Sensor-Based Interaction

Euan Freeman

euan.freeman@glasgow.ac.uk

University of Glasgow | School of Computing Science

Outline

Sensor-based interaction techniques

Case Study: touchless gesture interaction

Case Study: pressure interaction

Input recognition

Intended Learning Outcomes

Unit 2

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

Unit 4

Unit 5

Units 7 & 8

Part 1 – Sensors

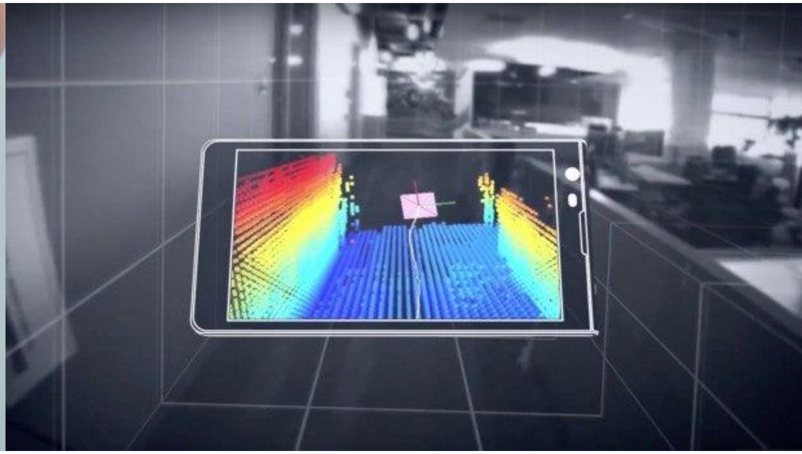
Sensing Capabilities

Today's devices have dozens of sensors:

- Accelerometer, gyroscope, magnetometer, temperature, humidity, moisture, ambient light, proximity, barometric pressure, GNSS, heart-rate, fingerprint, eye trackers, iris scanners, radar, LIDAR, depth, pressure, etc.

optical heart-rate

fingerprint

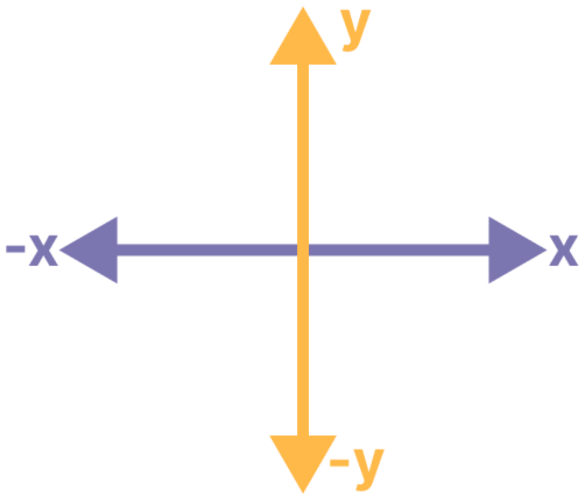


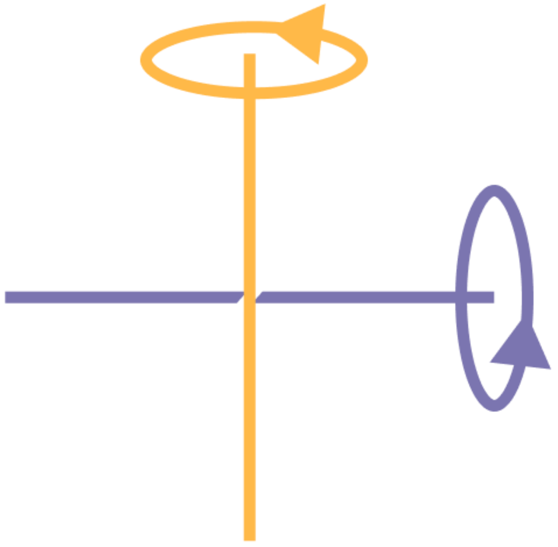
depth

Sensing Capabilities – Inertial Motion

Inertial Measurement Unit (IMU):

- Accelerometer: translation
- Gyroscope: rotation
- Magnetometer: heading





Sensing Capabilities – Surroundings

Immediate surroundings:

- Ambient light: how light/dark
- Proximity: how close is the nearest ‘thing’
- Camera: what is in front of the device
- Depth: how far away is each ‘pixel’ in the camera image
- Bluetooth: what other devices are nearby

Sensing Capabilities – Position

Position ‘in the world’:

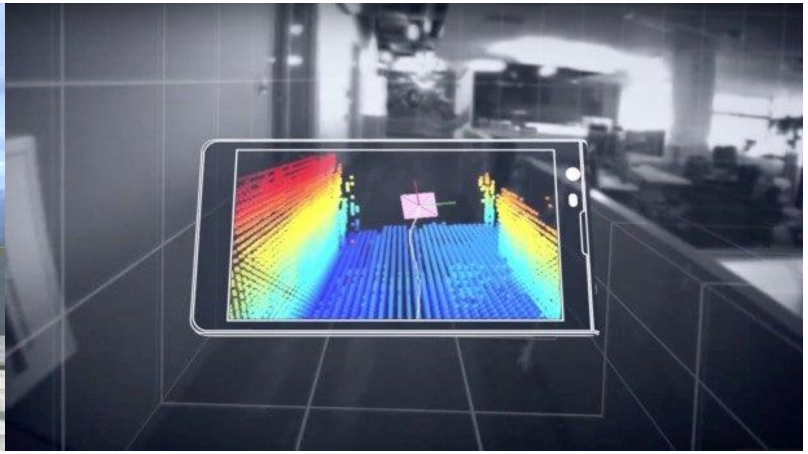
- Barometric pressure: elevation above sea level
- GNSS location: GPS, GLONASS, GALLILEO, etc
- Bluetooth, UWB: nearby devices

Sensing Capabilities – 3D

3D (e.g., objects, people, hands):

- Radar
- LIDAR
- Depth: measuring intensity of reflected infrared light
- Ultrasound: measuring intensity of reflected sound waves

‘Spatial’ for iPhone Pro (LIDAR)



Google Project Tango (depth)

Even More Sensing

Fundamental device components are also interaction sensors:

- Microphones: ambient audio detection, speech recognition
- Cameras: object recognition, hand tracking, eye tracking
- Touchscreens: touch gestures, pre-touch sensing, finger pose

Touchscreens can detect more than just touch:

- Typically provide (x, y) coordinate for point of contact
- Sensors actually detect the full contact area
- … and fingers up to 5cm above the screen

Sensor Fusion

Sensors can do useful things in isolation...

But are especially powerful when ‘fused’ with other sensor data:

- e.g., activity detection from accelerometer and gyroscope
- e.g., gaze detection from front camera and depth sensor
- e.g., indoor mapping from gyroscope and depth camera
  - Next slide shows an example from Google Project Tango


Google
Mapping
Common App
Unity Demos
Camera Trick
Walking Dem
Utilities

Sensor-Based Interaction

Lots of research about using sensors for new interactions:

- i.e., sensor-based interaction techniques

Examples:

- Speech input (e.g., Siri, Alexa, Ok Google)
- Mid-air gesture input (e.g., Google Motion Sense, many camera apps)
- 3D depth sensor input (e.g., Google Project Tango, iPad Pro, VR headsets)
- Grasp pressure input (e.g., Google Active Edge)
- Touchscreen pressure input (e.g., Apple 3D Touch)

Recap on Sensor-Based Interaction

Lots of sensors in modern mobile devices:

- Can be used for context-aware computing
- Can be used for sensor-based interaction techniques

Sensor fusion ‘greater than the sum of its parts’:

- Multimodal sensing gives a more accurate estimation of complex actions
- Leading to more expressive interaction techniques

Part 2 – Case Study: Touchless Gestures

Touchless Gesture Sensing

Mid-air gestures are hand movements or poses in air;

- e.g., swiping, waving, finger snapping, thumbs up;

What are gestures for?

