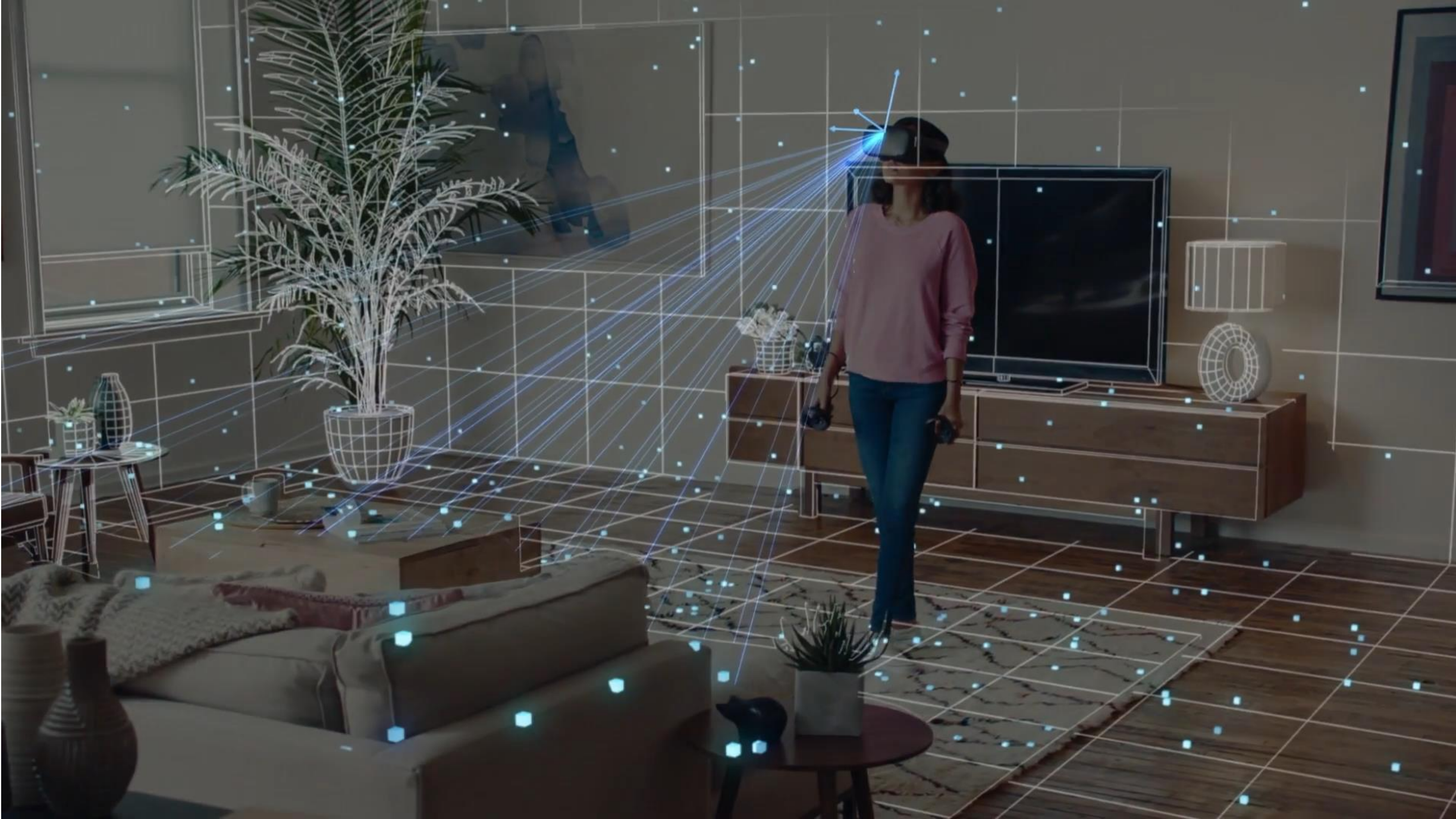
Visual SLAM

Uses optical sensors to detect features in the environment

- Including cameras, depth sensors, LIDAR
- Change in feature position builds up a model of physical space
  - Think of motion parallax: things that are closer appear to move faster
- And change in position within that space

Examples on next two slides:

- Google Project Tango
- Oculus Insight



SLAM Frame of Reference

SLAM tracking uses the device as the frame of reference

- Relative to self – i.e., egocentric
- Looking outwards to see what is there…
- … and discovering own motion based on how environment changes

Could fuse with other sensing for absolute frame of reference

- e.g., markers with known absolute position in the world
- e.g., other forms of location sensing

Room-scale Sensing

Sensors that track objects in calibrated space

- e.g., motion capture systems use networks of optical sensors
- e.g., depth cameras can sense objects within field of view
- e.g., “outside-in” VR sensors (e.g., Vive Lighthouse)

Better suited to spaces where sensing is unobtrusive

- e.g., using VR headset for playing games at home
- e.g., in environments where sensors can be installed permanently

In practice, ‘location’ used to detect movements in virtual space

- Rather than widespread use of bespoke room-specific applications

Location Sensing Summary

Different sensing requirements and performance trade-offs

- GNSS chip, internet access, radio emitters, optical sensors, etc

Different strengths and weaknesses

- ‘Best’ choice is application specific
  - e.g., GNSS for outdoor exercise tracking – simple and sufficiently accurate
  - e.g., beacons for indoor navigation – feasible, low resource requirement
  - e.g., visual SLAM for VR gaming – fast and reliable sensing, resource less relevant
  - e.g., outside-in sensing – when designing for a very particular space
- Key point: can you achieve sufficient accuracy to be functional?

