Activity and Gesture Recognition

If everyone is moving forward together,
then success takes care of itself.

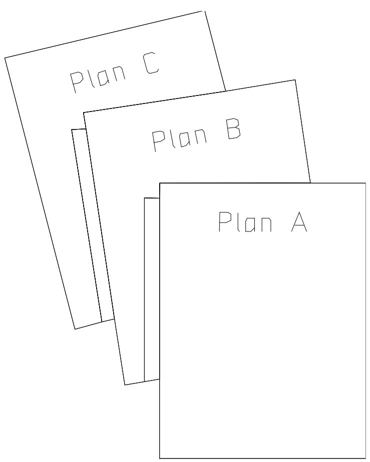
Henry Ford

Overview

- Objective
  - To understand exemplary techniques and challenges for activity and gesture recognition

- Content
  - Activity sensing and recognition (continued from Week 2)
  - Gesture sensing and recognition

- After this module, you should be able to
  - Understand the basics of activity recognition
  - Understand the basics of gesture recognition



Recap on Jigsaw

- GPS Pipeline?
  - What was the main problem for the GPS sensing pipeline?
  - What were the key idea suggested by the paper?
  - Do you agree with the author's approach?

- What do you think are the main pros and cons for the paper?

Activity Recognition

(this part is from Week 2)

Activity Recognition

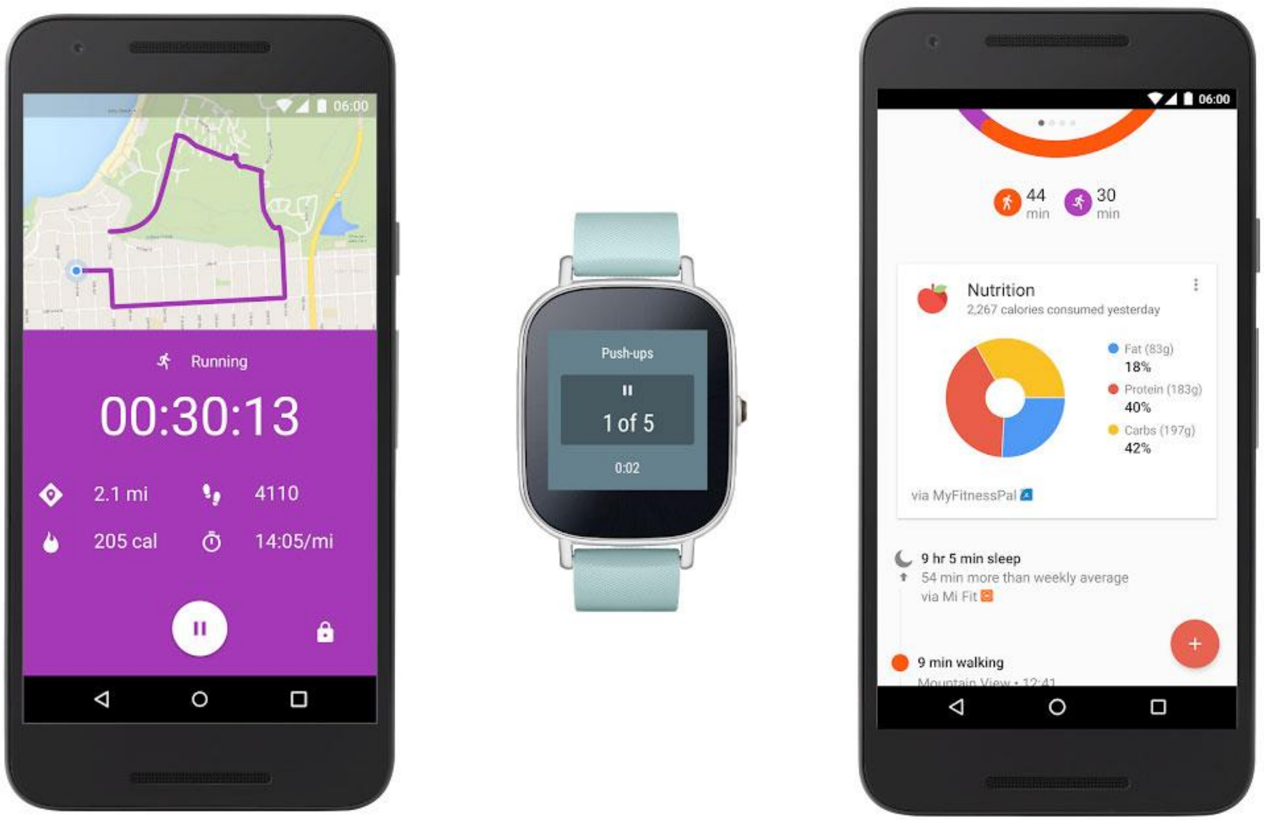
- Identifying the physical activity of a user
  - E.g., jogging, walking, sitting, standing
- Providing useful knowledge about the habits of millions of users passively—just by carrying cell phones.
- Wide range of applications
  - Activity-aware phone configuration (e.g., sending calls directly to voicemail if a user is jogging)
  - Daily/weekly activity profile for daily healthcare.

Detection vs. Classification

- Detection: Single activity type
- Classification: Multiple activity types
- Step detection vs. activity classification
- Classification usually needs a more general approach where it is harder to capture the distinguishing characteristics of each class (walking, sitting, etc.) ahead of time.

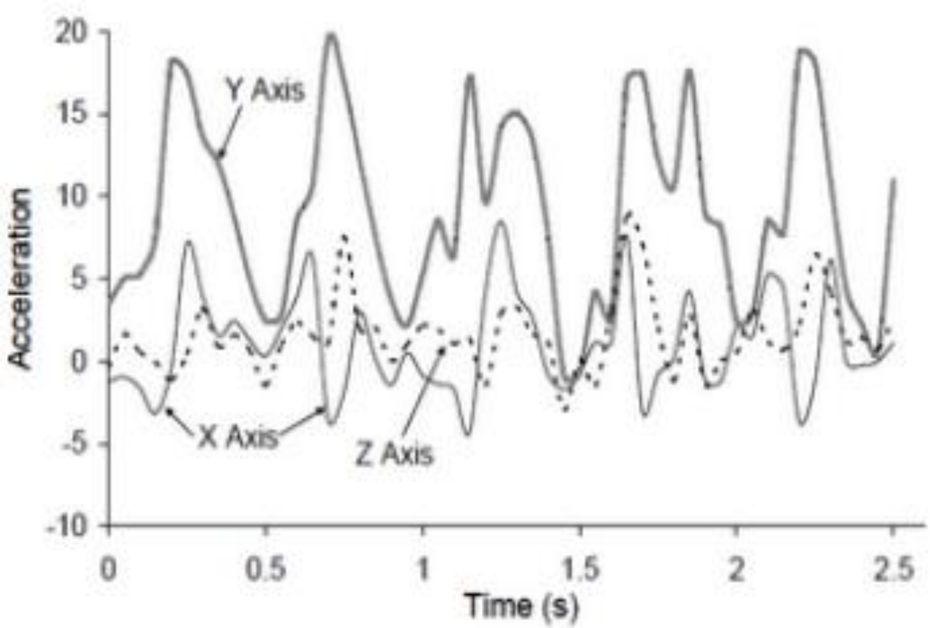
Example: Activity Tracker

- **How do we know if a person is waking or jogging?**

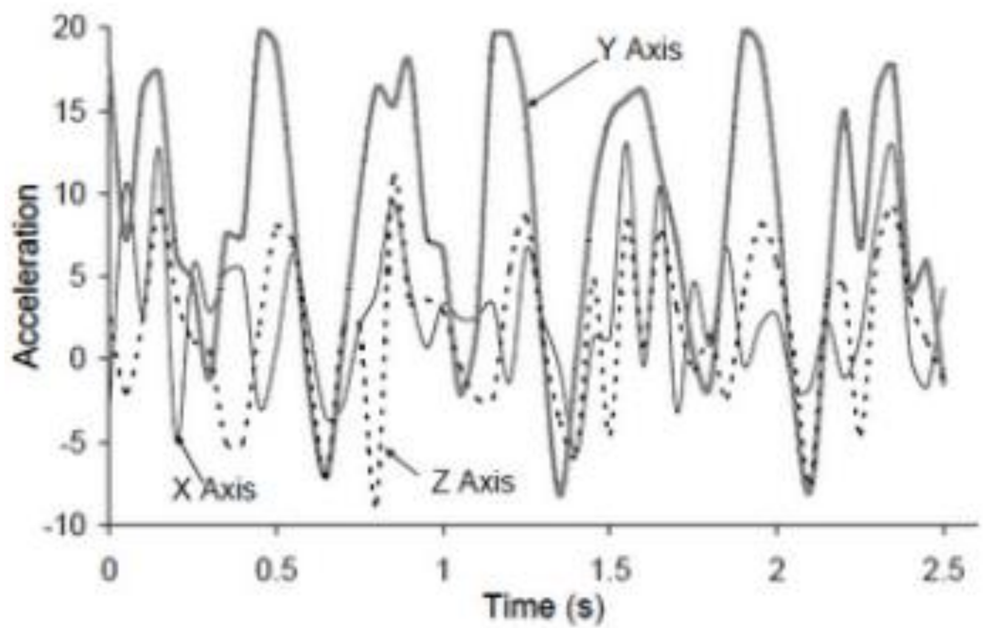


Acceleration Signals

- Assumption: Smartphone is in a front pocket.



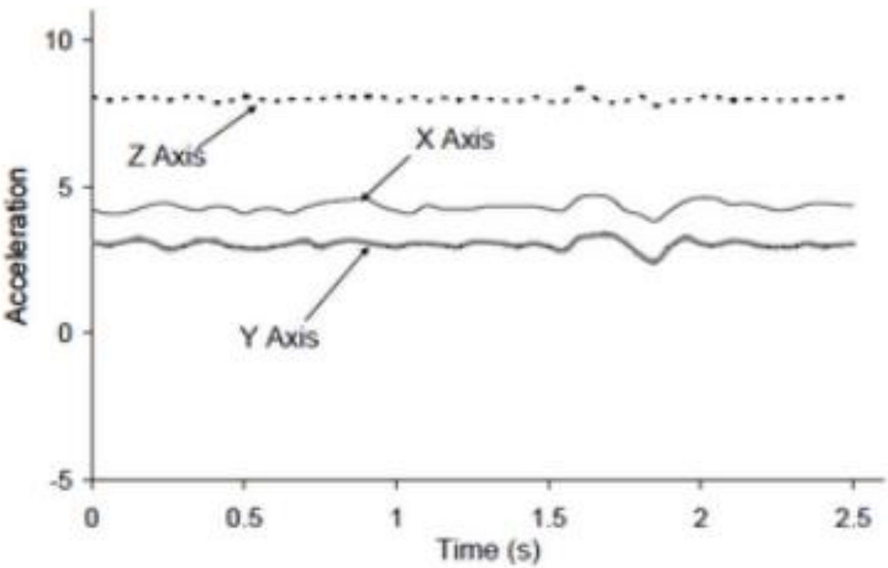
Waking



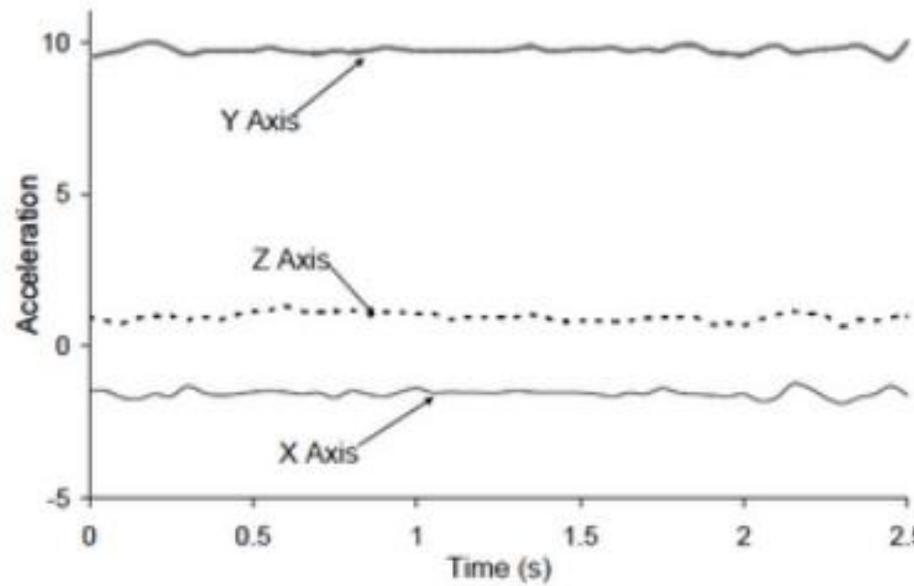
Jogging

Acceleration Signals

- Let’s say we also want to know if a person is standing or sitting.