- Quick low-effort interaction: e.g., skip song (Google Motion Sense)
- Needs less precision than touch: e.g., wave anywhere vs tap buttons
- Larger input space: e.g., small watch screen vs wider space around it
- When you can't touch or reach the screen: e.g., cooking, eating, driving
- Extra input states: e.g., 'hover' content previews (Samsung Air View)
- More expressive and dextrous

Touchless Gesture Sensing

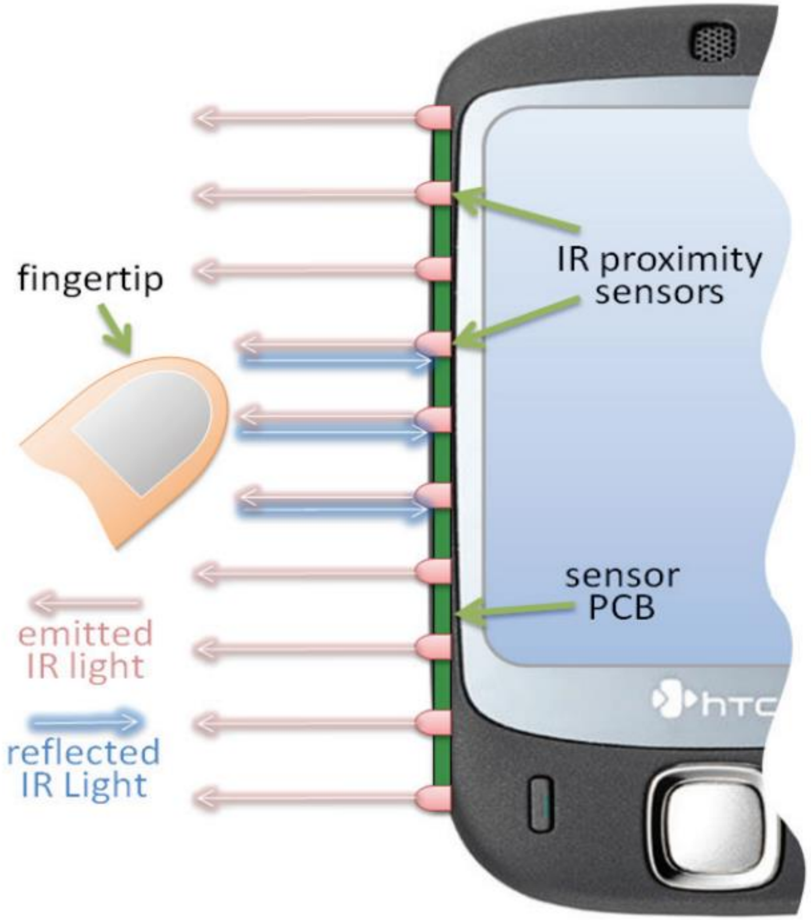
Requires ability to sense hands above the phone:

- Depth sensors
  - Common on recent phones (e.g., face detection, dynamic camera focus)
  - + proper 3D sensing; - high battery use
- Cameras
  - Using computer vision techniques for detecting hands
  - + ubiquitous; - high battery use; - only 2D sensing
- Proximity sensors
  - Infrared ‘one pixel’ sensors – mostly used to determine if phone held to head
  - + low battery use; - low resolution tracking
- Radar sensing
  - e.g., Project Soli
  - + efficient sensing

1D Sensing with Proximity Sensors

SideSight: multi ‘touch’ interaction around

- Alex Butler et al.
- Proceedings of UIST 2008
- Used 1D array of infrared sensors to detect f
- Users could control a ‘pointer’ on screen wit