Requesting Location Data

Consider trade-off between update rate and battery

- How often is updated location necessary?
  - e.g., every 1 second for recording mountain biking activities
  - e.g., every 5 seconds for pedestrian navigation
  - e.g., every 2 minutes for checking if I've entered a 200m geofence around home
  - e.g., every 1 hour for location-based weather

Some devices restrict rate or have preset rates

- e.g., sports wearables typically range from 1 Hz to 0.1 Hz for GNSS

Platforms enforce sensible rate limits

- e.g., Android restricts update rate for apps not being actively used

Processing Location Data

Inherent variability and noise in location estimates

- Even when you are standing still, your location probably doesn't stay still
  - i.e., jumps around
- Even when you move in a straight line, your location probably doesn't
  - i.e., tends to zig-zag, deviate from paths or roads

Signal processing can improve quality

- e.g., applying heuristics to remove outliers or improbable movements
- e.g., applying smoothing filters or moving averages to reduce variance
- e.g., fusing with other data to improve confidence (example next slide)

Google FLP

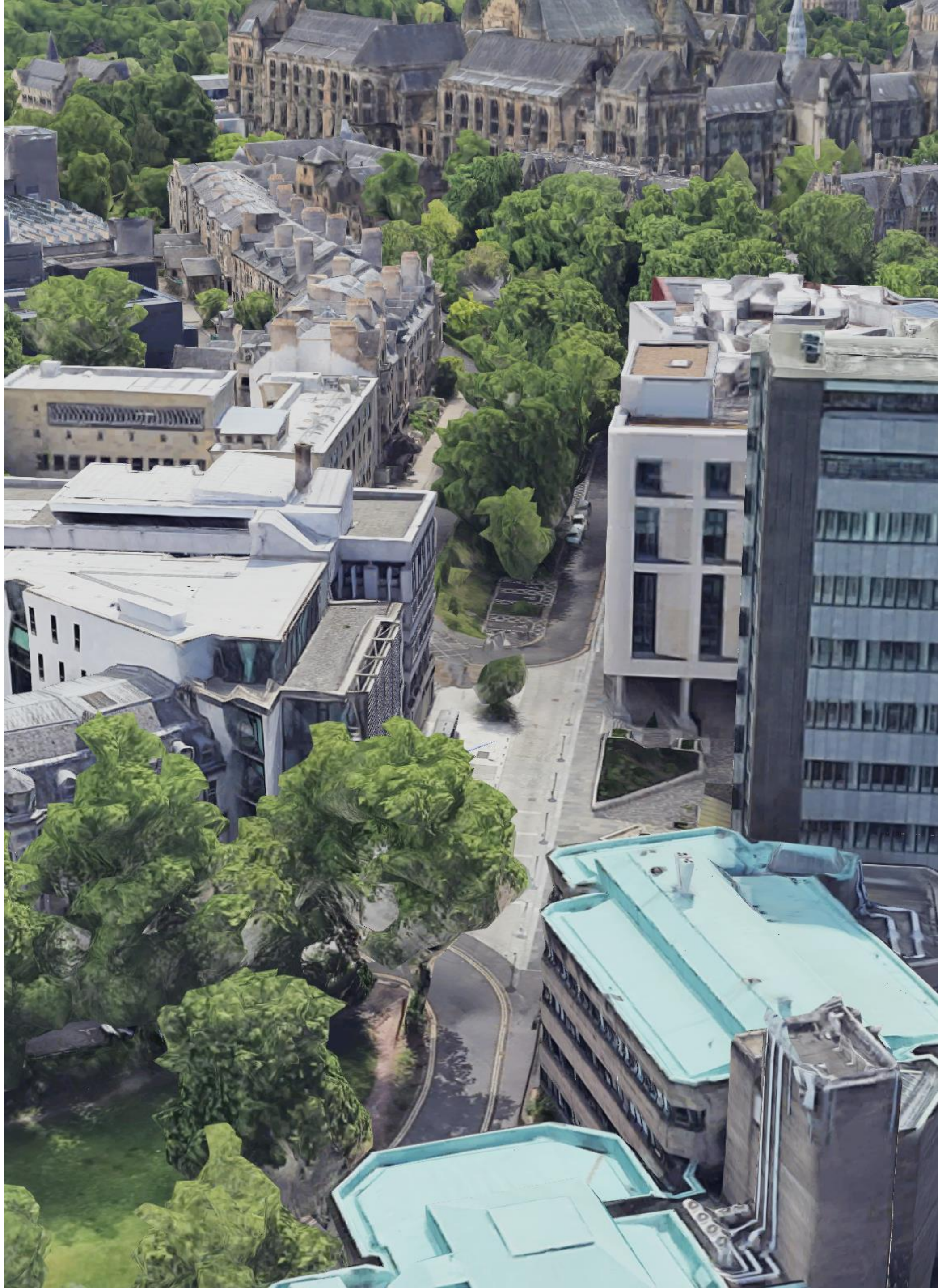
Some services, e.g., FLP, do multimodal signal processing for you

Buildings block satellite ‘line of sight’