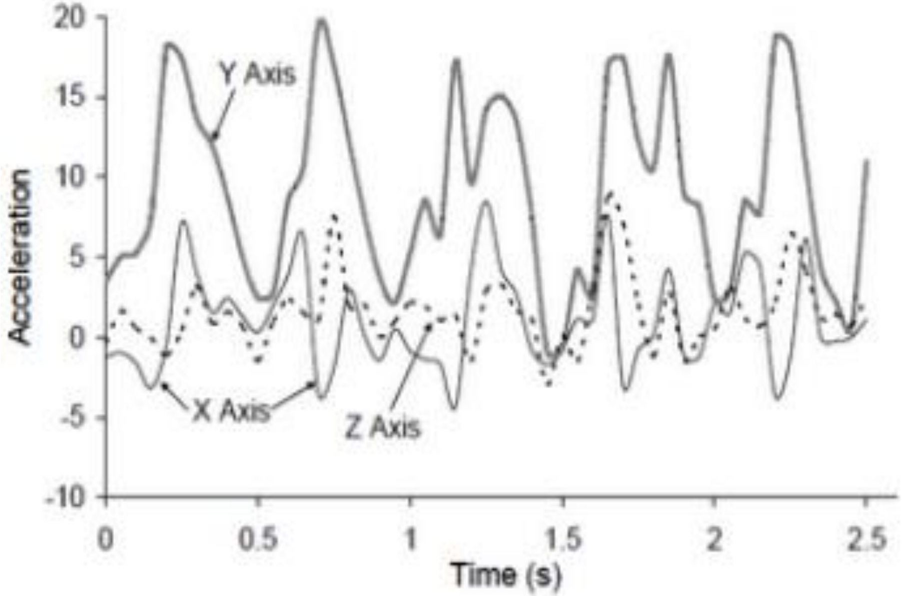


Sitting

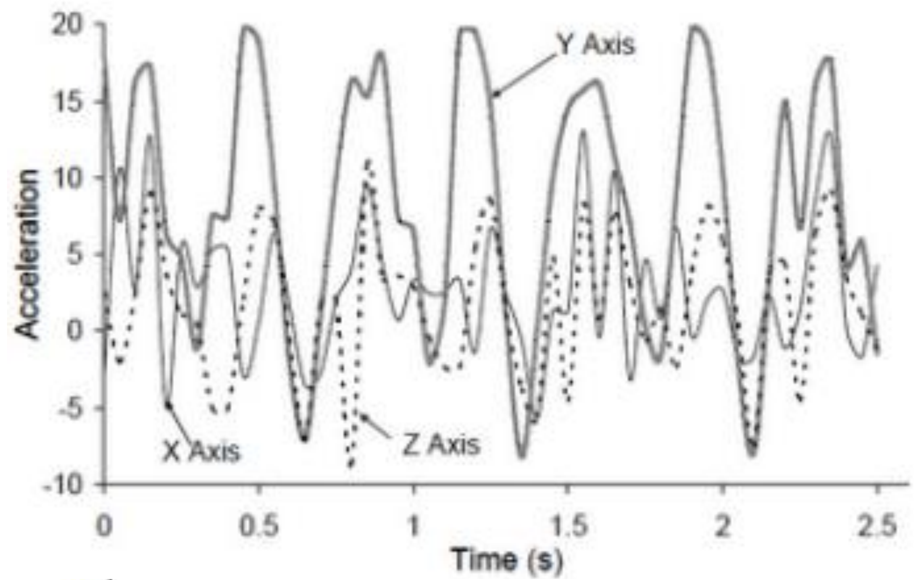


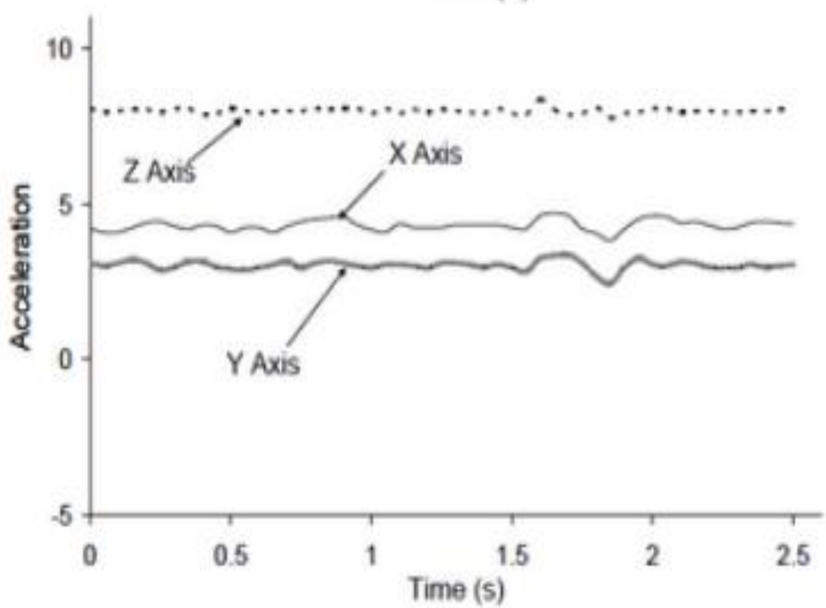
Standing

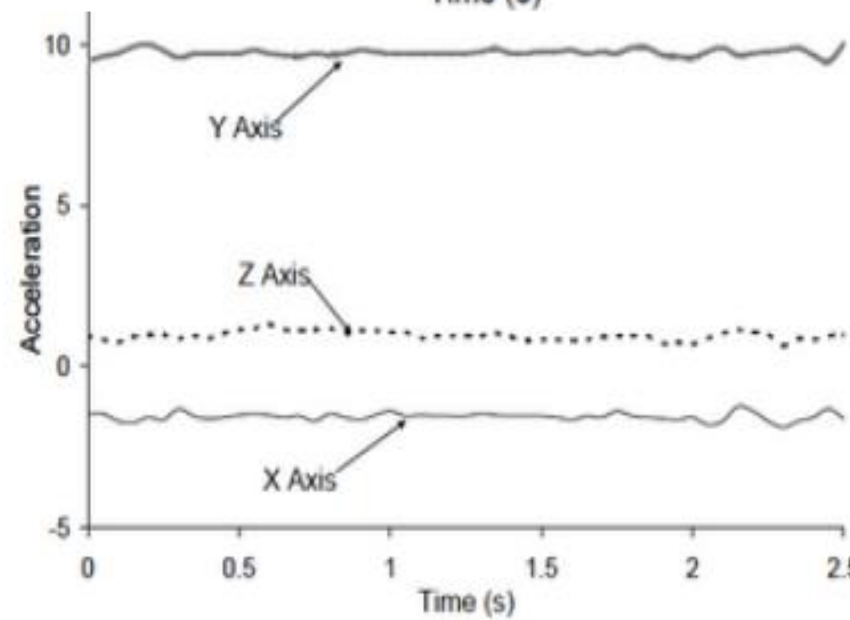
Waking



Jogging







Sitting

Standing

How Do We Classify Activities?

- Input: 2.5 seconds of 3-axis acceleration data (sampling rate: 120 Hz)
- Output: User activity (one of sitting, standing, walking, jogging)

A Simple Heuristic?

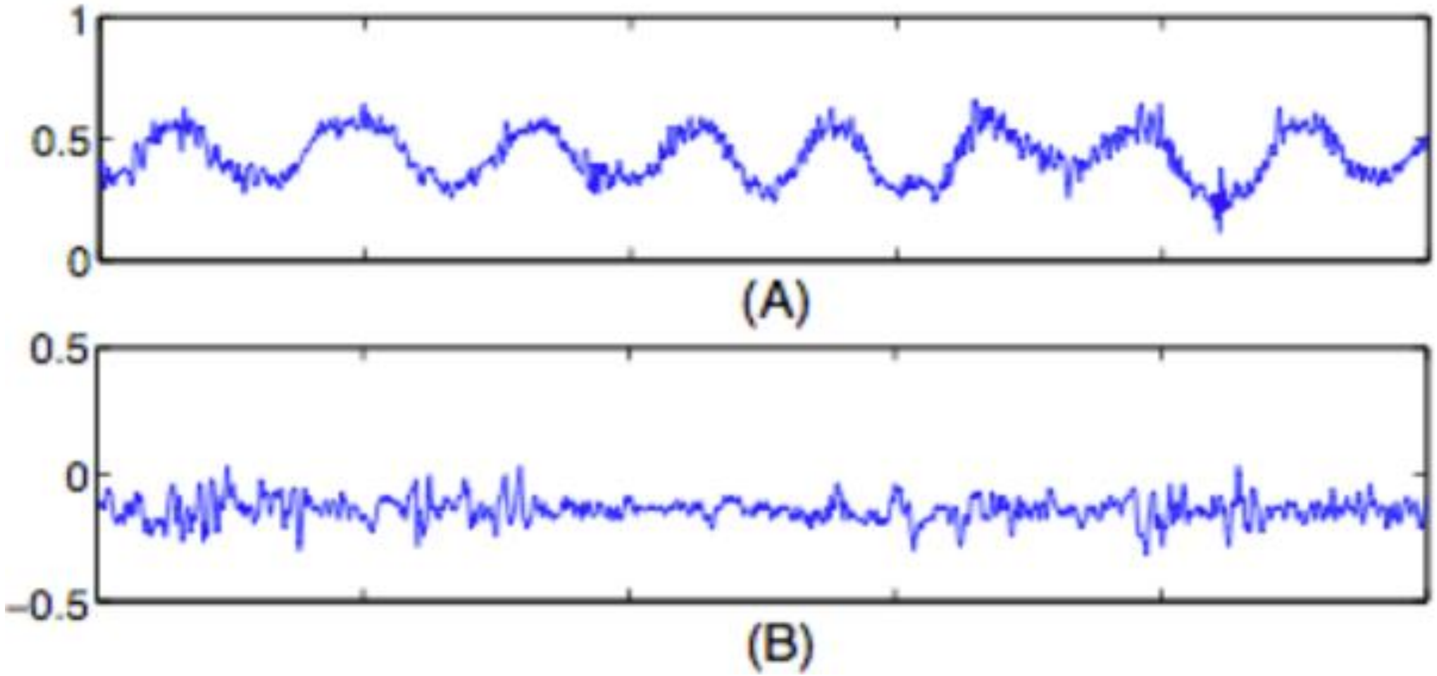
- If STDEV(y-axis samples) < $C_{Threshold1}$
  - If AVG(y-axis samples) > $C_{Threshold2}$
    - output standing
  - Else
    - output sitting
- Else
  - If FFT(y-axis samples) < $C_{Threshold3}$
    - output walking
  - Else
    - output jogging

Problems of The Heuristics

- How do we determine good features and good thresholds?
  - How do we know STDEV is better than MAX?
  - How do we know AVG is better than Median?
  - How do we know the right values for $C_{threshold}$?
- What if a user puts her phone in her bag, not in her front pocket?
  - The Y-axis of the phone is not anymore the major axis of movement.
- How do we solve these problems? A better heuristic?

