Activities and Gestures

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

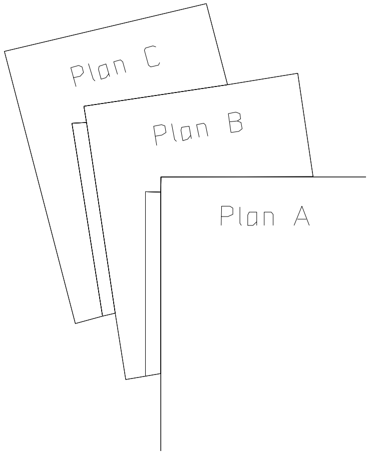
Henry Ford

SEOUL NATIONAL UNIVERSITY

Human-Centered
Computer Systems Lab

Overview

- Objective
  - ✓ To understand exemplary techniques and challenges for activity and gesture recognition
- Content
  - ✓ Activity sensing and recognition
  - ✓ Gesture sensing and recognition
- After this module, you should be able to
  - ✓ Understand the basics of activity recognition
  - ✓ Understand the basics of gesture recognition



Activity Recognition

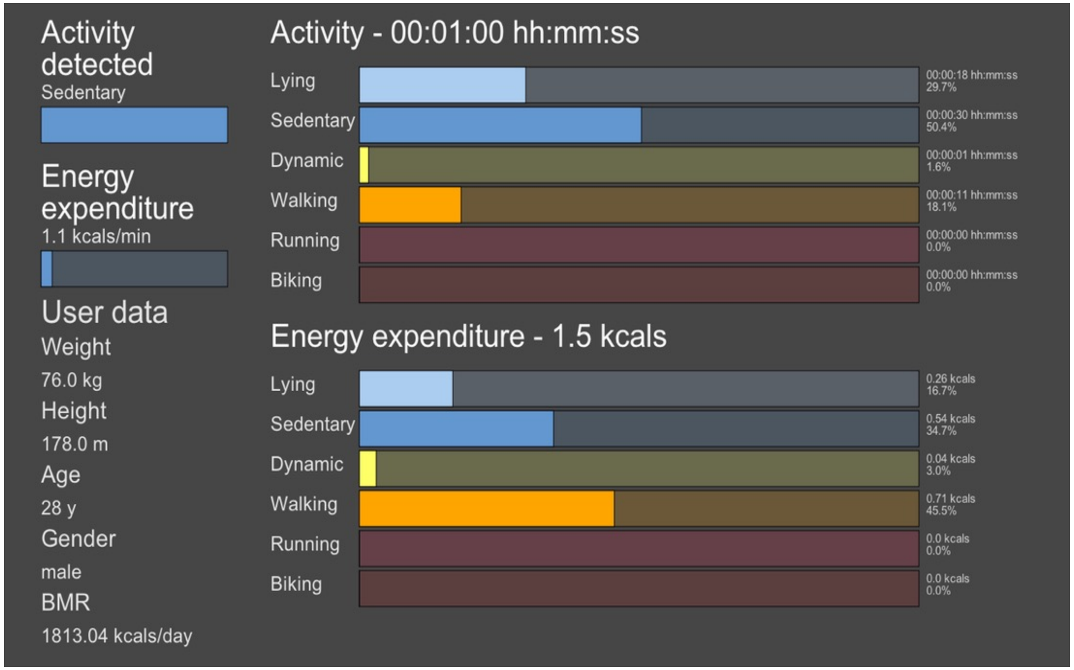
- Identifying the physical activity of a user
  - ✓E.g., jogging, walking, sitting, standing
- Providing useful knowledge about the habits of users passively—just by carrying smartphones.
- Wide range of applications
  - ✓Activity-aware phone configuration (e.g., sending calls directly to voicemail if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

Example App: Activity Tracker

- Everyday exercise progress monitor and motivator
- Provide reliable feedback about how much they move. (People often overestimate!)
- Provide instant and constant feedback about activity levels.
- Gamify to encourage individuals to compete in getting fit and losing weight.

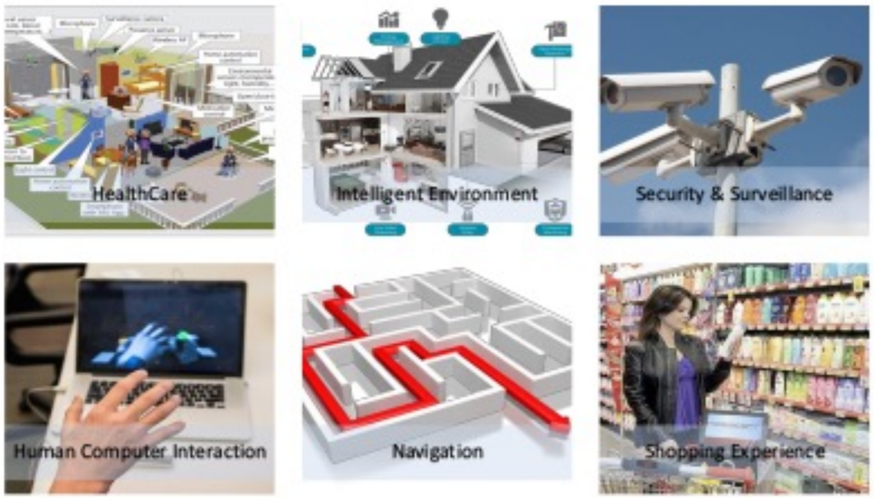
Example App: Quantified Self

- Continuously track user activities and objectively summarize and visualize data
- Elicit users behavioral changes in a positive way

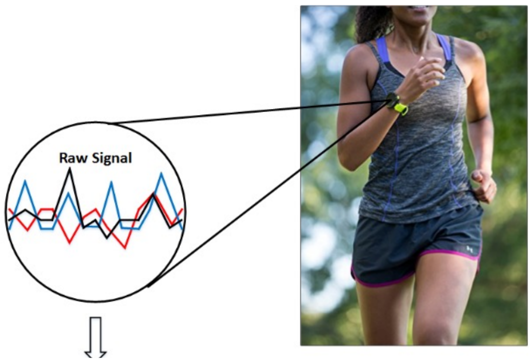


More Applications

- Elder Health Care
- Intelligent Environment
- Security and Surveillance
- Human Computer Interaction
- Indoor Navigation
- Shopping Experience
- Many more…

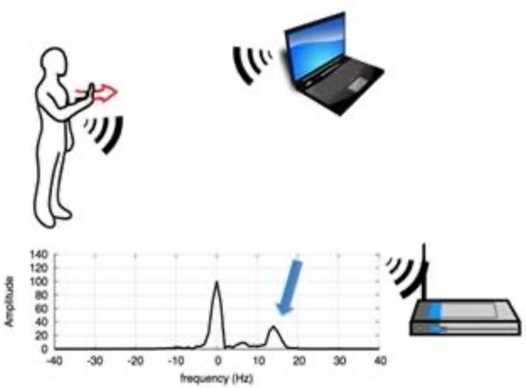


How Do we Monitor Activities?