- Device receives reflections instead
- Which throw off calculations

Google (right) use photogrammetry to create 3D models of urban areas

- Applies corrections based on predicted interference from tall buildings



Simple Filtering Methods

Moving average

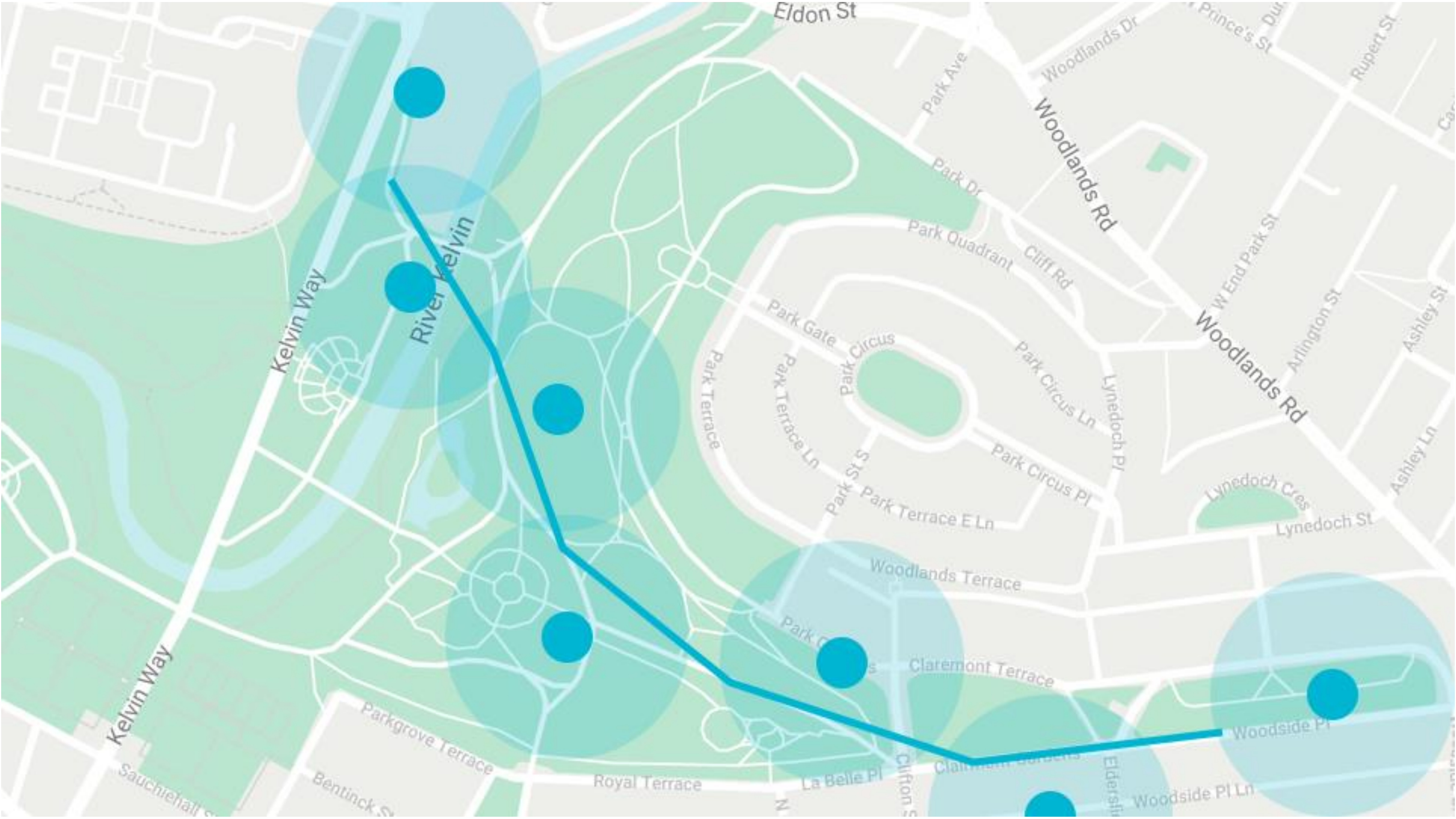
- Smooths signal by taking average of previous *n* samples
- Introduces latency as a linear function of ‘window size’ (*n*)

Weighted moving average

- Applies weighting to a moving average, rather than equal weighting
- e.g., so that recent samples affect average more than older samples

Exponential weighted moving average

- A subset of WMA where newer samples are exponentially weighted
  - Potentially reducing latency by creating better impulse response



Eldon St
Woodlands Dr
Prince's St
Rupert St
Park Ave
Woodlands Rd
Park Dr
Park Quadrant
Cliff Rd
W End Park St
Arlington St
Ashley St
River Kelvin
Kelvin Way
Park Gate
Park Circus
Park Terrace
Park Terrace Ln
Park Circus Ln
Lynedoch Pl
Park Circus Pl
Park St S
Park Terrace E Ln
Lynedoch Cres
Lynedoch St
Ashley Ln
Woodlands Terrace
Claremont Terrace
Woodside Pl
Parkgrove Terrace
Royal Terrace
La Belle Pl
Clifton St
Woodside Pl Ln
Bentinck St
Sauchiehall St

Kalman Filtering

Commonly used with ‘fast’ movements at low sample rates

- Fuses data with predictions about future expected movements
- Sometimes incorporates inertial measurement data

Initially developed for automotive applications

- Cars tend to make smooth movements on mostly straight trajectories
- Use this assumption to correct subtle deviations in GNSS

You can make application-specific assumptions

- e.g., what kinds of movements are expected if user is cycling?