One Activity, Two Distinct Patterns

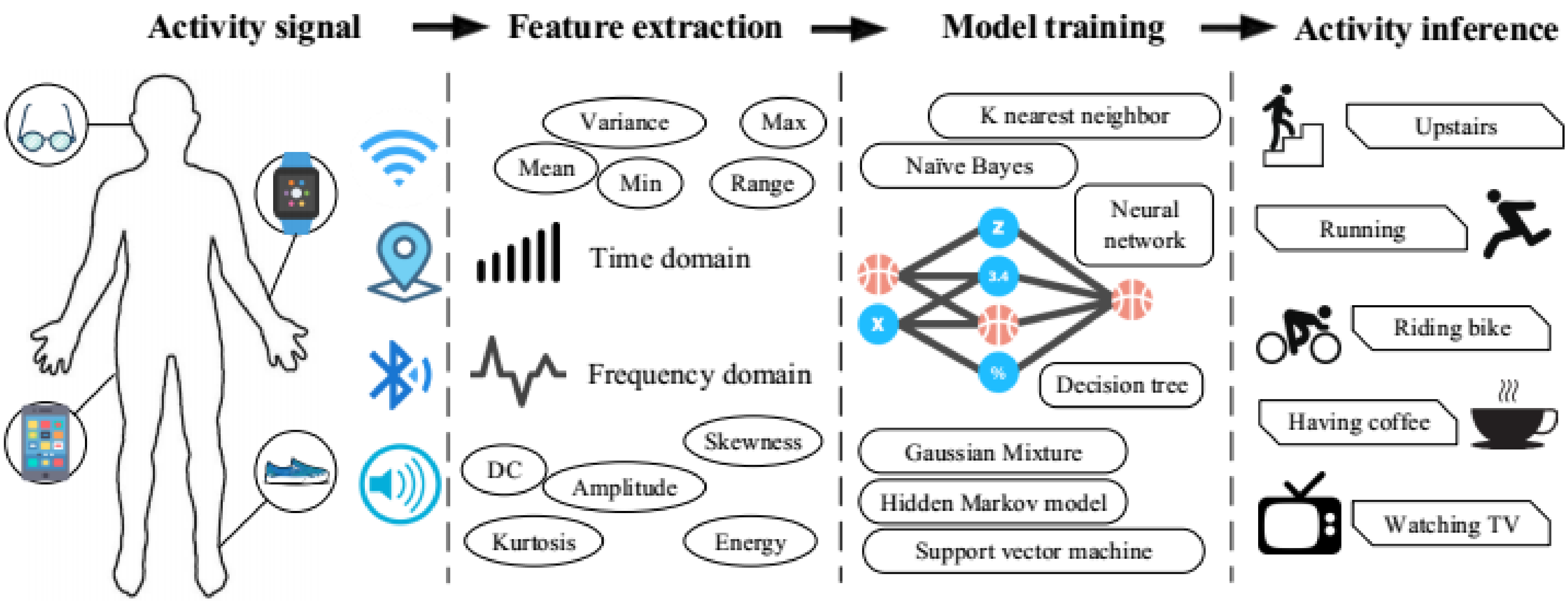
Acceleration while cycling



(a) Phone in the pocket (b) phone in the backpack

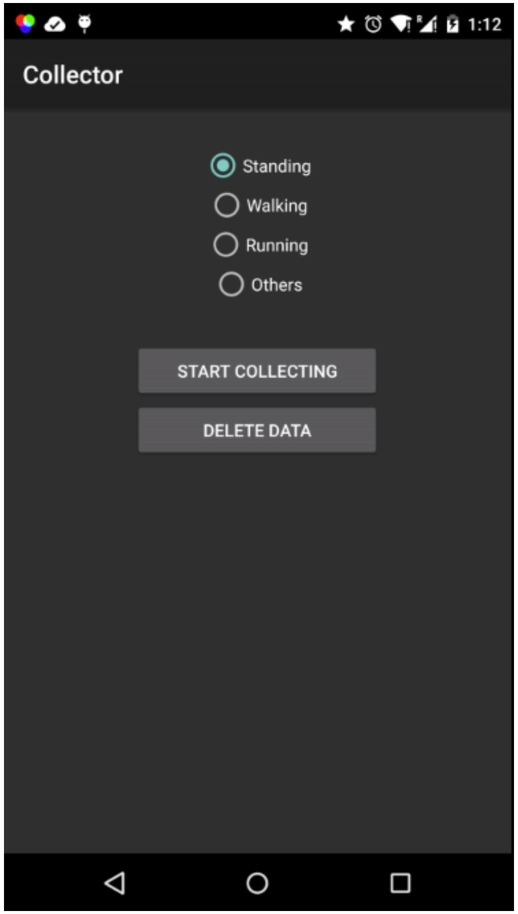
Source: Prof. Andrew Campbell's Lecture Note

Machine Learning Techniques



Step 1: Data Collection

- The first step is to collect labeled data.
- Labels mean that the ground truth corresponding to the raw data.
- E.g.) an hour of the raw accelerometer data from a phone, as well as user-provided labels regarding their state (walking, running, etc.).
- This data is referred to as a training dataset.
- Need to collect sufficient data for each activity to classify.



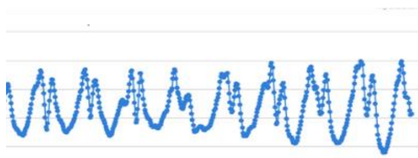
Step 2: Feature Extraction

- Identify distinguishing features in the data
- Time-domain features: Aggregate statistics of the data (e.g., avg. stdev.)
- Frequency domain features: Periodic patterns and rhythmic behavior in the signal. (e.g., walking and running have different dominant frequencies)

Time domain features	Frequency domain features
Mean, Median, Variance, Standard deviation, Min, Max, Range, Zero-crossings, Angle, Angular velocity, etc.	Dominant frequency, Signal Energy, etc.

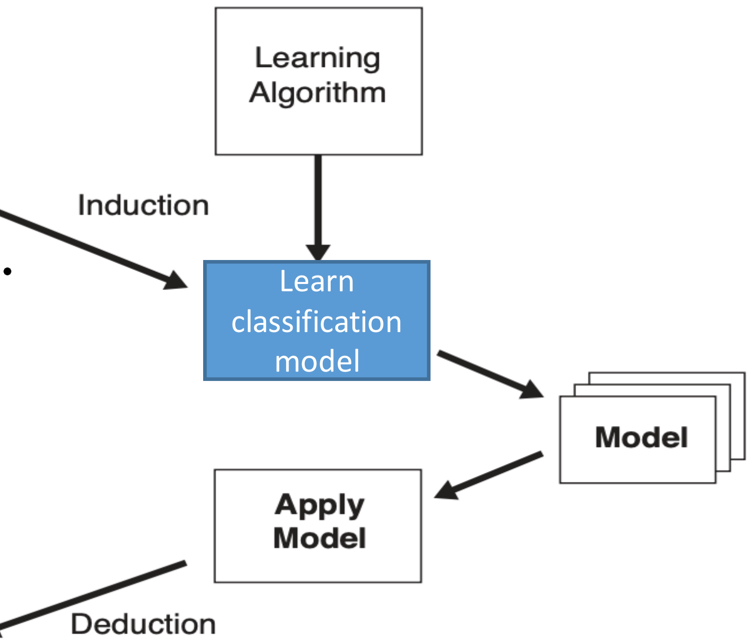
Step 3: Classifier Training

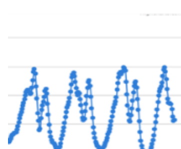
- A classifier identifies which of the features is most useful in distinguishing between the different activities.

Training Data: 

Labels: walking, running, sitting, ···

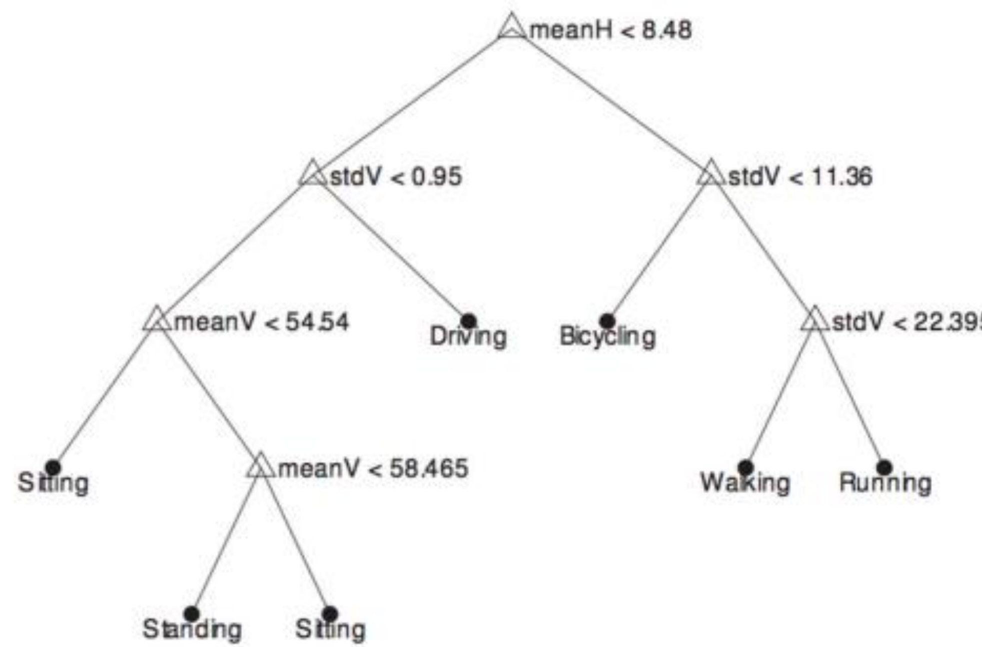
Features to use



New
Data  

Decision Tree

- A simple but effective ML classifier.
- Given training data, the algorithm can automatically determine the important features and their thresholds.
- Then, when a new data is given, it is trivial to classify what activity it belongs to



An example decision tree to distinguish 6 activities with 3 features

How to Train a Decision Tree?