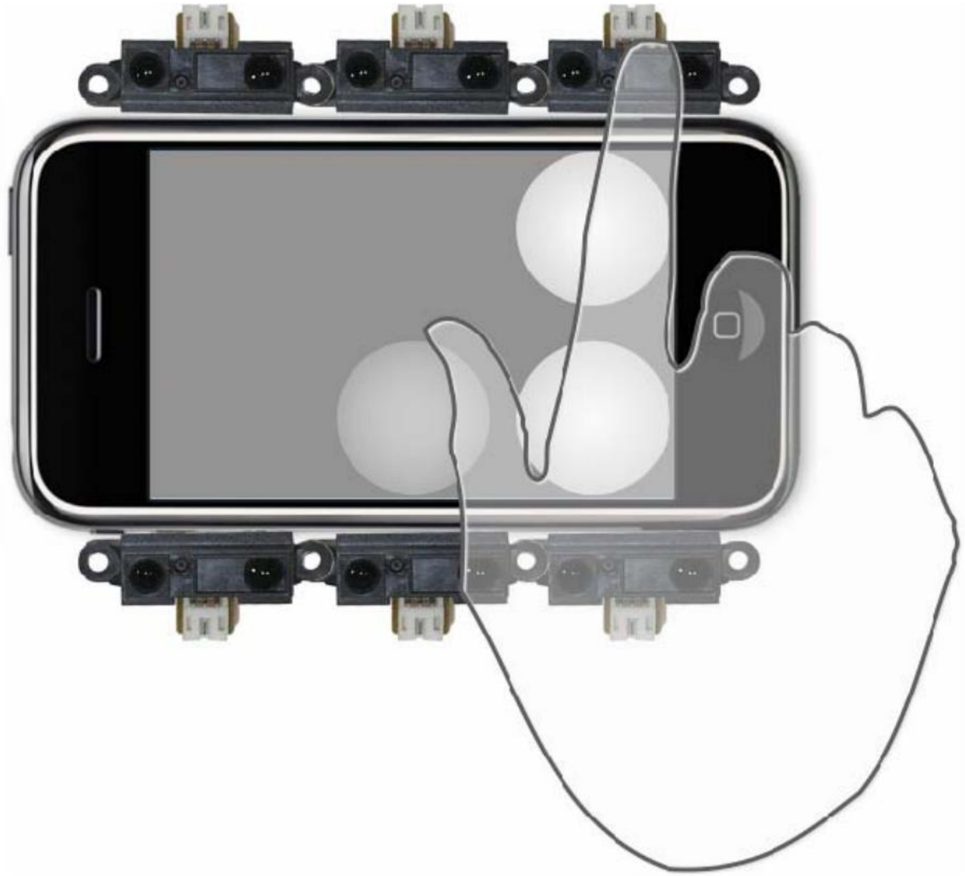


https://dl.acm.org/doi/10.1145/1449715.1449746

2D Sensing with Proximity Sensors

HoverFlow: expanding the design space of around-device interaction

- Sven Kratz et al.
- Proceedings of Mobile HCI 2009
- Used 2D array of infrared sensors to detect gestures above the screen.
- Users could swipe to skip songs.

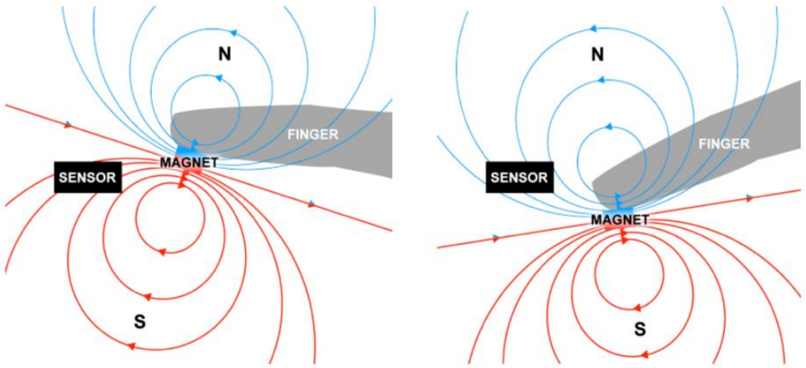


https://dl.acm.org/doi/10.1145/1613858.1613864

2D Sensing with Magnetic Field Sensor

Abracadabra: Wireless, High-Precision, and Unpowered Finger Input for Very Small Mobile Devices

- Chris Harrison et al.
- Proceedings of UIST 2009
- Used watch magnetometer to sense movements from a magnetic ring worn on the finger.



https://dl.acm.org/doi/10.1145/1622176.1622199

3D Sensing with Radar

Google “Motion Sense” in the Pixel 4

- Uses radar to detect hand motion
- Emerged from Google’s Project Soli

Soli: ubiquitous gesture sensing with millimeter wave radar

- Jaime Lien et al.
- ACM Transactions on Graphics 35(4)

https://dl.acm.org/doi/10.1145/2897824.2925953

Touchless Interaction Challenges

Interaction challenges:

- How do users know what the gestures are?
- How do users know where to perform them?
- How do users know if they are gesturing correctly?
- How do users know how to solve problems when they don’t work?
- How do users learn to improve their gesture performance?