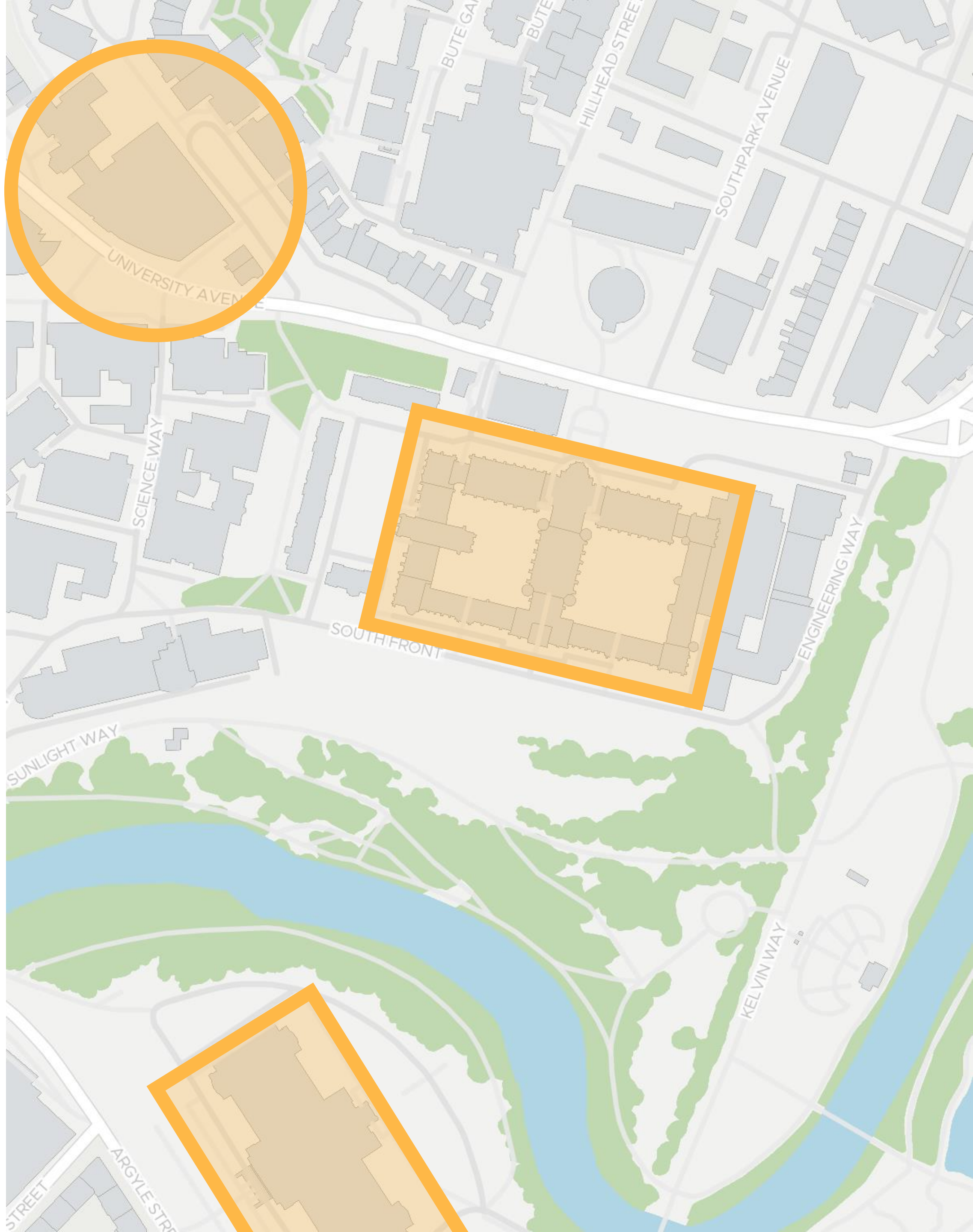
Interaction Techniques

Proximity

- Functionality based on being near points-of-interest
- e.g., if distance < 10m ...

Geofencing

- Functionality based on being within a defined region
  - inside a polygon or within a circle around a point-of-interest
- e.g., if inside(poi) ...