- Search for the C4.5 algorithm if you are interested.

- Pseudocode

  1. For each feature *f*, find the normalized information gain (a metric to effectively split data into classes) from splitting on *f*
  2. Let *f_best* be the attribute with the highest normalized information gain
  3. Create a decision node that splits on *f_best*
  4. Recurse on the sublists obtained by splitting on *f_best*, and add those nodes as children of node

Other ML Techniques

- Random Forest
- Support Vector Machine
- Naïve Bayes
- Hidden Markov Model
- Gaussian Mixture Model
- Neural Networks
- …

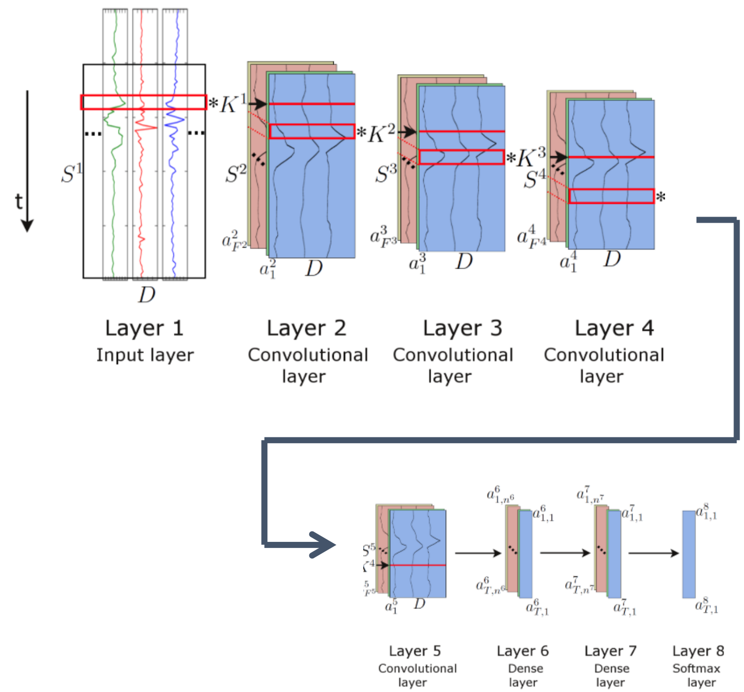
Activity Recognition

- This still is an active on-going research topic.
- To recognize various types of activities (e.g., eating, smoking, exercising, swimming, etc.).
- To recognize activities more accurately using different types of devices, sensors, and machine learning algorithms.
- To recognize various activities in a resource efficient way.

Active Research Directions

- **Use and fuse data from other sensing devices (e.g., smart bulbs)**

- **Use of deep learning**

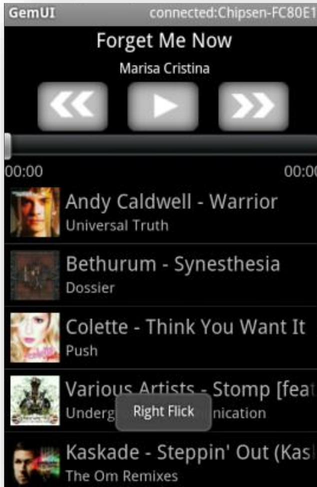


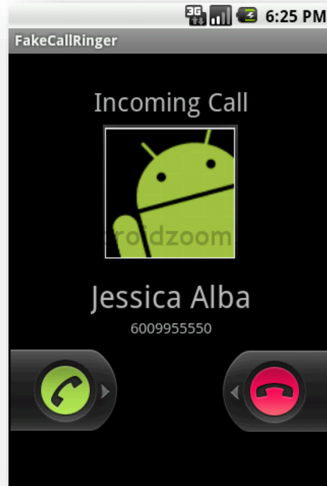
Gesture Recognition

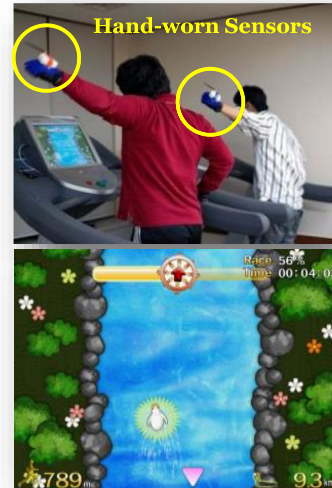
Gesture-Based Interaction

- Gestures are a natural way of interacting with object and other people.
- Gestures can be particularly useful when other forms of interactions are difficult.
  - Controlling a phone while running
  - Communicating with impaired people
- It can be used as complement to other types of interaction modalities.

Applications

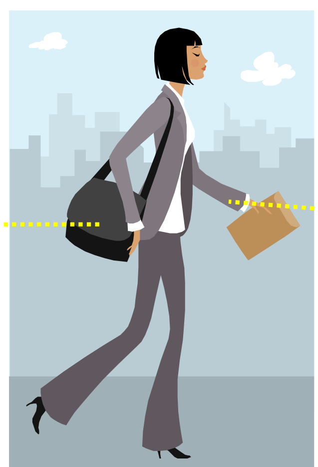






…

Gesture Interaction

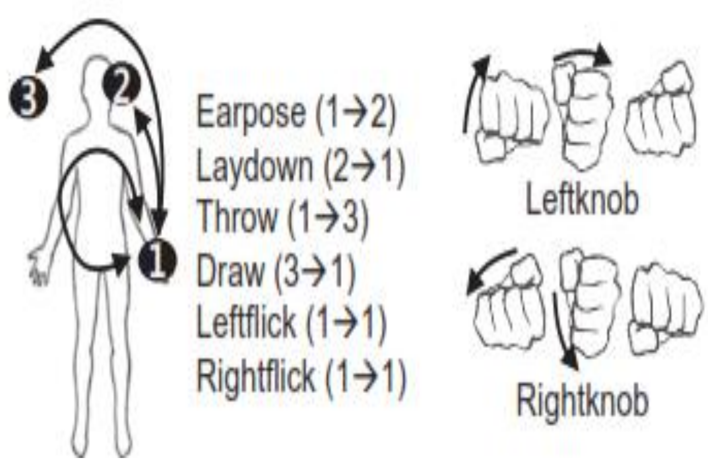


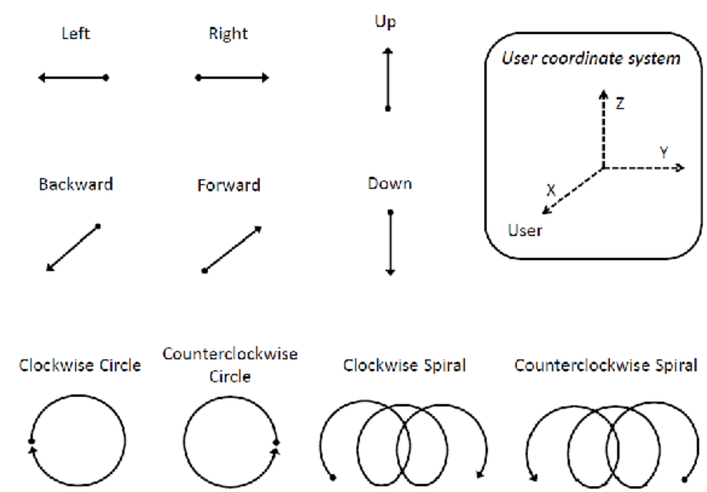
Smartphone

Motion Sensor(s)

Example Gesture Vocabulary

- What are the intuitive and accurately recognizable set of gestures?
  - Active research area in HCI





Gesture Recognition

- Many different approaches have been studied and developed.
  - Vision-based
  - Sound-based
  - Motion-based
  - Wireless signal-based

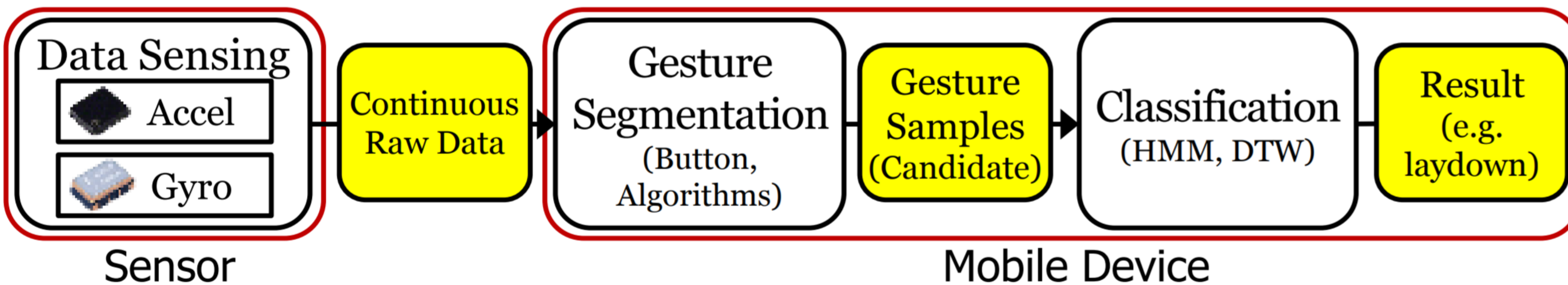
- We will look into a motion-based approach
  - Users are increasingly adopting wearable devices.