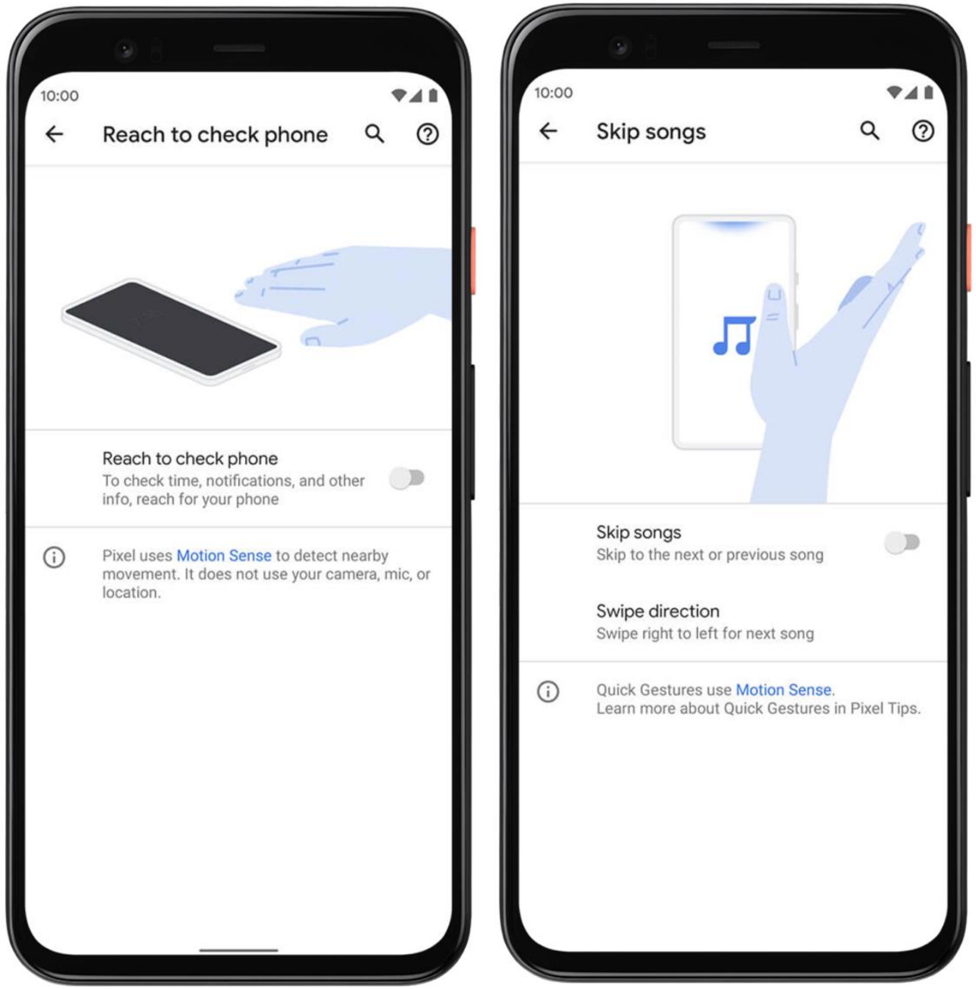
Touch is simple: you touch the screen, something happens

- Sensor interactions are unfamiliar and often ambiguous
- Designers need solutions to address these design challenges…

User Training?

‘Gated’ interactions :

- Samsung Air View, Samsung Air Gesture and Google Motion Sense need to be enabled first
- Creates opportunity to show users how to perform the gestures and tells them what they do
- Limits accidental input
- Avoids unwanted resource use



Feedback?

Give as much feedback as possible

- Visual, auditory, haptic…
- Confirm system is responding to user's actions ("showing attention")
- Visualisations of sensor data can be helpful (e.g., hand silhouette?)

Feedback in commercial examples was limited…

- Most only had functional feedback
  - When the system behaviour confirms input was correct
  - e.g., knowing the 'pause' gesture worked when the music stops
- Difficult to discover and learn without sufficient feedback!