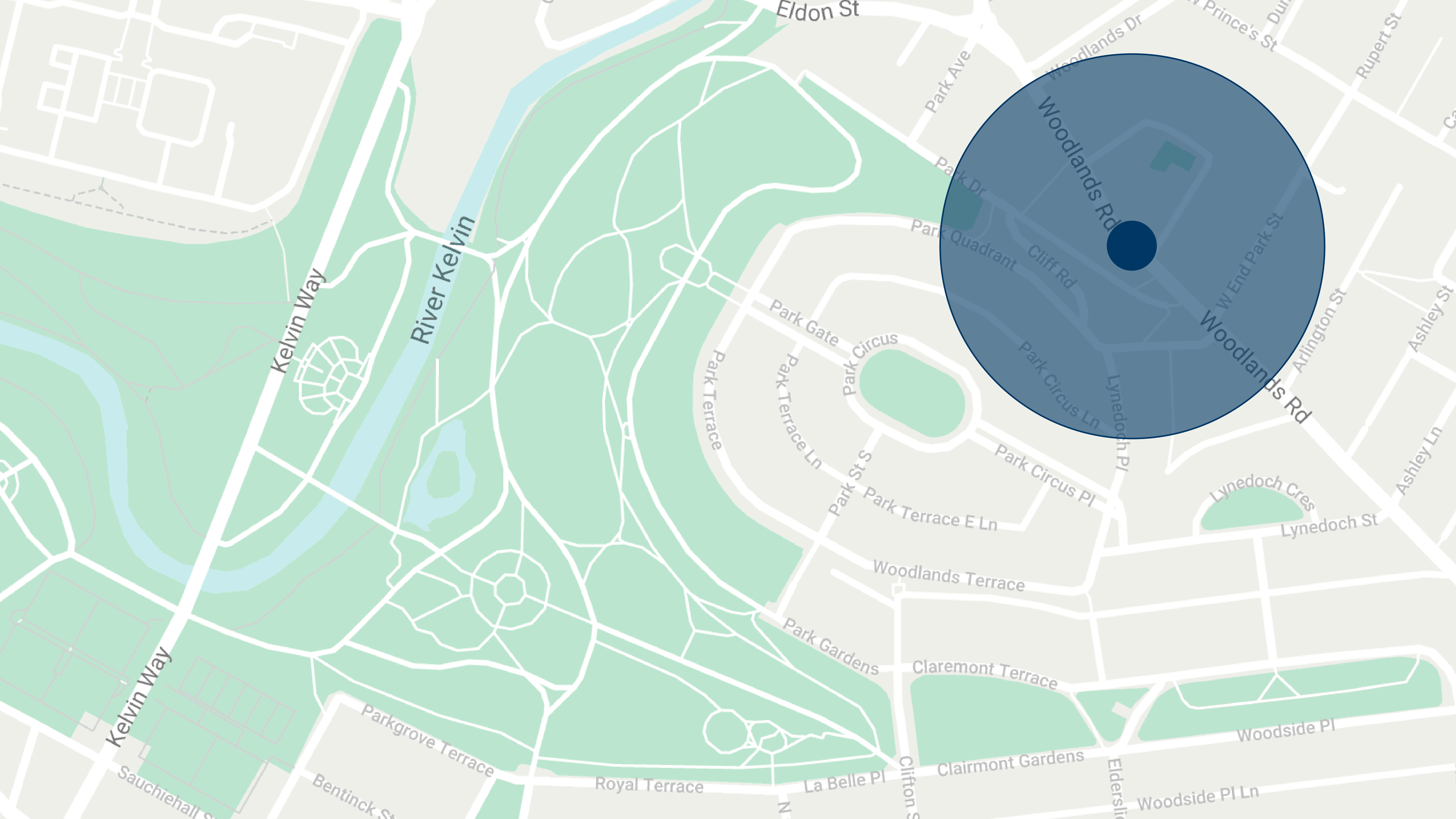
Unit 6 Lab Exercise

Uses proximity and geofencing

- Proximity triggers when user position is within n metres of POI
- Geofence triggers when user position is inside rectangle region
  - Simple geometry to determine if user is ‘inside’ or ‘outside’ the fence

Is not smoothing or filtering data

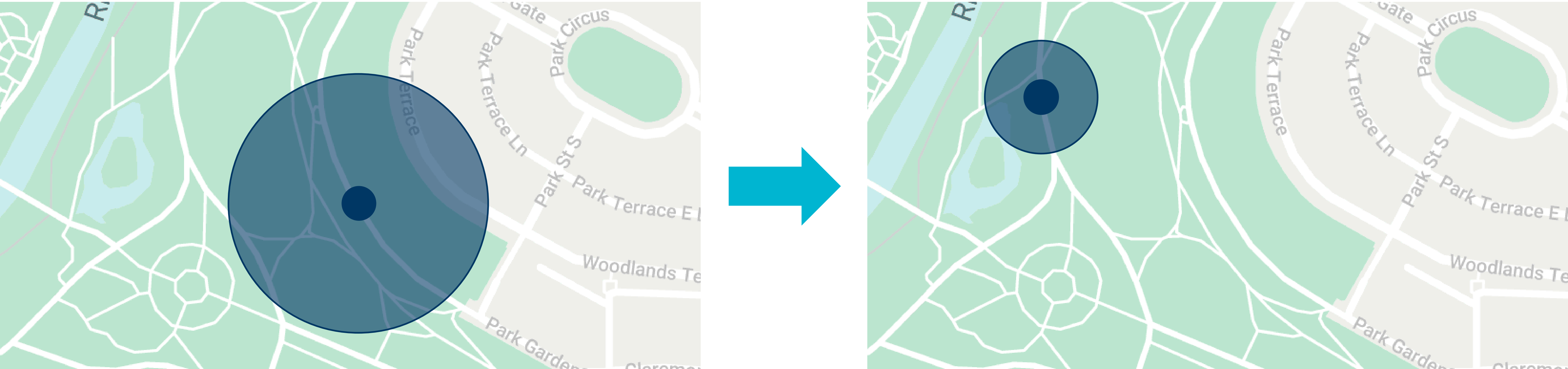
- Just uses the camera position
- Skips processing to keep it simple


Eldon St
Woodlands Dr
Prince's St
Rupert St
Park Ave
Woodlands Rd
Park Dr
Park Quadrant
Cliff Rd
W End Park St
Park Gate
Park Circus
Park Circus Ln
Lynedoch Pl
Woodlands Rd
Arlington St
Ashley St
Park Terrace
Park Terrace Ln
Park St S
Park Circus Pl
Park Terrace E Ln
Lynedoch Cres
Lynedoch St
Ashley Ln
Woodlands Terrace
Park Gardens
Claremont Terrace
Woodside Pl
Clairmont Gardens
Woodside Pl Ln
La Belle Pl
Royal Terrace
Parkgrove Terrace
Bentinck St
Kelvin Way
River Kelvin
Kelvin Way

Location Ambiguity

Reveal uncertainty through feedback:

- So users know ambiguity is present and might affect system behaviour
- So users know if their corrective actions are having a positive effect
- Why? Users are often forgiving of systems that are honest…