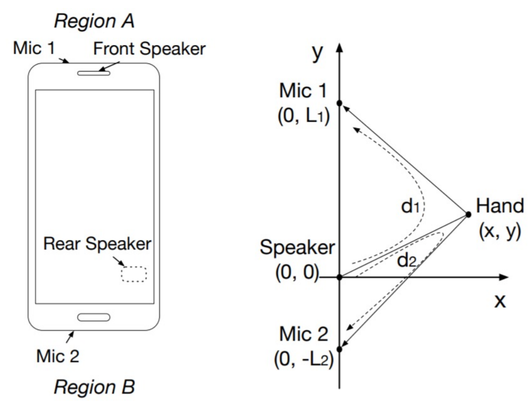


Inertial Sensors

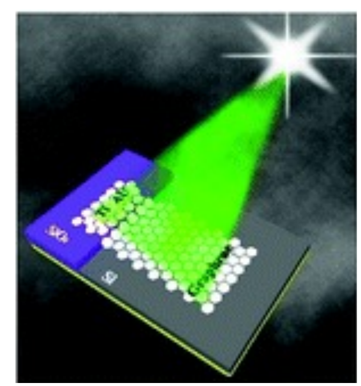
Cameras



Wireless Signals



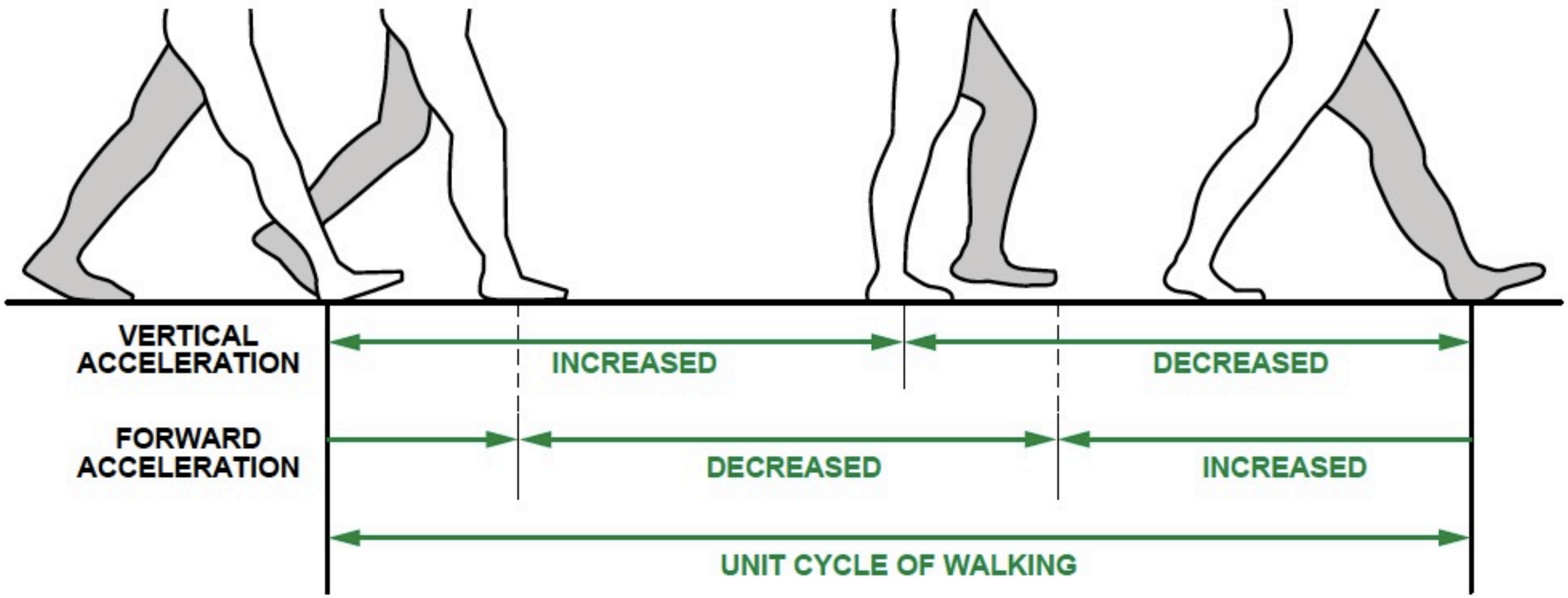
Ultrasounds



Light sensors

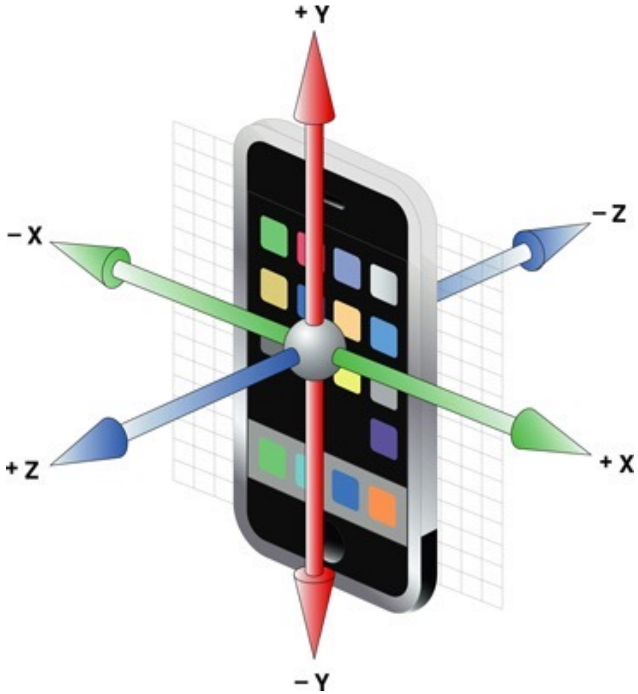
Inertial Sensor-based Approach

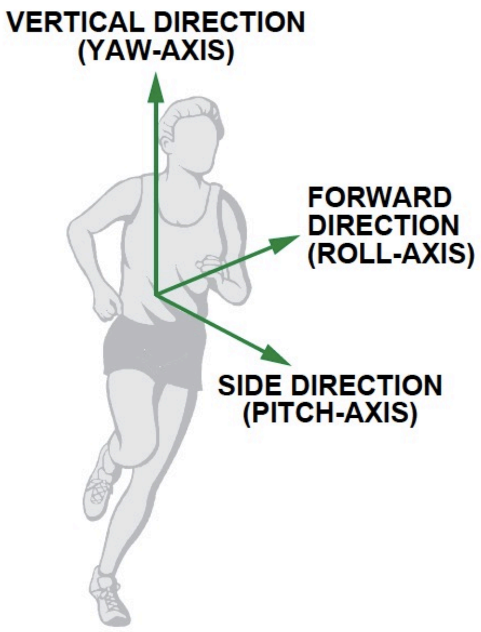
- Activities involve physical movement of body limbs.