E-Gesture: A Collaborative Architecture for Energy-efficient Gesture Recognition with Hand-worn Sensor and Mobile Devices

ACM SenSys 2011

Gesture Recognition Pipeline

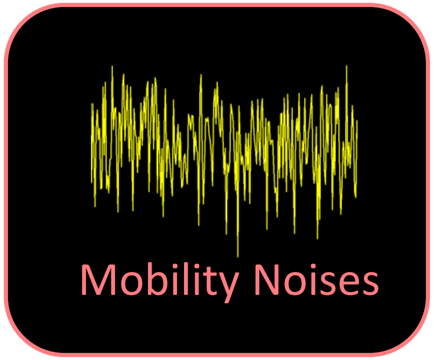
- Smartwatch: Data source
- Smartphone: Gesture recognizer



Challenges

- Providing Energy-efficient Gesture Processing

- Accurately Detecting and Classifying Hand Gestures



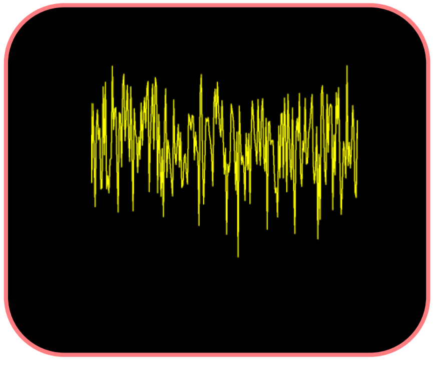
Design Challenges

- Providing Energy-efficient Gesture Processing

20hrs **Sensor, 250mAh**

24hrs → 17hrs **Smartphone**

- Accurately Detecting and Classifying Hand Gestures



Over 90% **False detections**

Only 70% **Classification**

Approaches

- Investigated characteristics of Accel and Gyro
  - Accelerometer: Mobility-Sensitive, Energy-Efficient
  - Gyroscope: Mobility-Robust, Energy-Hungry
- Designed energy-efficient, mobility-robust gesture detection architecture
  - Triggering Gyroscope by analyzing Accelerometer Signal
  - Adjusting Accelerometer sensitivity by Gyroscope Validation
- Suggested two gesture classification architectures considering users' mobilities (based on HMM)

Mobility Noises

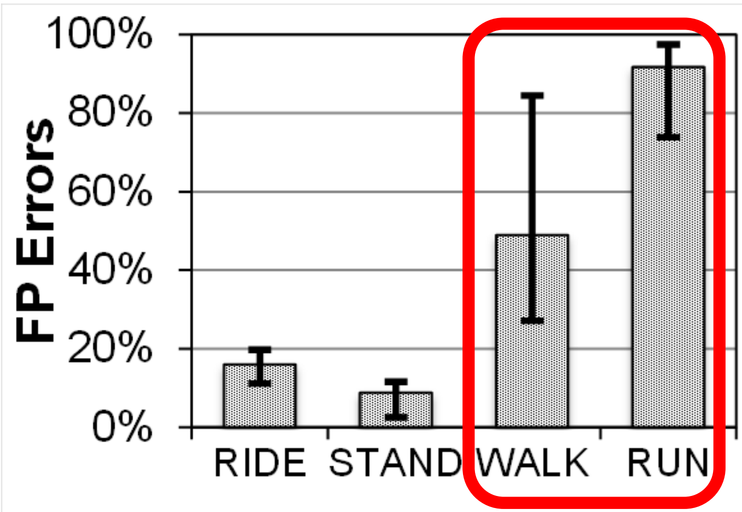
- Makes it difficult to distinguish intended hand motions from noises

Standing still

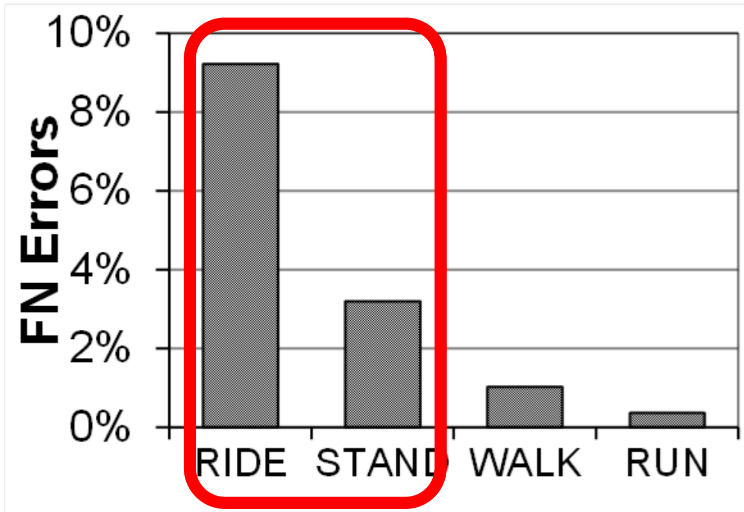
Walking

Running

Gesture Segmentation: Accel



Lower fixed threshold
➔ False-positives
on high mobility



Higher fixed threshold
➔ False-negatives
on low mobility

Gesture Segmentation: Gyro

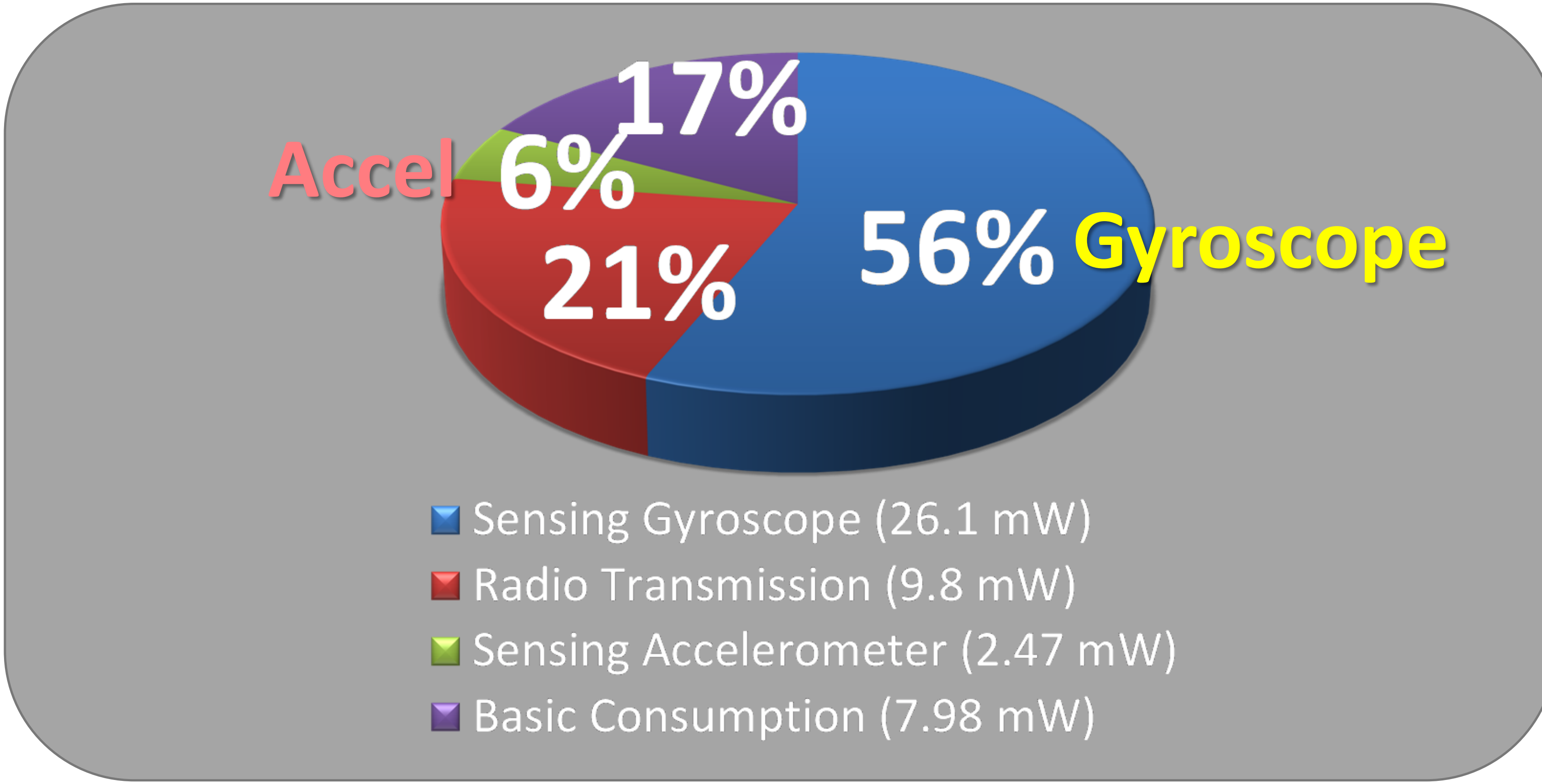
- Accelerometer is more sensitive to mobility
- Gyroscope is more robust to mobility

	Mobility Situation			
	RIDE	STAND	WALK	RUN
Accel-based	0.15G	0.15G	0.2G	0.35G
Gyro-based	25 degree/sec			

Optimal threshold for Accel and Gyro

(minimizes FPs without incurring FNs)

Problem with Gyroscope



Sensor-side Energy Profile
(Atmega128L, CC2420, Accel and Gyro)

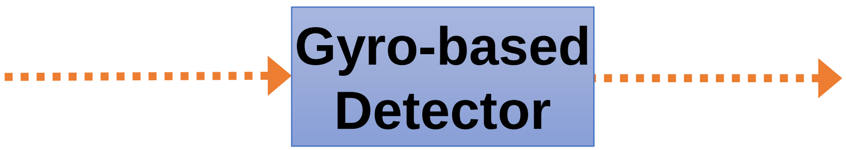
Energy-Performance Tradeoff

	Energy Consumption	Mobility Robustness	Segmentation Accuracy
Accel-based	Low	Poor	Passable
Gyro-based	High (9x accel)	Good	Good

Approaches

- Investigated characteristics of Accel and Gyro
  - Accelerometer: Mobility-Sensitive, Energy-Efficient
  - Gyroscope: Mobility-Robust, Energy-Hungry
- Designed energy-efficient, mobility-robust gesture detection architecture
  - Triggering Gyroscope by analyzing Accelerometer Signal
  - Adjusting Accelerometer sensitivity by Gyroscope Validation
- Suggested two gesture classification architectures considering users' mobilities (based on HMM)