Part 3 – Case Study: Pressure Input

Pressure Sensing

Pressure (isometric force) is an important aspect of touch:

- We skilfully use pressure when manipulating physical objects
- Pressure could be used to interact with mobile devices:
  - e.g., grip pressure (squeezing the phone)
  - e.g., touch pressure (pressing the screen)

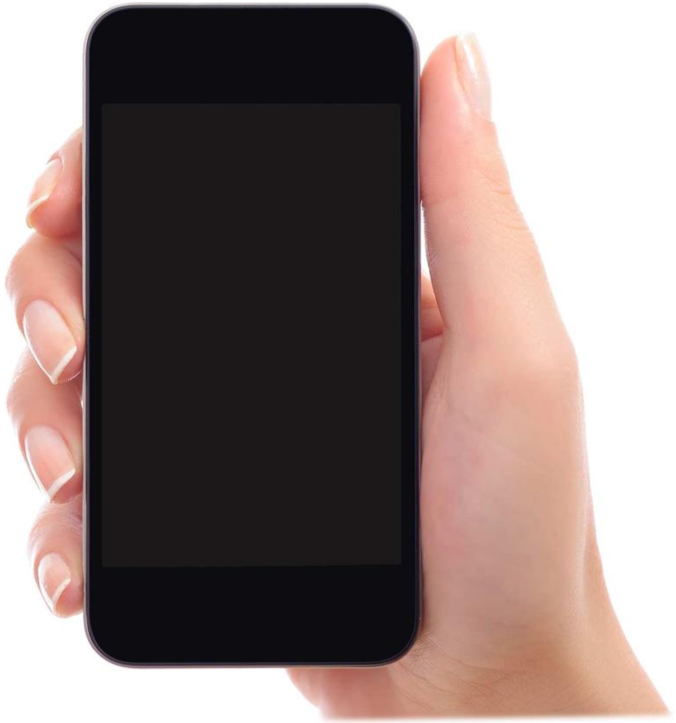
Why would you use pressure?

- Adds extra input states to touch input: e.g., ‘peek’ (Apple 3D Touch)
- Passive grip to active input: e.g., launch assistant (Google Active Edge)
- One-handed input: e.g., answer a call when phone detects grasp
- Eyes-free input: e.g., squeeze phone in pocket to ignore call

Pressure Sensing with Single FSR

Pressure-based menu selection for mobile devices

- Graham Wilson et al.
- Proceedings of Mobile HCI 2010
- Used force sensitive resistor on the edge of a smartphone.
  - Applying force varies resistance
- Apply pressure to move cursor down a list of items.

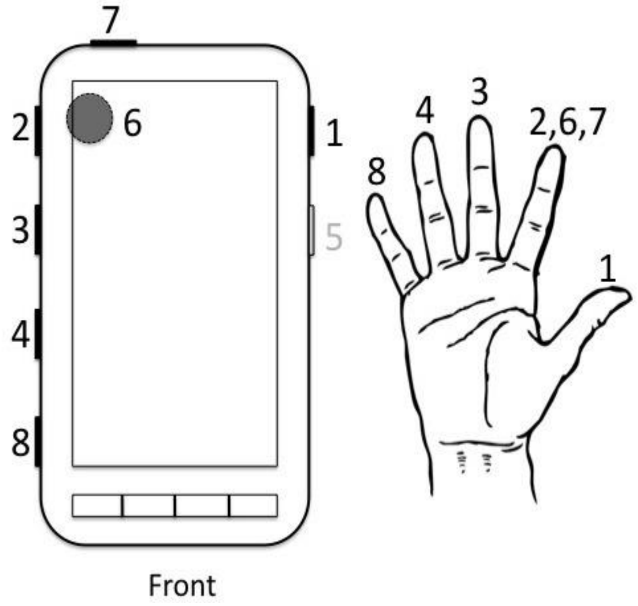


https://dl.acm.org/doi/10.1145/1851600.1851631

Pressure Sensing with Multiple FSRs

Towards utilising one-handed multi-digit pressure input

- Graham Wilson et al.
- Proceedings of CHI EA 2013
- Used many force-sensitive resistors along device.
- Explored use-cases of multi-level pressure from grip.