Inertial Sensors: Accelerometer

- All commodity smartphones have accelerometers.
- Measure linear acceleration (m/s^2) in three different directions.



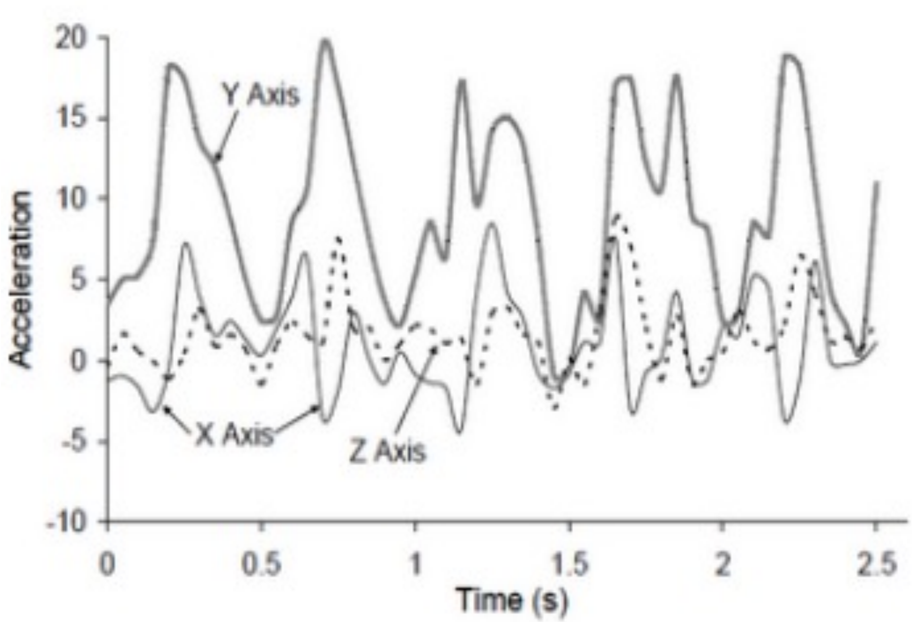


Inertial Sensors: Gyroscope and Compass

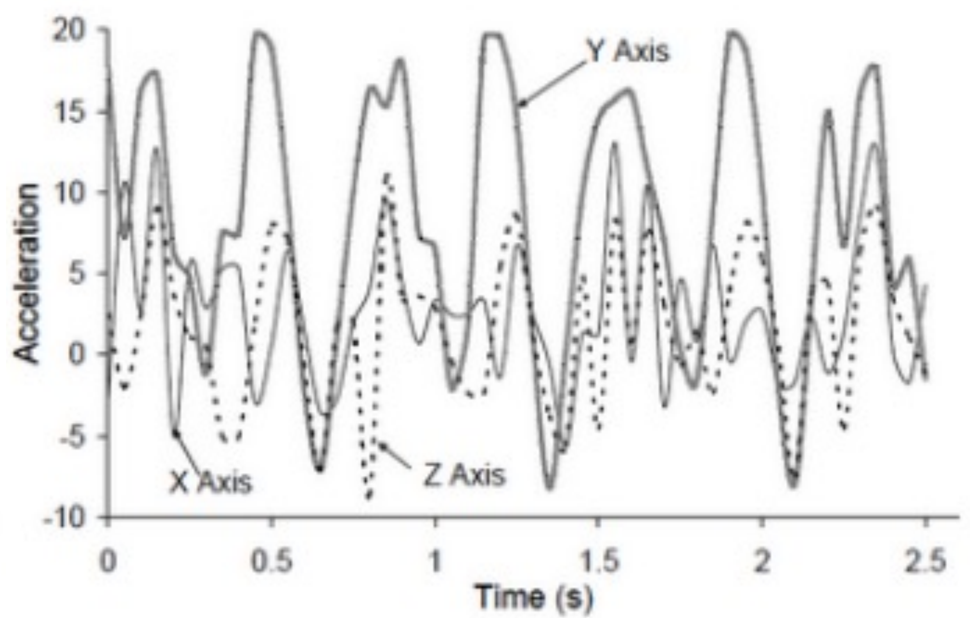
- Gyroscope: measure orientation and angular velocity
- Compass: measure the direction on the earth's surface toward the north.

Acceleration Signals

- Let's say we want to know if a person is walking/running/standing/sitting.
- Assumption: Smartphone is in a front pocket.