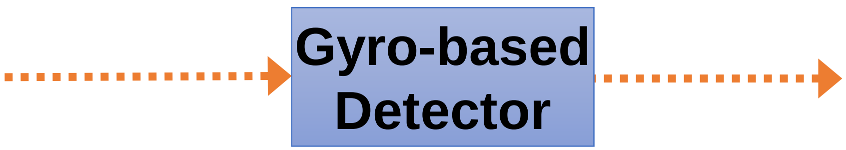
Closed-loop Collaborative Segmentation



Accurate, High energy

Gyro-based Detector

Closed-loop Collaborative Segmentation



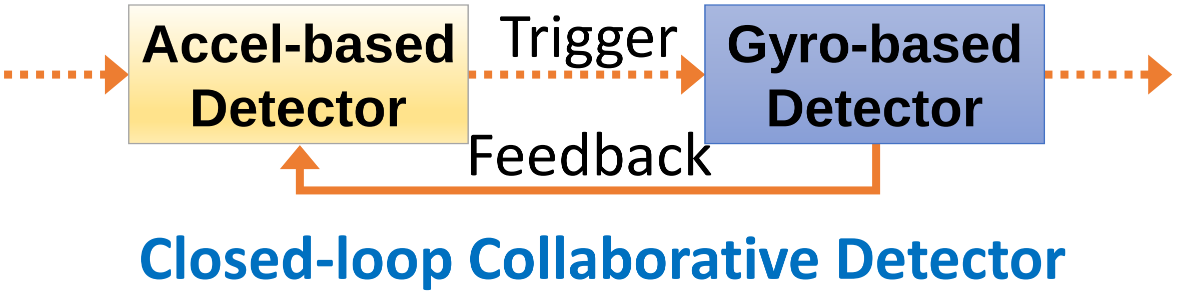
Accurate, High energy

Gyro-based Detector

Accurate, High energy because of mobility

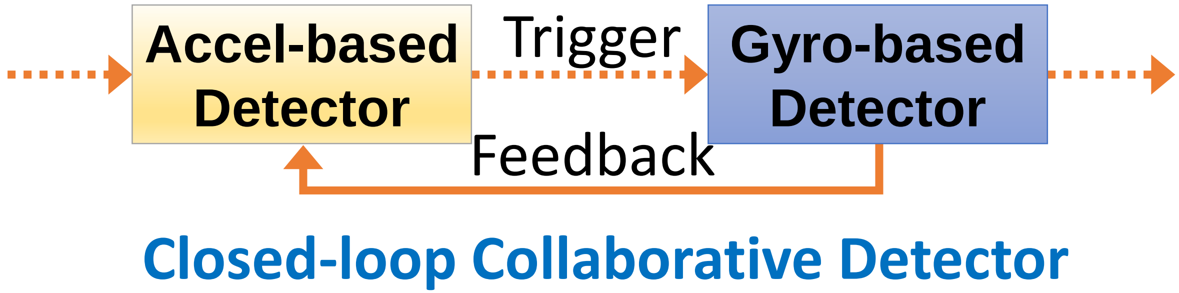
Open-loop Detector

Closed-loop Collaborative Segmentation



Accurate, Low energy

Closed-loop Collaborative Segmentation



Accurate,
Low energy

Performance-preserving, Energy-saving Collaborative Sensor Fusion

Approaches

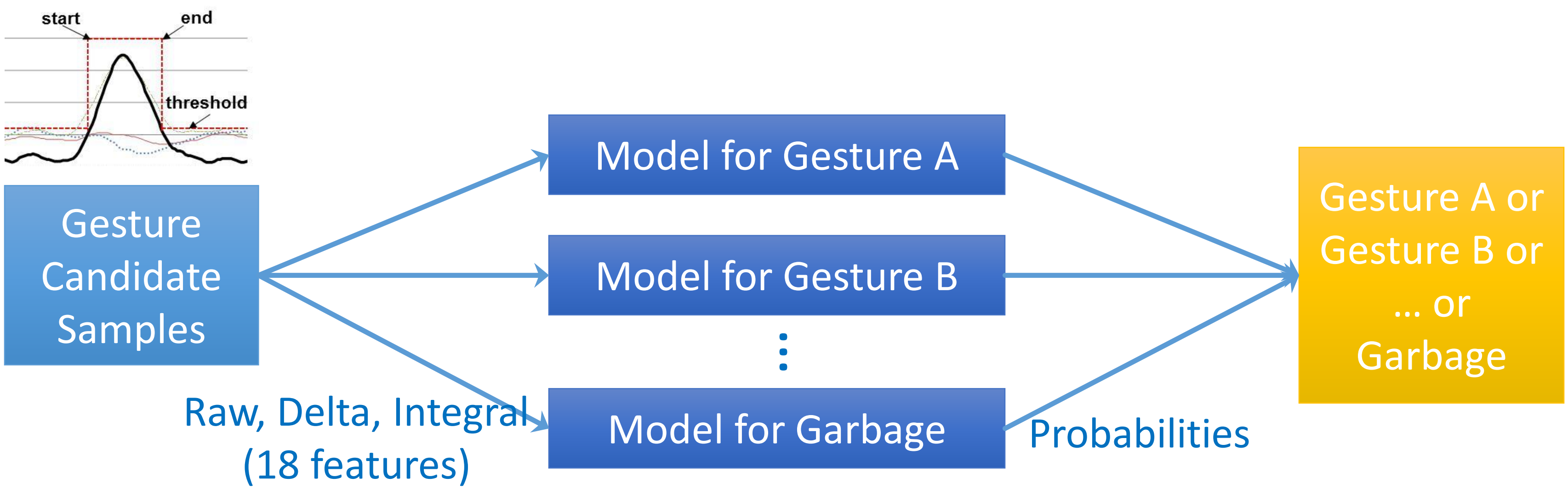
- Investigated characteristics of Accel and Gyro
  - Accelerometer: Mobility-Sensitive, Energy-Efficient
  - Gyroscope: Mobility-Robust, Energy-Hungry

- Designed energy-efficient, mobility-robust gesture detection architecture
  - Triggering Gyroscope by analyzing Accelerometer Signal
  - Adjusting Accelerometer sensitivity by Gyroscope Validation

- Suggested two gesture classification architectures considering users' mobilities (based on HMM)

Basic HMM

- Trained with samples collected in stationary setting
- Classification accuracy drops in mobile situations



8-state left-right HMMs

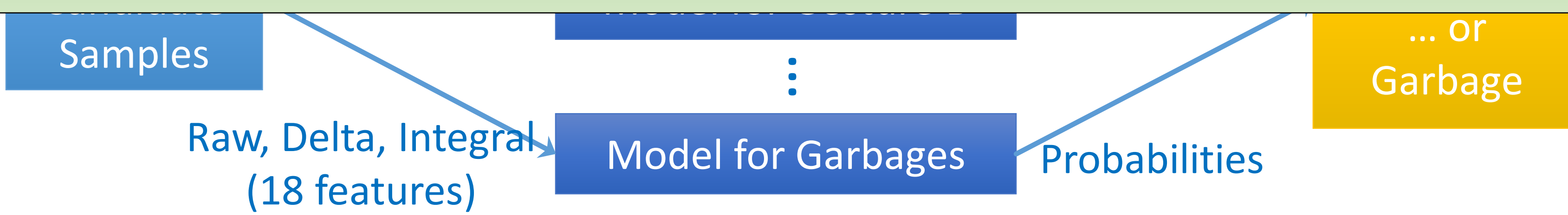
Basic HMM

- Trained with samples collected in stationary setting
- Classification accuracy drops in mobile situation

Design alternatives:

1) Adapt models to mobility changes (in run-time)

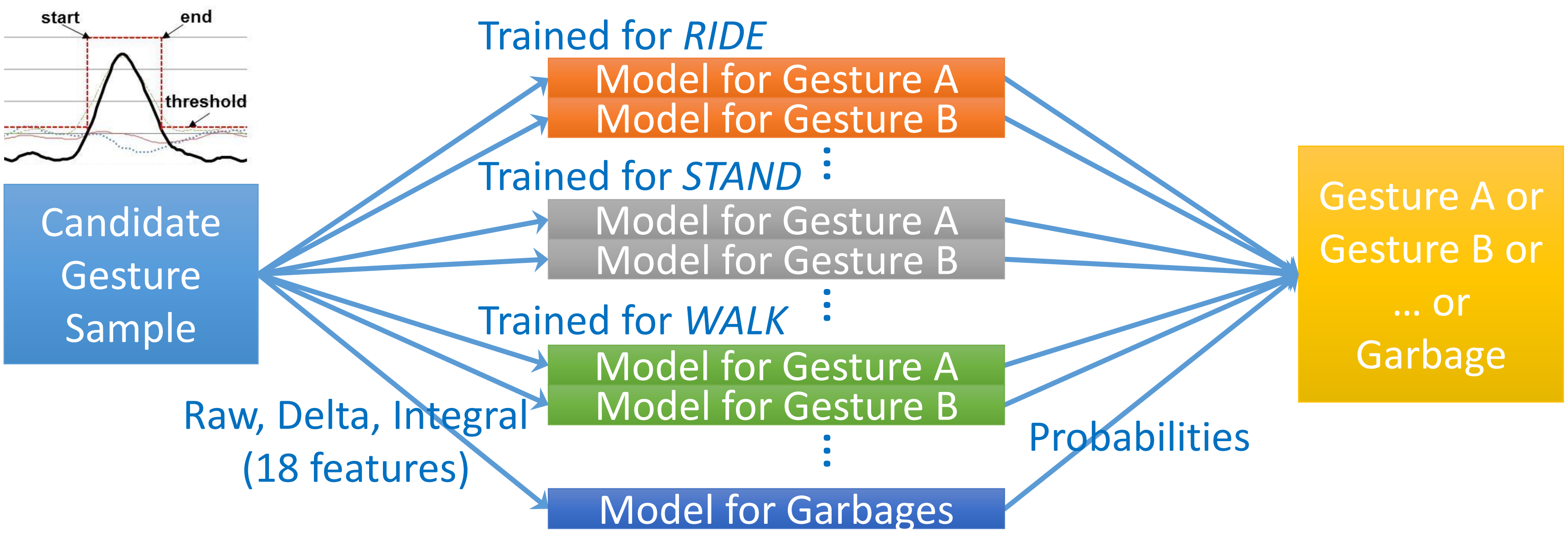
2) Train several different models for predefined set of mobility situations



8-state left-right HMMs

Multi-Situation HMM

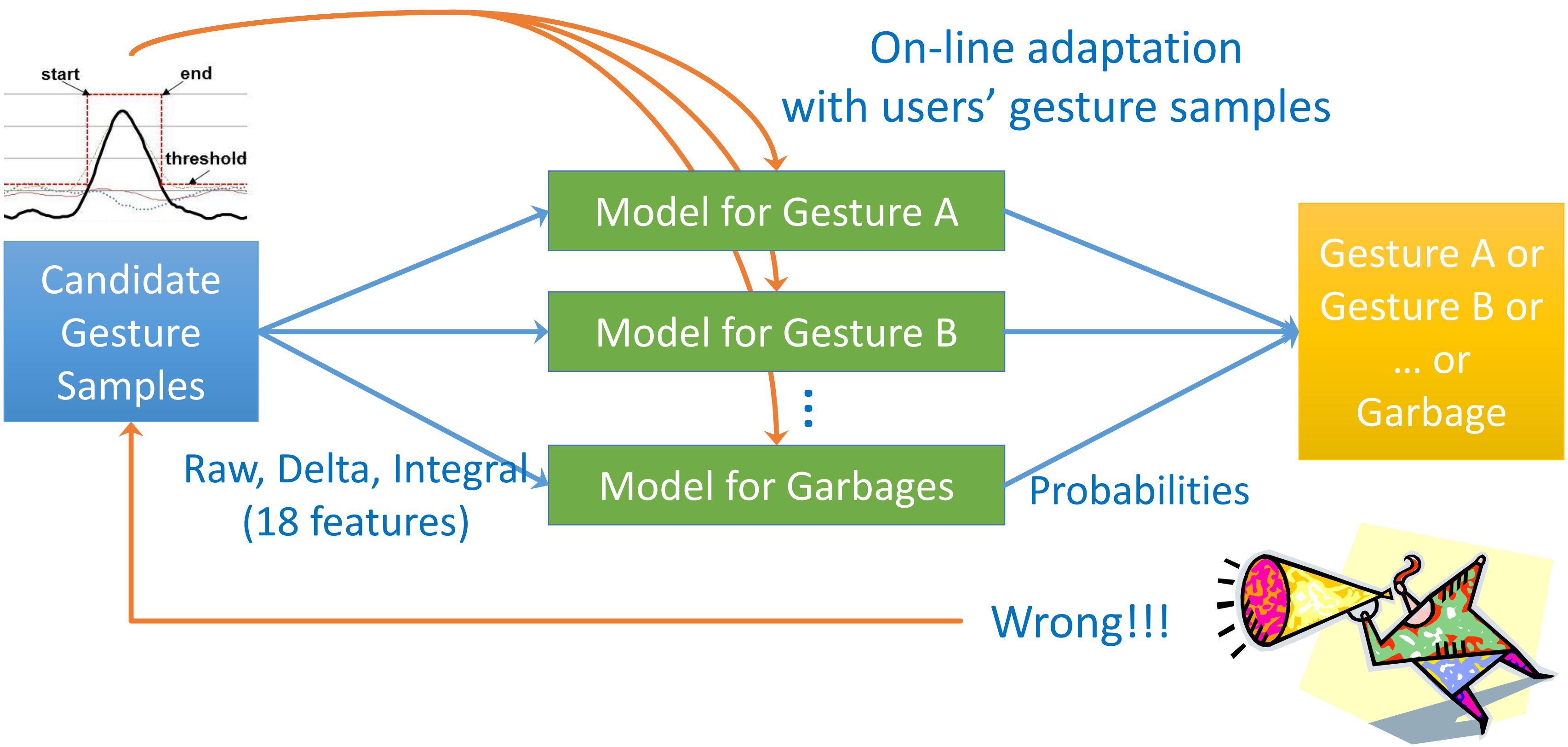
- Train models separately for representative mobility situations
  - e.g. Riding a car, Standing, Walking, Running
- Classify by evaluating all models



$$Number\ of\ models = Number\ of\ situations\ \times\ Number\ of\ gestures$$

Adaptive HMM

- Update the models with gesture samples
  - Negative update scheme of uWave [PerCom09]
  - By MLLR (Maximum Likelihood Linear Regression) adaptation