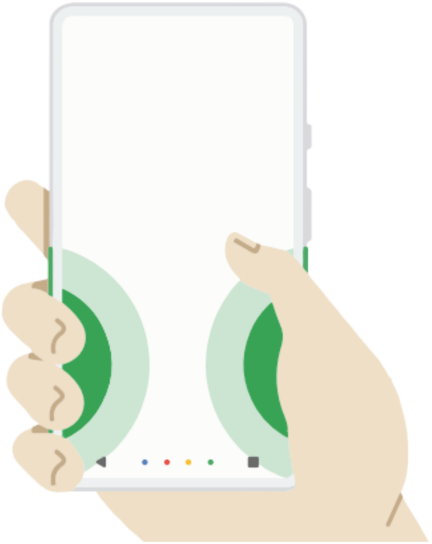


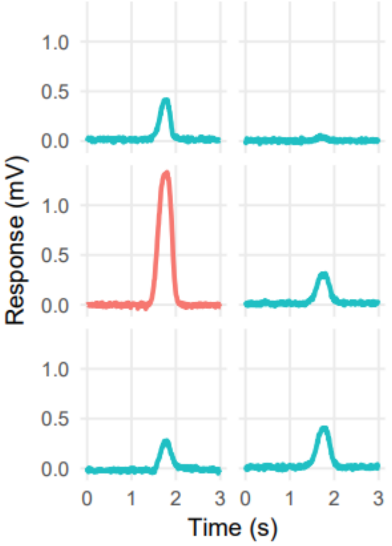
https://dl.acm.org/doi/10.1145/2468356.2468591

Pressure Sensing with FSRs and Strain Gauges

Active Edge: Designing Squeeze Gestures for the Google Pixel 2

- Philip Quinn et al.
- Proceedings of CHI 2019
- Describes the HCI research that led to "Active Edge" being implemented in the Google Pixel phones.
- Used force sensitive resistors and strain gauges to detect grip.





https://dl.acm.org/doi/10.1145/3290605.3300504

Pressure Sensing with FSR array

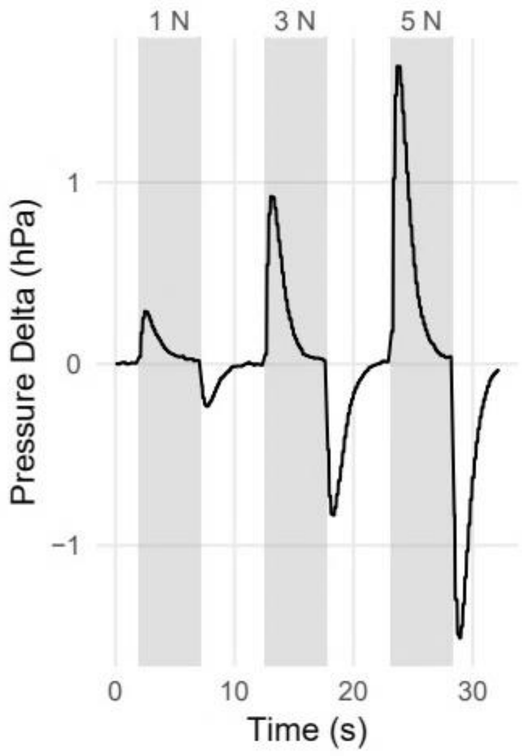
Force sensors below screen

Apple 3D Touch can detect three levels of pressure on screen

Pressure Sensing with Altitude Sensor

Estimating Touch Force with Barometric Pressure Sensors

- Philip Quinn
- Proceedings of CHI 2019
- Used barometric pressure sensor to classify touchscreen input pressure.
  - Touching the screen creates atmospheric pressure differential as air gets forced out of the device body.
- Image shows internal device pressure after applying three levels of force to screen.



https://dl.acm.org/doi/10.1145/3290605.3300919

Human and Device Capabilities

Research shows that users can skilfully modulate pressure

- e.g., Wilson et al. found pressure can be discretised to ten levels

Current implementations use fewer pressure levels

- e.g., Apple’s 3D Touch used three levels: soft, medium, hard;
- e.g., Google’s Active Edge was binary: hard to activate, or nothing;

Many sensing approaches:

- Force-sensitive resistor, strain gauge, barometric pressure, touchscreen
- And EMG from wearables?