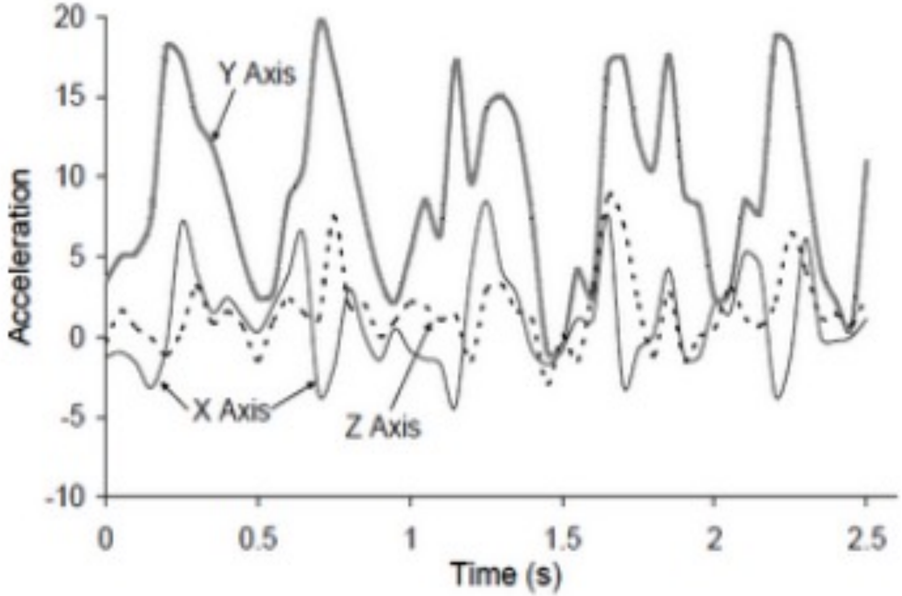


Waking

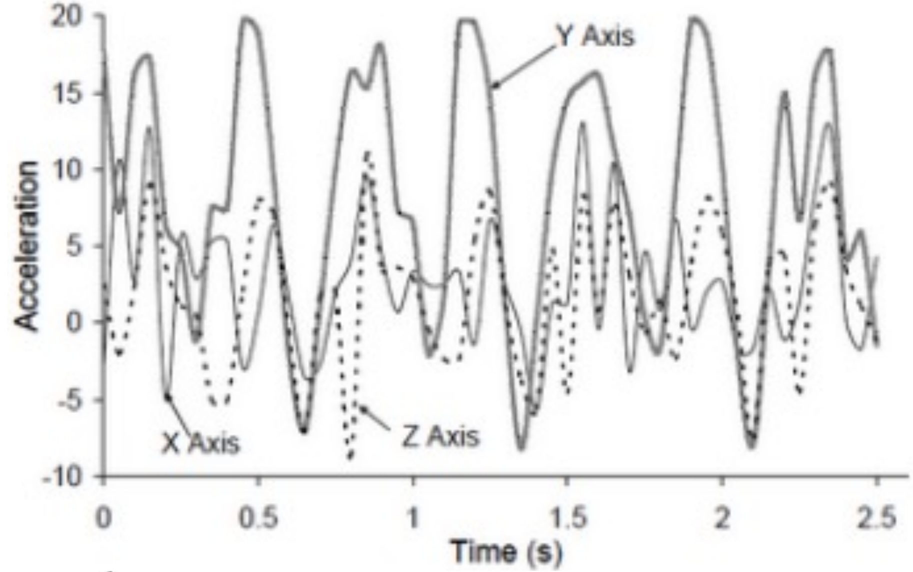


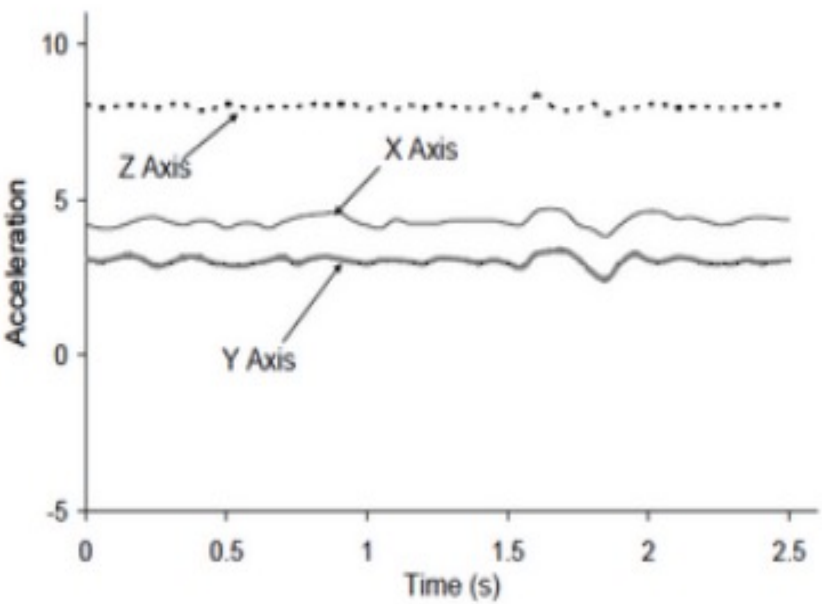
Jogging

Waking

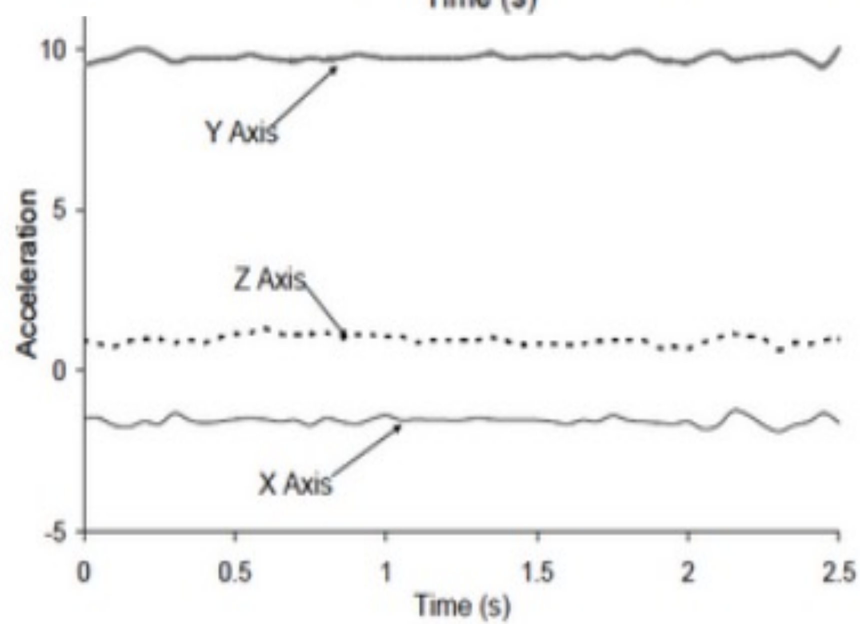


Jogging





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?

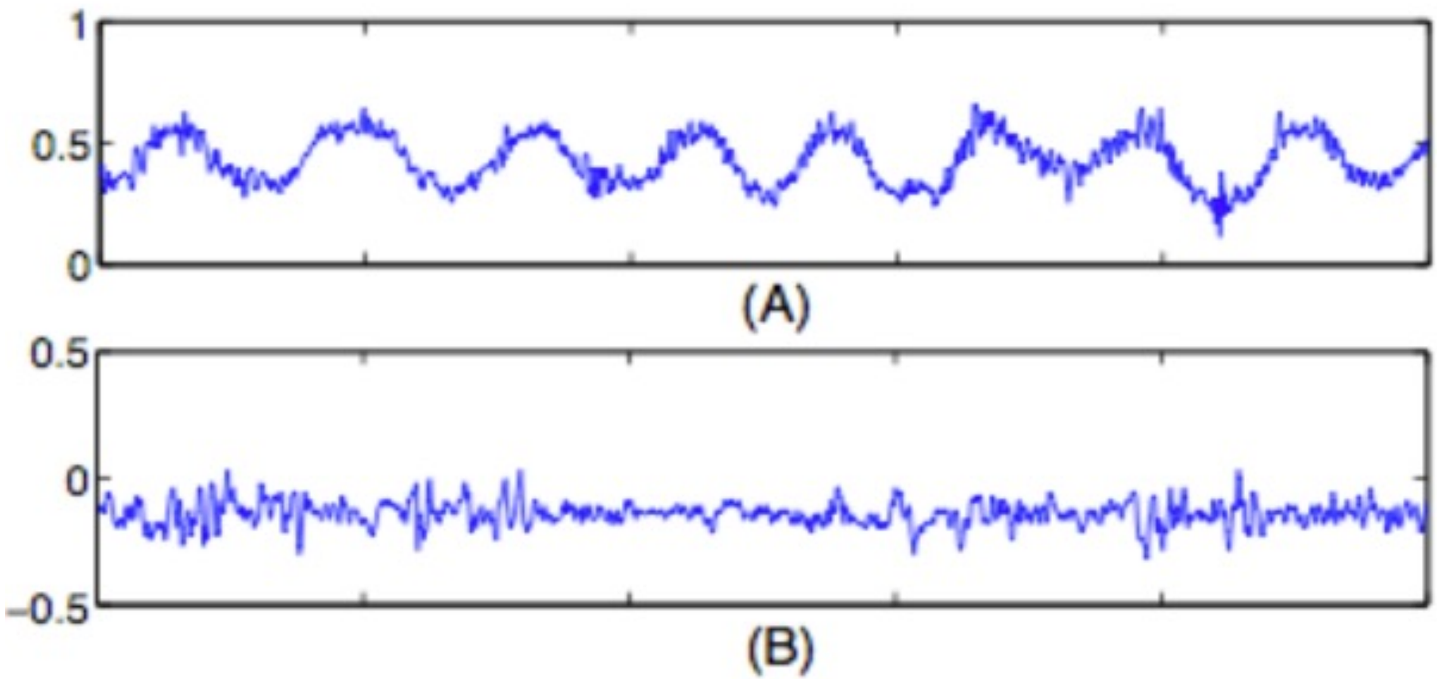
- If STDEV(y-axis samples) < $C_{Threshold1}$
  - ✓ If AVG(y-axis samples) > $C_{Threshold2}$
    - output standing
  - ✓ Else
    - output sitting
- Else
  - ✓ If FFT(y-axis samples) < $C_{Threshold3}$
    - output walking
  - ✓ Else
    - output jogging