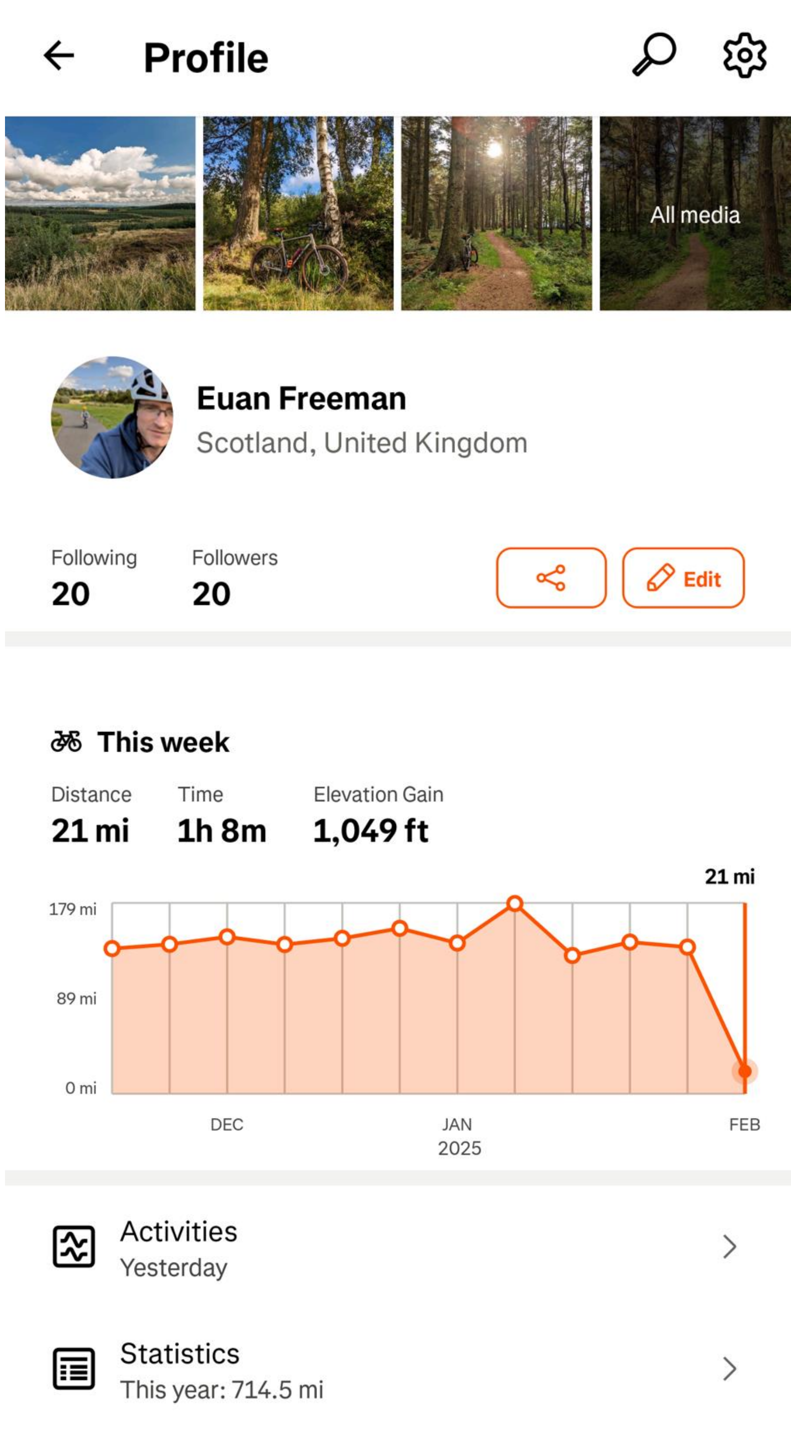
Privacy Case Study: Strava

Widely used social app for activity tracking

- Users upload primarily location data
  - But also includes physiological data, photos, etc

Many privacy concerns related to location data

- In part due to how data gets used by Strava
- And lack of user engagement with privacy controls
  - Arguably also an app's responsibility to educate its users



Strava Heatmap

Strava heatmap (www.strava.com/heatmap)

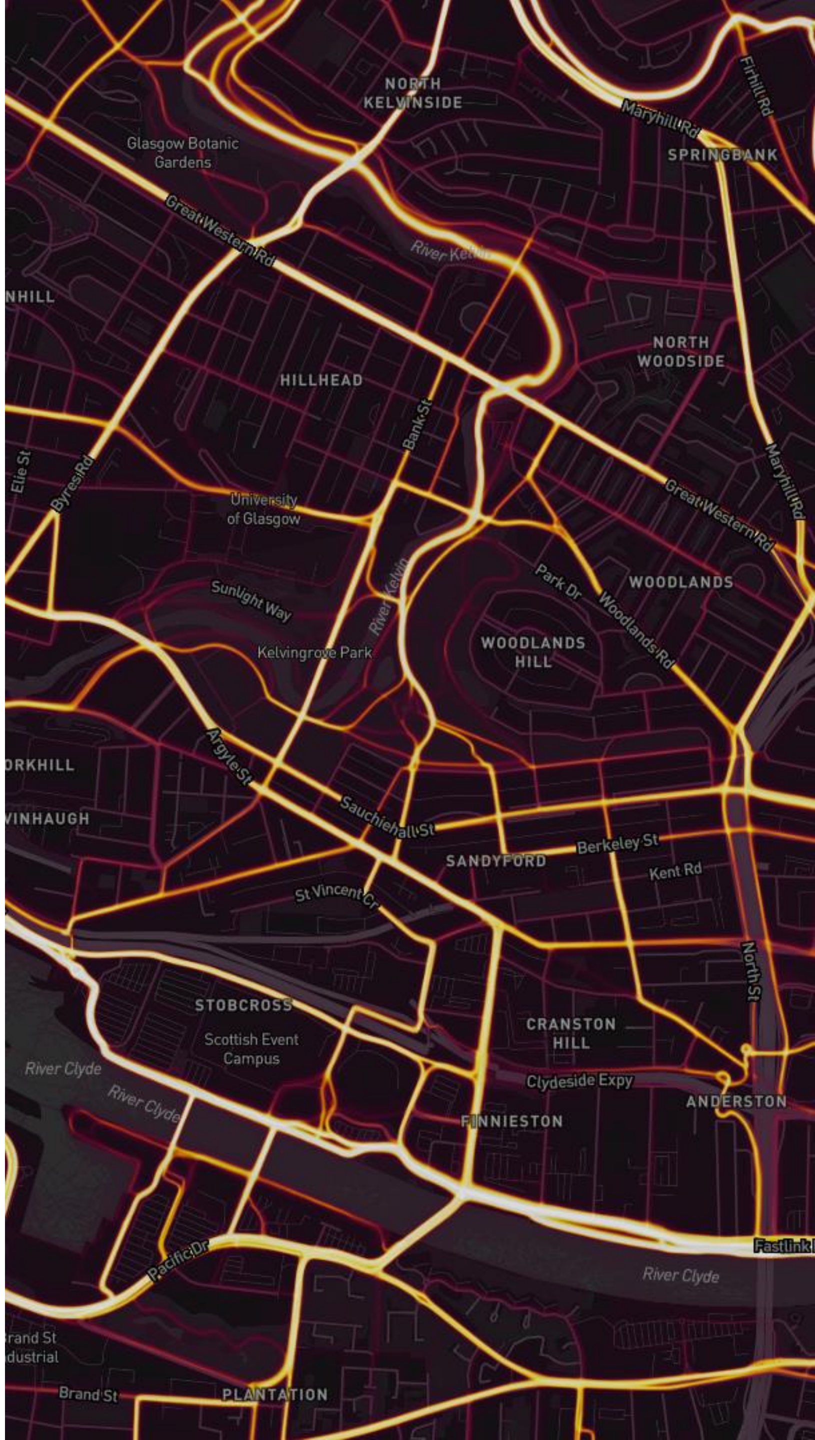
- Aggregates all activity data over long term
- Shows where its users are riding and running