Adaptive vs. Multi-Situation HMMs

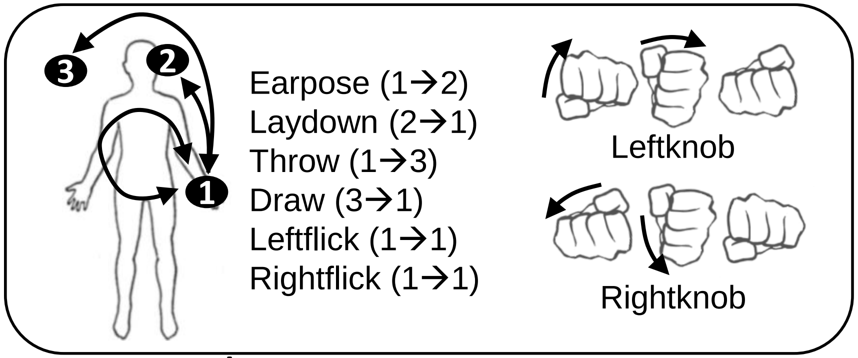
	Basic	Adaptive	Multi-Situation
Adaptation cost (Users' burden)	none	large	none
Training cost	# of gestures	# of gestures	# of gestures x # of mobile situations
Evaluation cost (Processing)	# of gestures	# of gestures	# of gestures x # of mobile situations

Implementation

- Sensor node
  - Atmega128L MCU
  - CC2420 Zigbee Radio
  - Sensors
    - 3-Axis Accelerometer (ADXL335)
    - 3-Axis Gyroscope (3 XV-3500CB)
    - 40Hz Sensing
  - Vib motor
- Smartphones
  - Nokia N96, Google Nexus One
  - Bluetooth Radio
    - Bridge node to convert Zigbee→ Bluetooth

Gesture Data Workload

- 4 Representative mobility situations
  - Riding a car, Standing, Walking, Running
- 8 Intuitive gestures

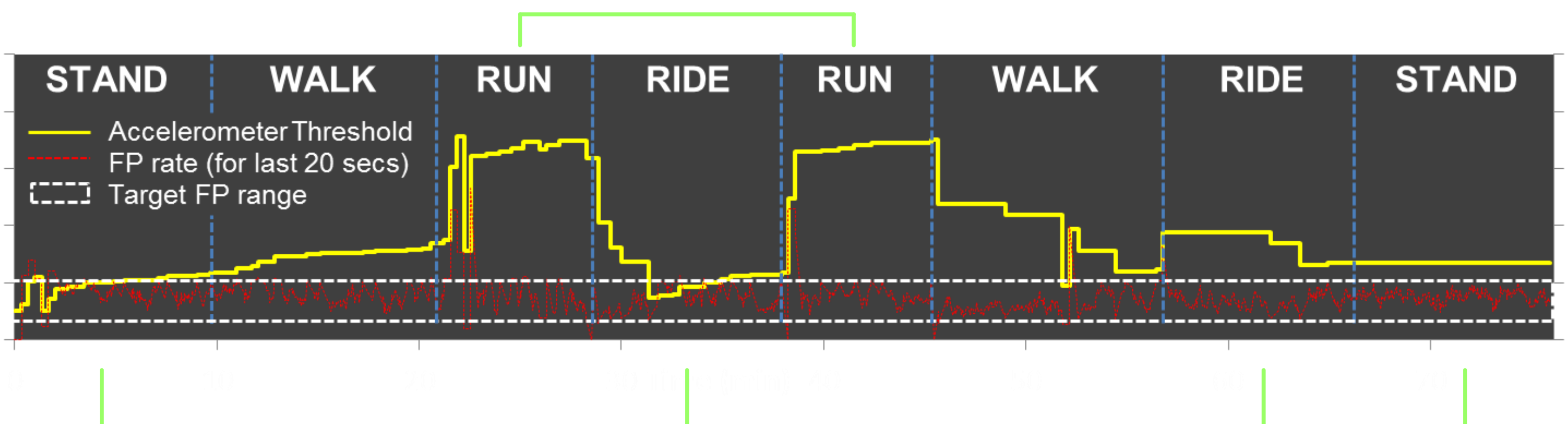


- Data Collection
  - 4 situations × 8 gestures × 30 samples × 7 participants = Collected 6720 gesture samples in total
  - Also collected non-gestures to generate test workloads
- Workload configuration (for energy efficiency)
  - Ratio of gestures: 10% of total time
  - Mobility mixture: 75% from stationary (50% STAND, 25% RIDE)
    25% from mobile (12.5% WALK, 12.5% RUN)

Threshold adaptation of Closed-loop detector

Q: Does the closed-loop collaborative detector adapt accelerometer threshold well?

Higher Threshold in Higher Mobility

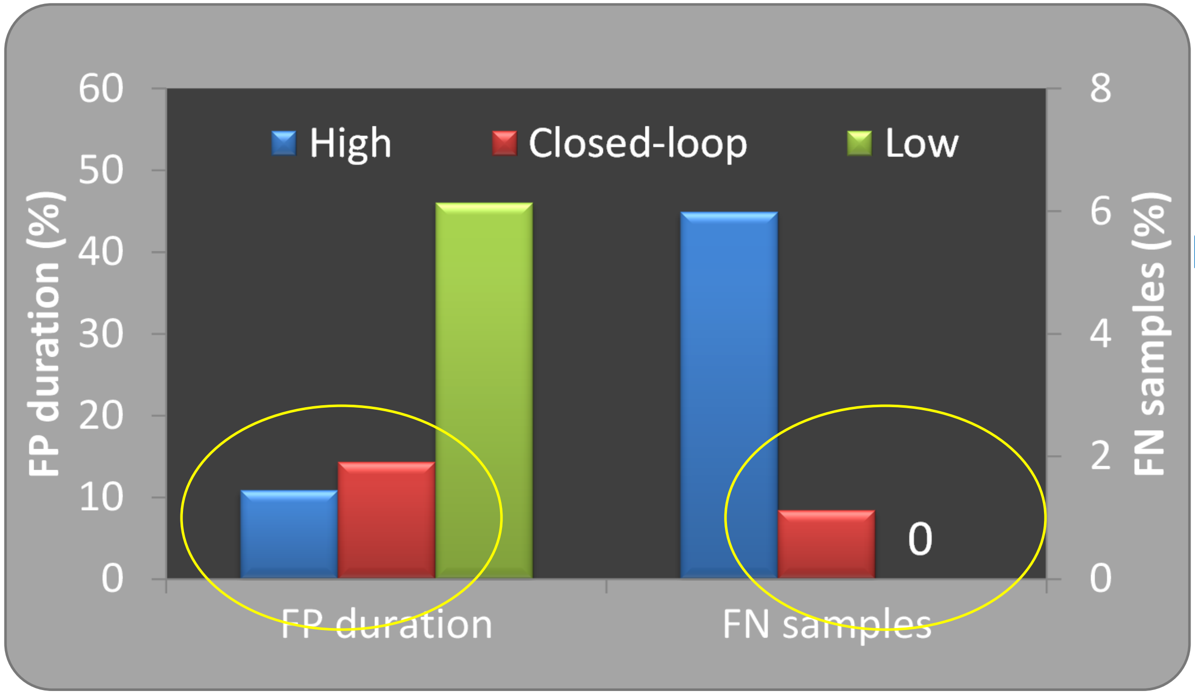


Lower Threshold in Lower Mobility

Performance of Closed-loop Detector

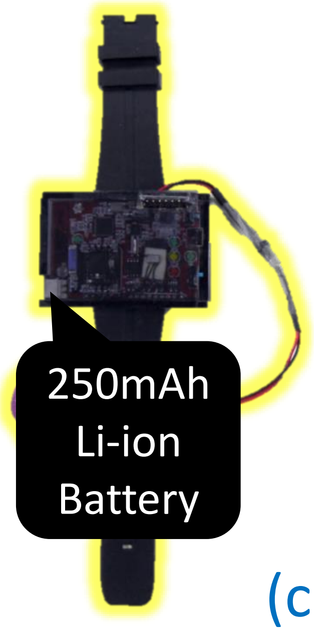
Q: How much does the closed-loop detector suppress false-positives and false-negatives from the accel-based detector?

False positives from accel detector



False negatives from accel detector

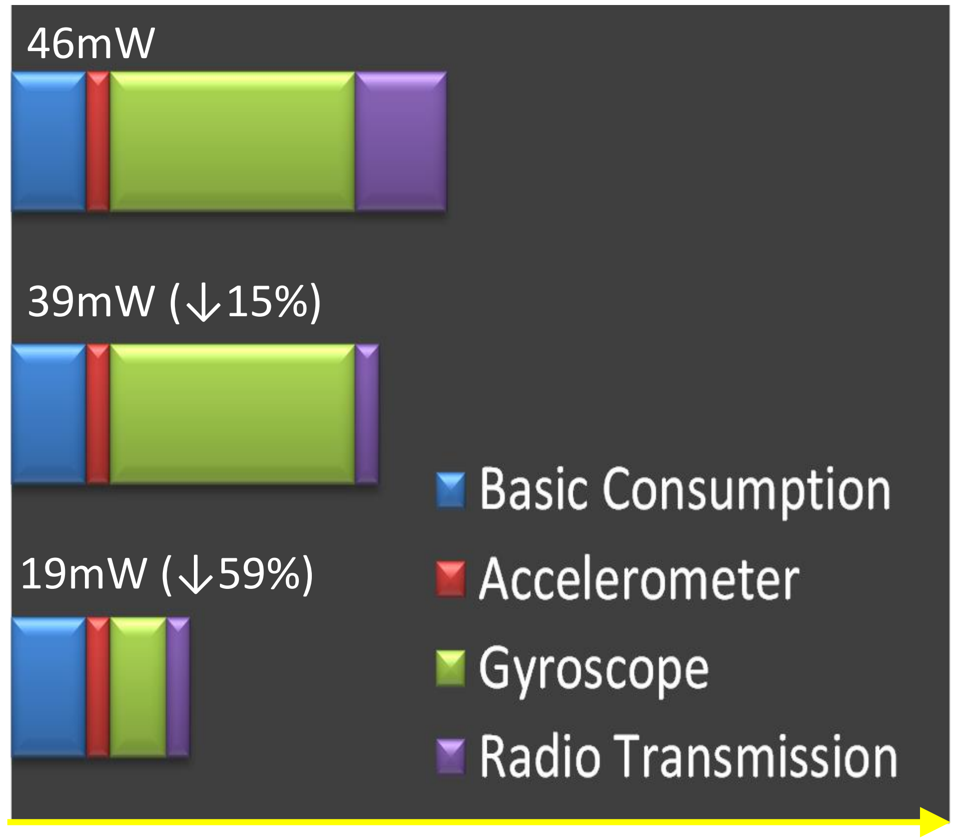
Sensor-side Energy Savings from Closed-loop Architecture



transmit
raw sensing data:
20 hrs

transmit detected
gestures using gyro
(no sensor control)
23.7 hrs (1.2x)

transmit only
detected gestures
(closed-loop detection):
48.7 hrs (2.4x)