Problems of 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

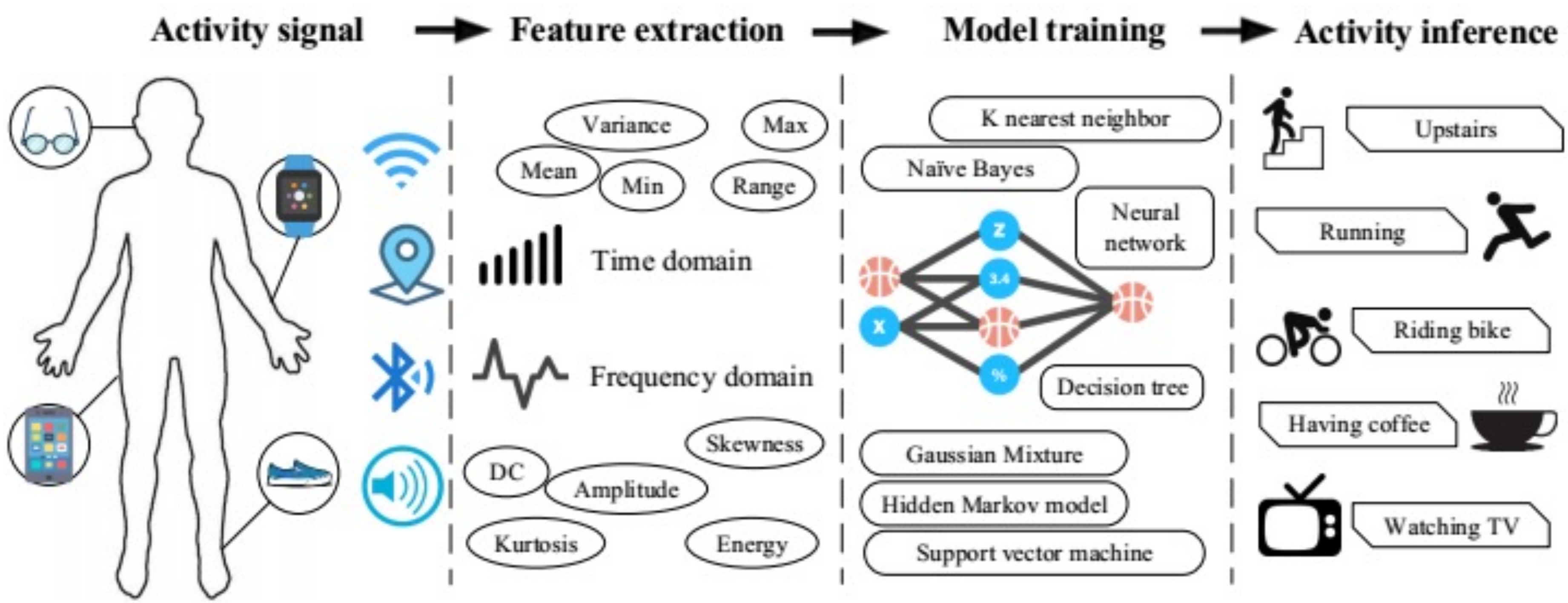


(a) Phone in the pocket

(b) phone in the backpack

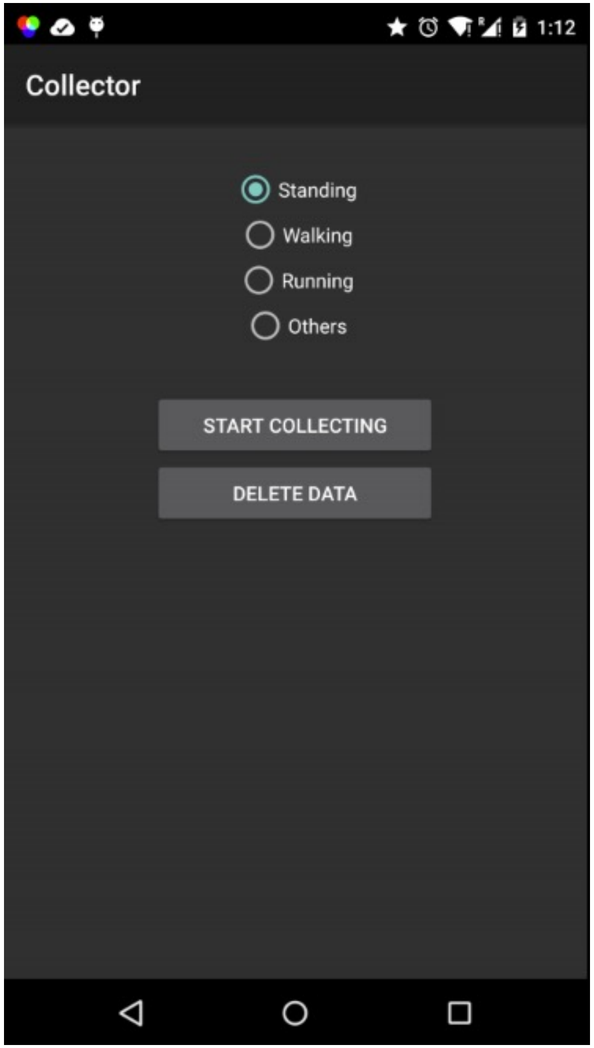
Source: Prof. Andrew Campbell's Lecture Note

Machine Learning Techniques



Step 1: Data Collection

- The first step is to collect labeled acceleration 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.