Useful for finding popular cycling routes

- Especially in unfamiliar areas – e.g., abroad

Integrated into route creation tools:

- Prioritises roads where most people ride or run



Aggregated Data Concerns

Individual users/households identifiable

- Especially in lower density activity areas

Revealing users' homes

- ... to bike thieves?

Human Error

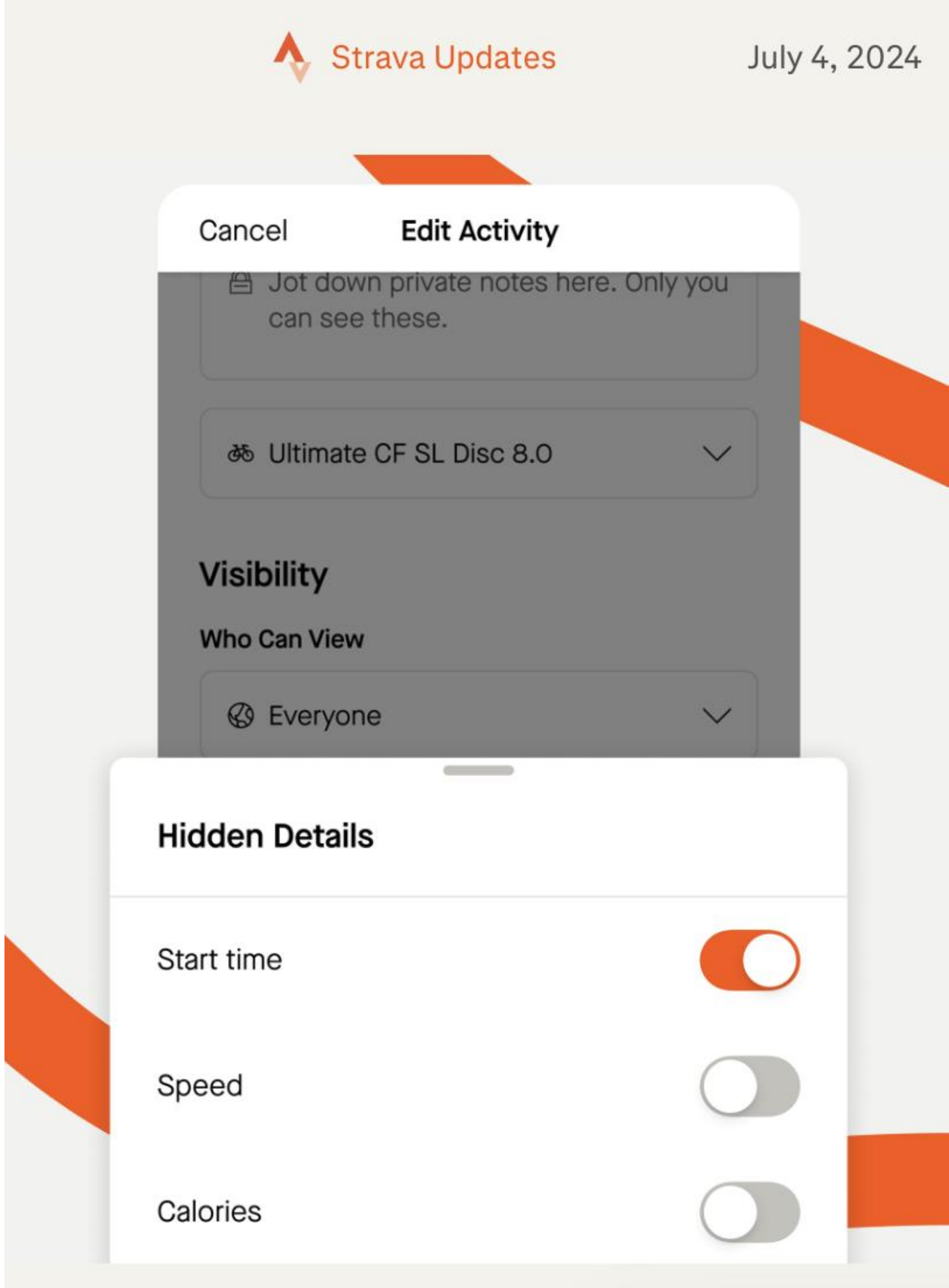
Heatmap revealed US Army base:

- Users overlooked implications of sharing
  - Or were unaware of how their data might be used
- Strava mostly used by westerners...
  - Revealing sensitive areas in places without local users

Location + Time

Concerns with timestamped location

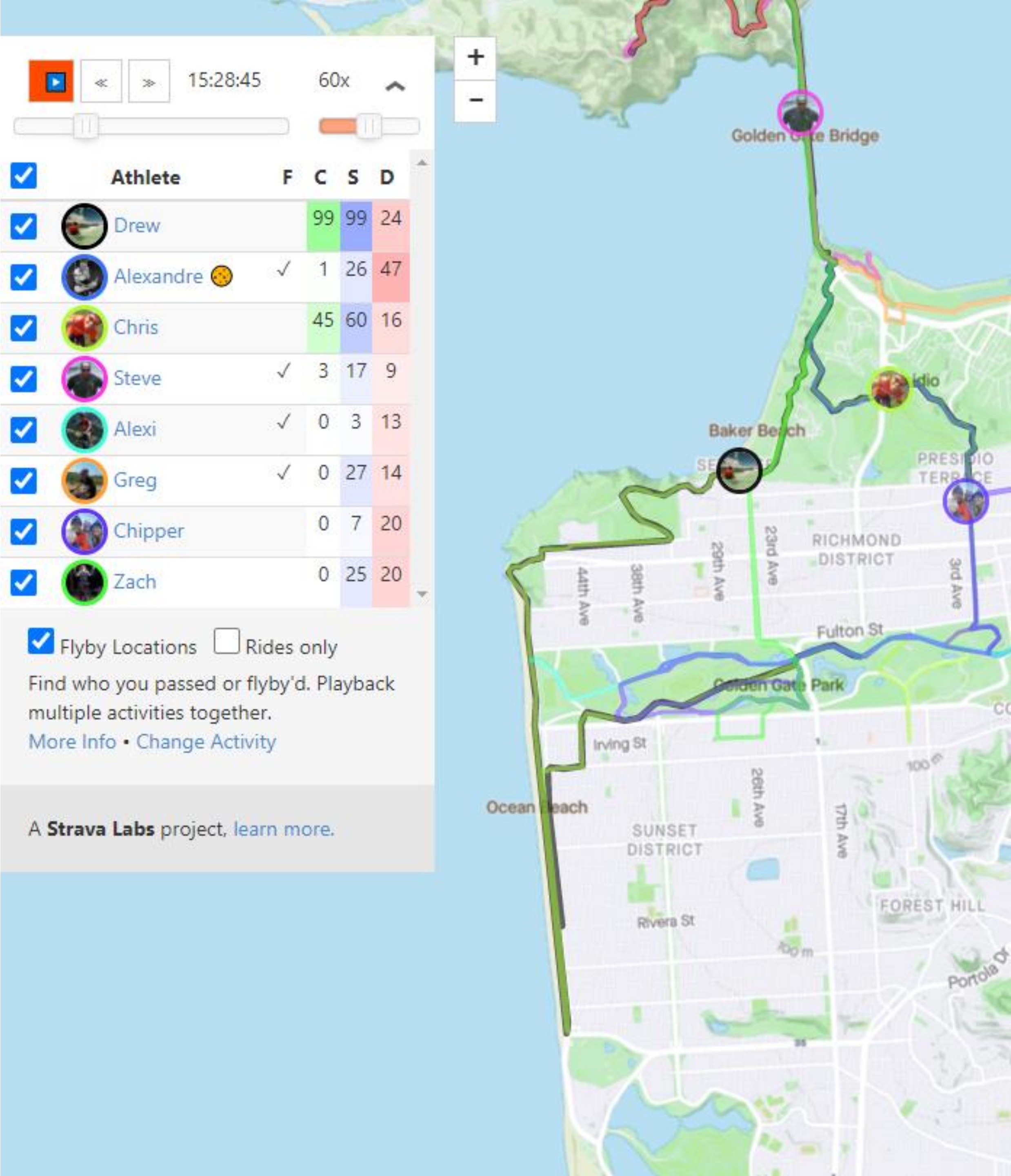
- e.g., knowing *when* you are in a place
- e.g., shows when your house is empty
- e.g., shows your habits
  - Ukrainian Intelligence Services allegedly assassinated a Russian naval commander because he uploaded his daily run to Garmin Connect
- e.g., shows when you're outside cycling during working hours
  - Oops



Unknown Features?

Strava "Fly-By":

- Users you 'flew by' on your activity
- See full route of people whose activities would otherwise not be discoverable to you



Shared Responsibility

Who is responsible for privacy?

- Users?
  - Have not engaged with, or have misunderstood, privacy settings
  - … but probably didn’t expect their data to be publicly shared!
- Strava?
  - Lack of sensible defaults: all users were opted-in by default
  - Ignored privacy settings in their aggregated data
  - Not pro-active about educating users or promoting privacy controls

Privacy Summary

Privacy violations may occur in future, not just ‘now’:

- Data creates opportunities for future privacy violation
- Violations that might not even be imaginable when features first launch

Partly due to ambiguity over data scope, use, and access:

- What data is collected?
- What else is it being used for (that users aren’t immediately aware of)?
- Who can access that data (in what format)?
- What can users do to control access to data?