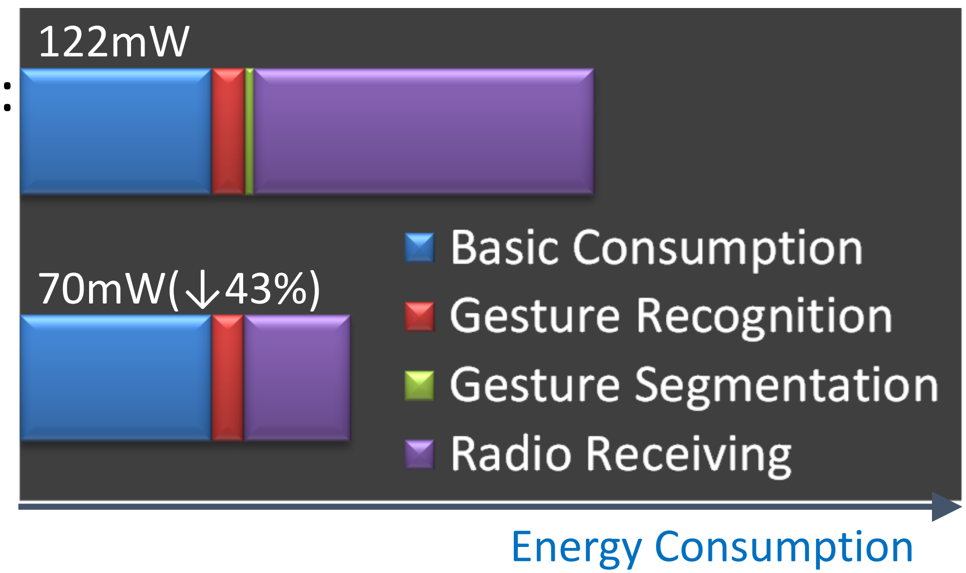


59% less energy consumption, 2.4x longer lifetime

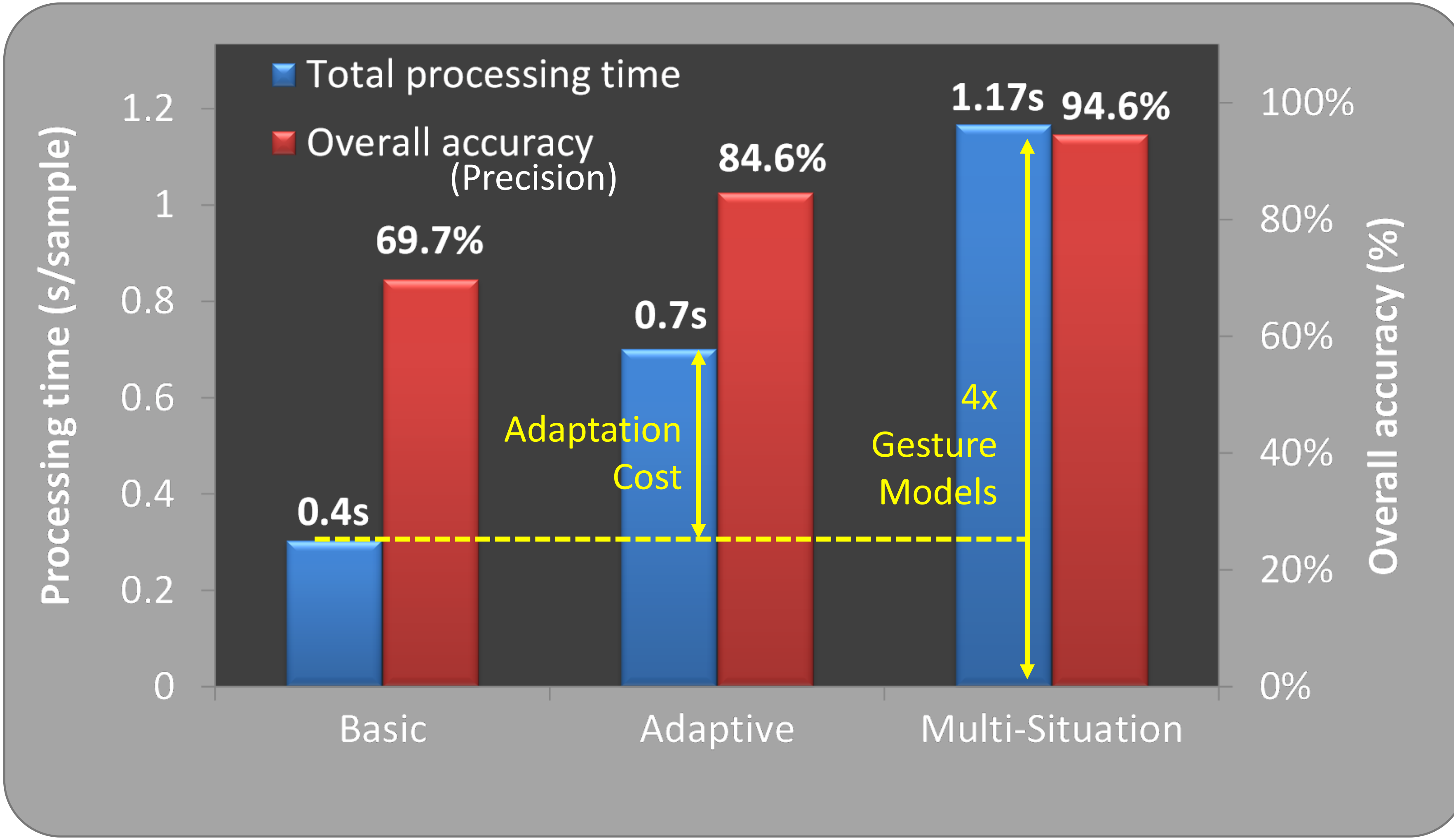
Mobile-side Energy Savings from Sensor-side Gesture Detection

All processing on mobile:
42.1hrs

Sensor-side Gesture Detection:
74hrs (1.8x)



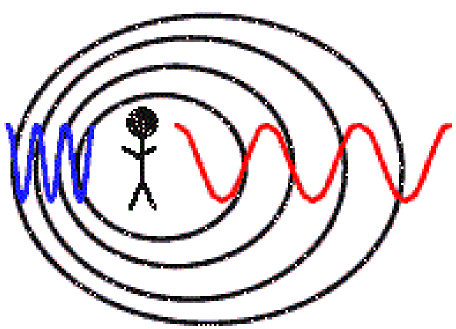
Gesture Classification Performance

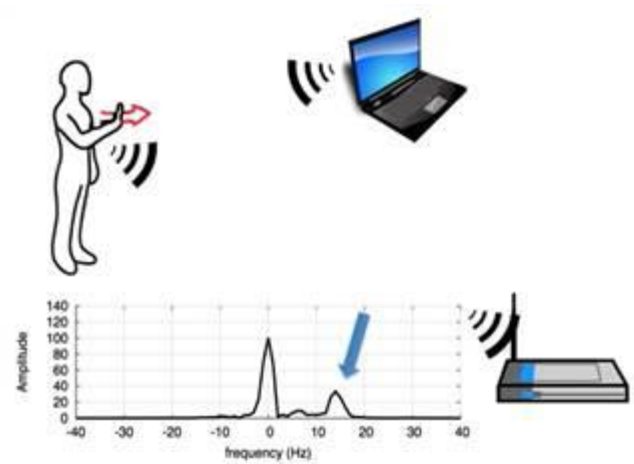


Other Techniques

WiSee:
Device-free Gesture Recognition

- Wi-Fi Doppler shifts
  - Humans reflect Wi-Fi signals, thus can be treated as signal sources.
  - Human motion introduce Wifi Doppler shifts.

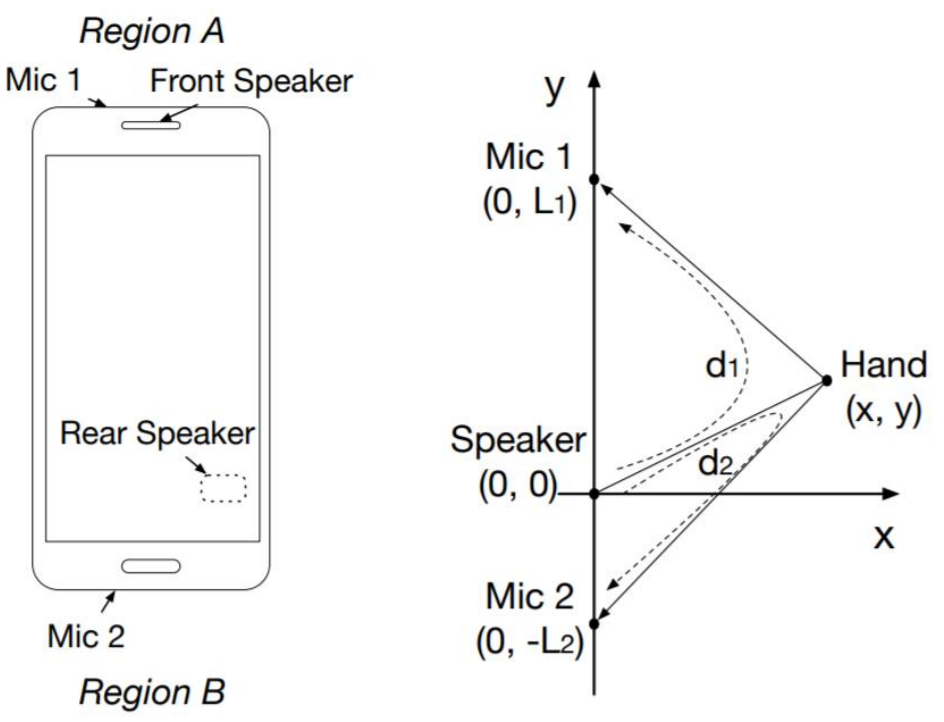




- Different gestures exhibit different patterns.

LLAP: Sound-based Gesture Recognition

- Extracts the sound signal reflected by the moving hand/finger.
- Measures the phase changes of the sound signals caused by hand/finger movements.
- Converts the phase changes into the distance of the movement.



Other Systems and Issues

- Active area of research to design and develop an accuracy, robust, and resource-efficient gesture recognition techniques.

- There could be many other approaches using ambient light, depth camera, etc. depending on the use cases.

- Battery-free gesture recognition is a direction that people started exploring extensively.