Public Datasets

- There are many public datasets available.
- Mostly small scale with limited subject and sensor diversity

Dataset	# of sensors	# of motions	# of subjects	Duration (hour)
RealWorld [60]	7	8	15	17.8
PAMAP2 [48]	3	18	9	7.9
DSADS [3]	5	19	8	12.7
mHealth [6, 8]	3	12	10	1.5
REALDISP [7]	9	33	17	39
HHAR [56]	2	6	9	20
OPPORTUNITY [49]	19	16	4	6.1

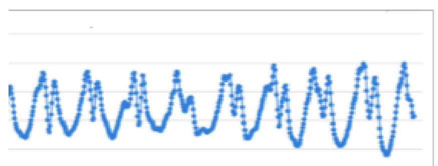
Step 2: Feature Extraction

- 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.

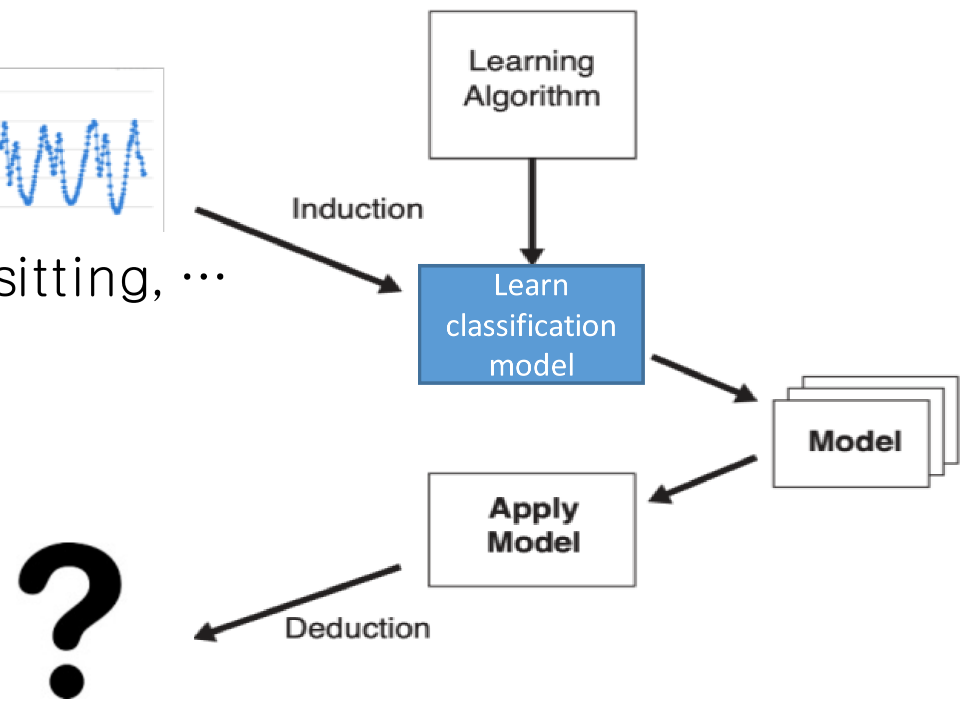
Step 3: Classifier Training

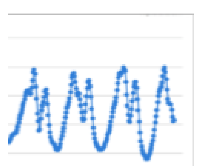
- A classifier identifies which of the features is most useful in distinguishing between the different activities.

Training Data: 

Labels: walking, running, sitting, …

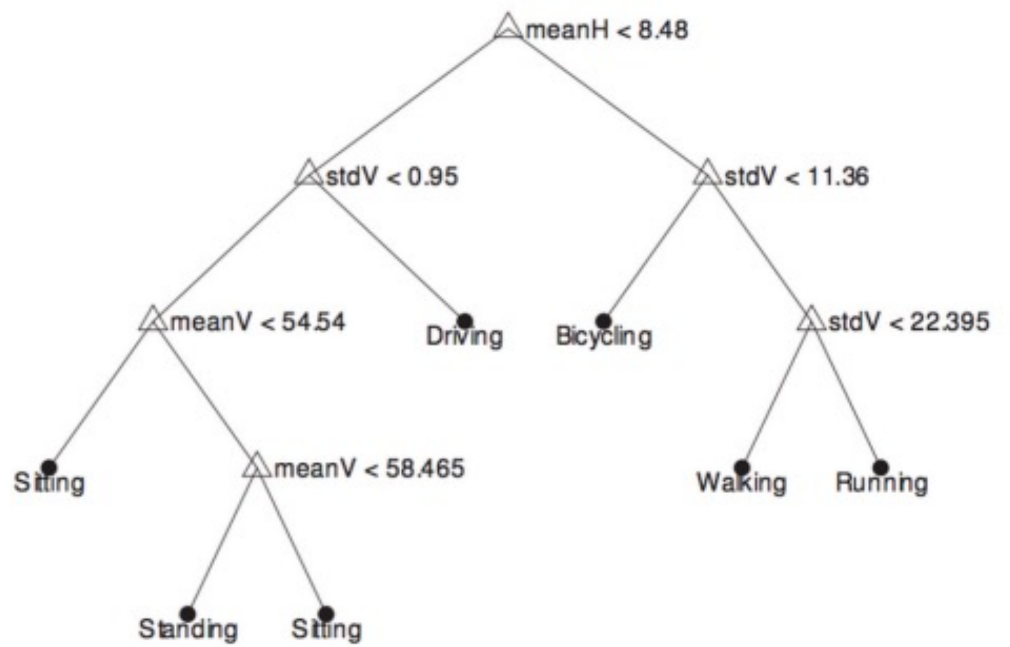
Features to use



New Data 

Decision Tree

- A simple but effective ML classifier.
- Given training data, the algorithm can automatically determine the important features and their thresholds.
  - ✓ See C4.5 algorithm if you are interested.
- 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

Other ML Techniques

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

See More Details in

Activity Recognition from User-Annotated Acceleration Data

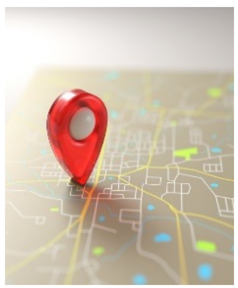
Ling Bao and Stephen S. Intille

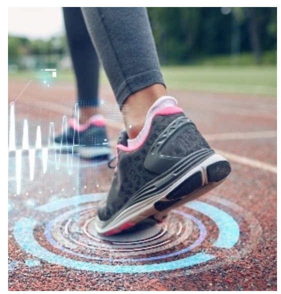
Massachusetts Institute of Technology

Abstract. In this work, algorithms are developed and evaluated to detect physical activities from data acquired using five small biaxial accelerometers worn simultaneously on different parts of the body. Acceleration data was collected from 20 subjects without researcher supervision or observation. Subjects were asked to perform a sequence of everyday tasks but not told specifically where or how to do them. Mean, energy, frequency-domain entropy, and correlation of acceleration data was calculated and several classifiers using these features were tested. Decision tree classifiers showed the best performance recognizing everyday activities with an overall accuracy rate of 84%. The results show that although some activities are recognized well with subject-independent training data, others appear to require subject-specific training data. The results suggest that multiple accelerometers aid in recognition because conjunctions in acceleration feature values can effectively discriminate many activities. With just two biaxial accelerometers – thigh and wrist – the recognition performance dropped only slightly. This is the first work to investigate performance of recognition algorithms with multiple, wire-free accelerometers on 20 activities using datasets annotated by the subjects themselves.