Part 4 – Input Recognition

Implementing Sensor Interactions

Sensors provide a stream of data

- Hopefully with a high sample rate (e.g., many Hz)

Input recognition requires looking for patterns or events in data

- e.g., detecting when user has performed a ‘squeeze’ pressure gesture
- e.g., detecting when user has ‘waved’ at the device

Recognition is challenging

- Need to balance false-positive and false-negative recognition

Recognition Approaches

Approaches vary in complexity and robustness

- Thresholding
  - e.g., did a sensor reading exceed a threshold value
- Pattern matching
  - e.g., does a time-series of data match a ‘template’
- Machine learning
  - e.g., using neural networks to classify a set of features

You’ll use the first two approaches in the lab exercises

- Unit 4: using thresholds to determine when a ‘shake’ has occurred
- Unit 5: applying template rules to classify swipe direction

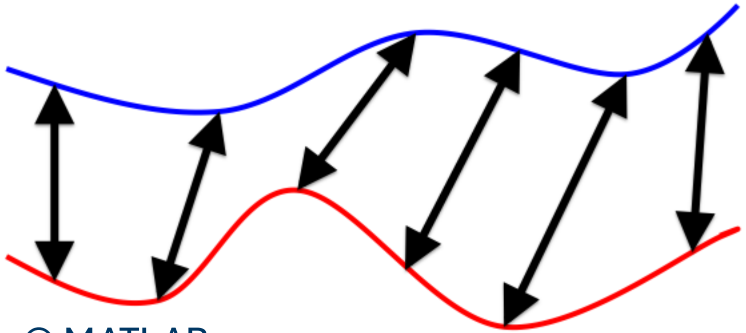
Recognition Challenges: Variability

Users performing ‘same’ actions vary in both space and time

- e.g., some users will make larger movements
- e.g., some users will move their hands quickly

Recognition approaches need to account for variability

- e.g., using scale-independent features and thresholds
- e.g., using methods like Dynamic Time Warping to adjust for time/speed



Recognition Challenges: Sensor Capability

Temporal resolution (i.e., sample rate)

- e.g., low optical frame rate ‘blurs’ fast motions

Spatial resolution

- e.g., low optical resolution less able to resolve smaller details

Degrees of freedom

- e.g., sensors lack the sensitivity to variance in actions

Recognition Challenges: Segmentation

Input recognition requires continuous time series analysis

- i.e., looking at several frames of sensor data across time

Segmentation involves identifying when an action begins and ends

- e.g., did a ‘swipe’ gesture occur in the past 240 frames of data?

Challenging because motion is complex and continuous

- e.g., think about your hand movements before and after a ‘swipe’
- Also, because actions often occur in sequence
  - e.g., three left-to-right swipes requires at least two right-to-left motions

Recognition Challenges: Training Data

Datasets do not generalise

- Individual differences in sensor technology
- Relationship between sensors and platform

Datasets need significance variance

- e.g., environmental conditions
- e.g., individual human differences

Calibration or specific user training?

- A common approach to individualise recognition

Recognition Challenges: Sensitivity

False-positive recognition

- When the system accepts an invalid input
- Overly sensitive systems prone to unintentional behaviour

False-negative recognition

- When the system does not accept a valid input
- Frustrates users and reduces their sense of agency over the system

Finding balance is tricky

- The ‘best’ bias depends on intended use case (see 2023-2024 Q3)

Summary of Input Recognition

Sensor input is not straightforward

- But lots of potential to create new ways of engaging with technology!
- Makes for fun technical work as well…

Be aware of the challenges of input recognition

- What factors might affect a particular sensor input modality?
- And how might these affect usability?
  - Relate to other issues discussed during the lecture and course

Good interaction design is cognizant of these issues

- i.e., interactions and user interfaces designed to work within system capabilities

Lecture Summary

Many sensors in modern mobile devices:

- Context sensing, interaction sensing

Sensor fusion (multimodal sensing):

- Combining sensor data for better inference

Case studies of sensor-based interactions:

- Mid-air gesture input, pressure input
- Several potential benefits, but many usability challenges

Input recognition challenges