Active Research Directions

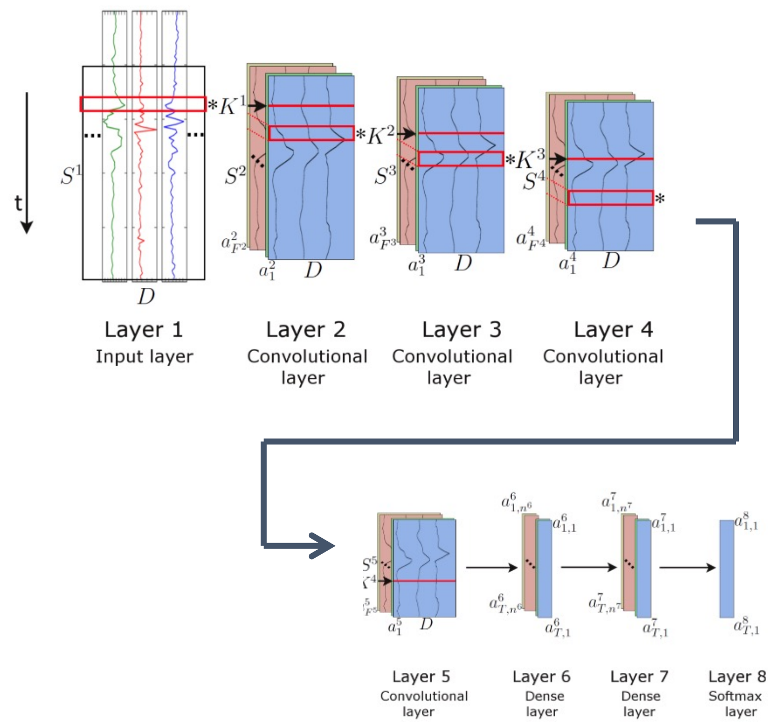
- Use other sensing modalities or and fuse data from other sensing devices (e.g., smart bulbs)





- Detecting a wider variety of activities or fine-granule activities
  - (e.g., eating habits, gait patterns, swimming posture, falls)

- Use of deep learning to enhance recognition accuracy (e.g., CNN + LSTM)



Ensembles of Deep LSTM Learners for Activity Recognition using Wearables [ACM IMWUT '17]

Gesture Recognition

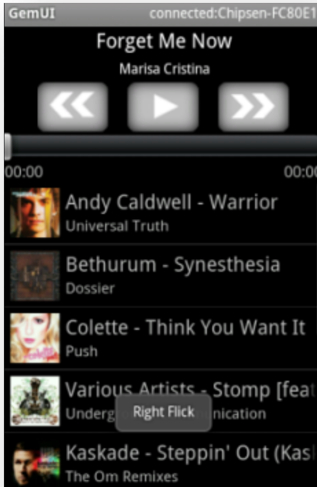
Gesture-Based Interaction

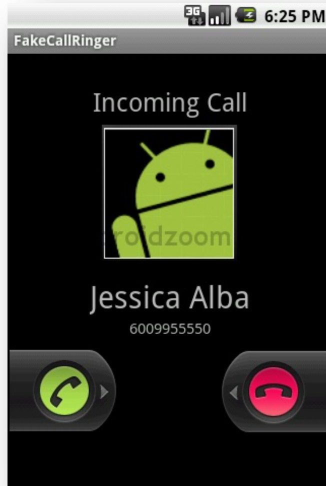
- Gestures are a natural way of interacting with object and other people.

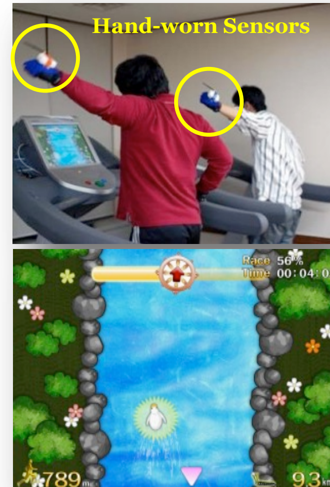
- Gestures can be particularly useful when other forms of interactions are difficult.
  - ✓Controlling a phone while running
  - ✓Communicating with impaired people

- It can be used as complement to other types of interaction modalities.

Applications

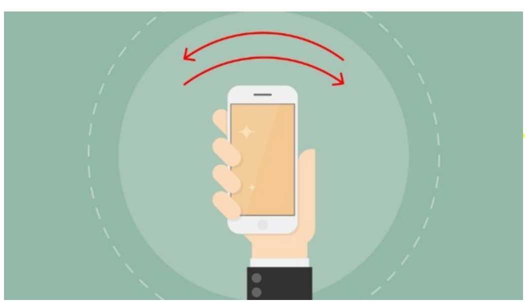


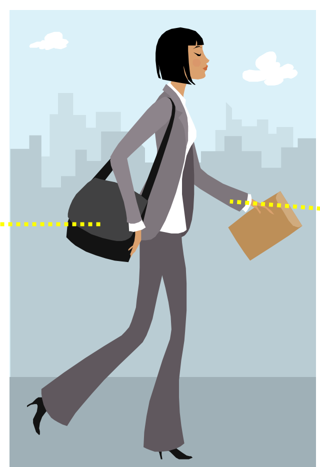




…

Gesture Interaction



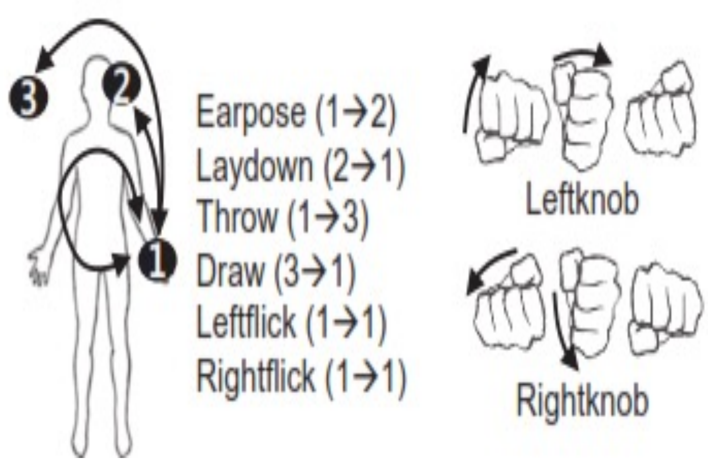


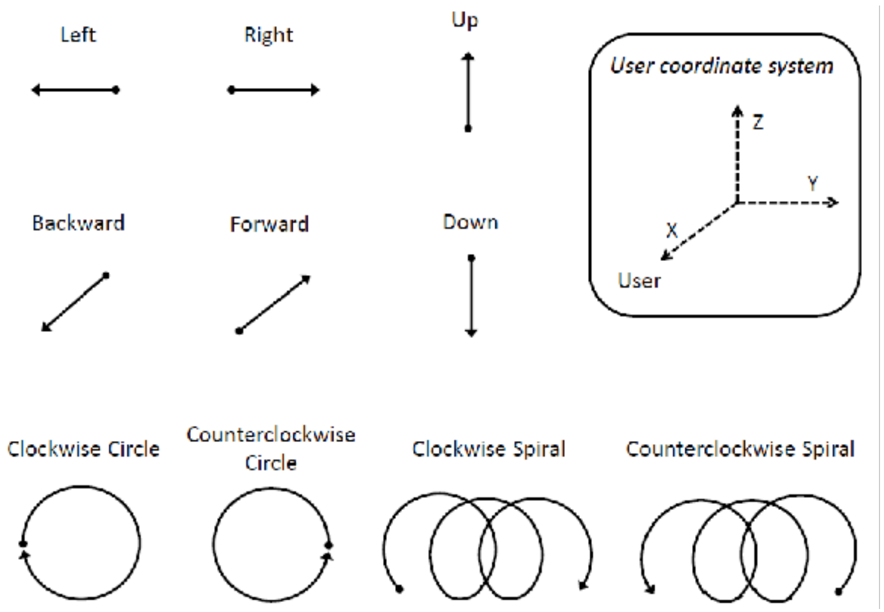
Smartphone

smartwatch

Example Gesture Vocabulary

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





Technical Approaches

- Similar to activity recognition, many different sensors and algorithms have been studied.
  - ✓Vision-based
  - ✓Sound-based
  - ✓Motion-based
  - ✓Wireless signal-based

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

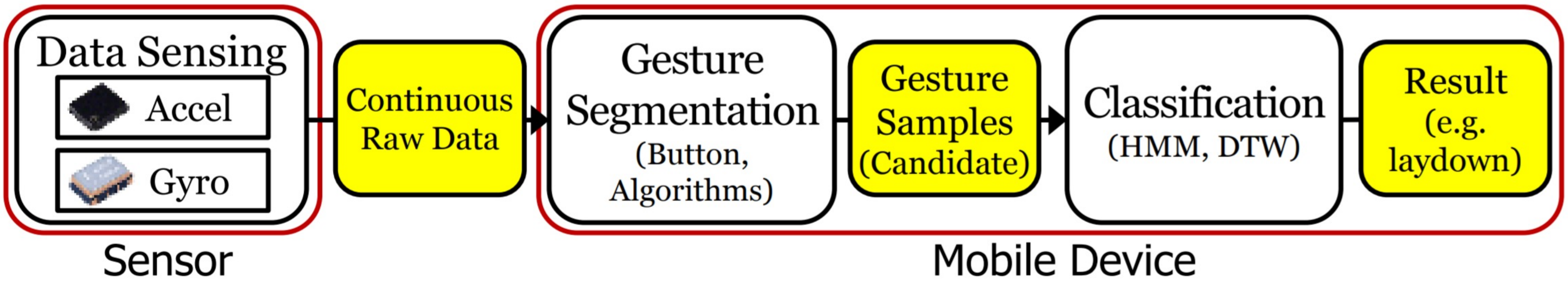
E-Gesture: A Collaborative Architecture for Energy-efficient Gesture Recognition with Hand-worn Sensor and Mobile Devices

Taiwoo Park et al.

ACM SenSys 2011

Gesture Recognition Pipeline

- The paper aims at detecting and classifying various hand gestures continuously using a smartwatch.
- The default gesture recognition pipeline is as below:
  - ✓Smartwatch: Data source
  - ✓Smartphone: Gesture recognizer



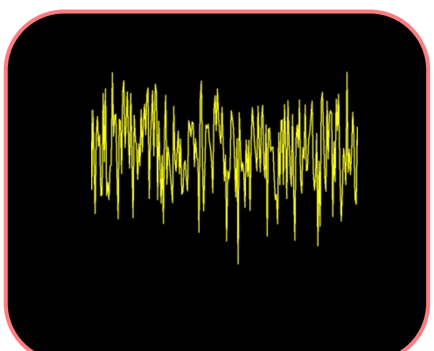
Challenges

- Providing Energy-efficient Gesture Processing

20hrs **Sensor, 250mAh**

24hrs → 17hrs **Smartphone**

- Accurately Segmenting and Classifying 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

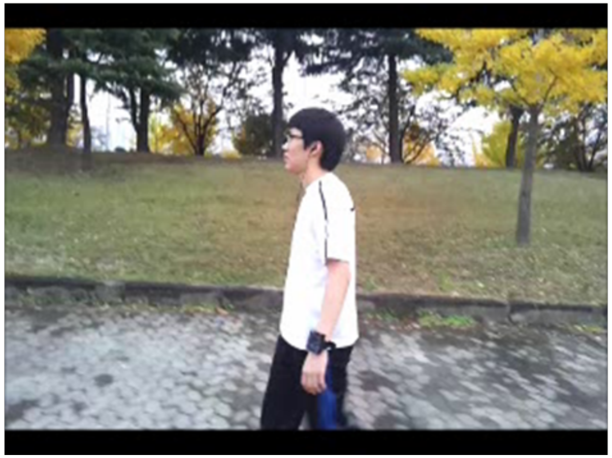
- Designed energy-efficient, mobility-robust gesture detection architecture
  - ✓Triggering gyroscope by analyzing accelerometer signal
  - ✓Adjusting accelerometer sensitivity by gyroscope validation

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

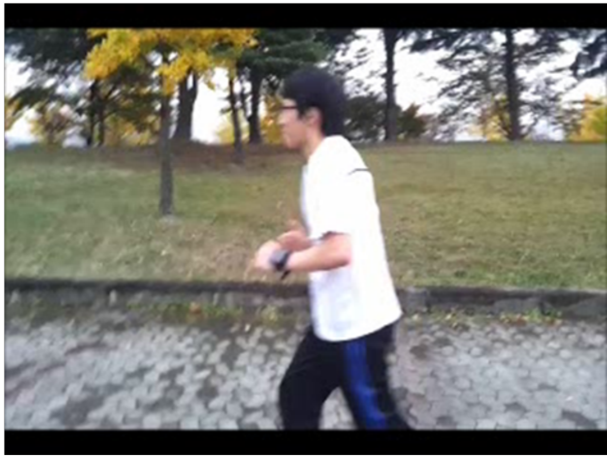
Mobility Noises

- Makes it difficult to distinguish intended hand motions from noises

Standing still

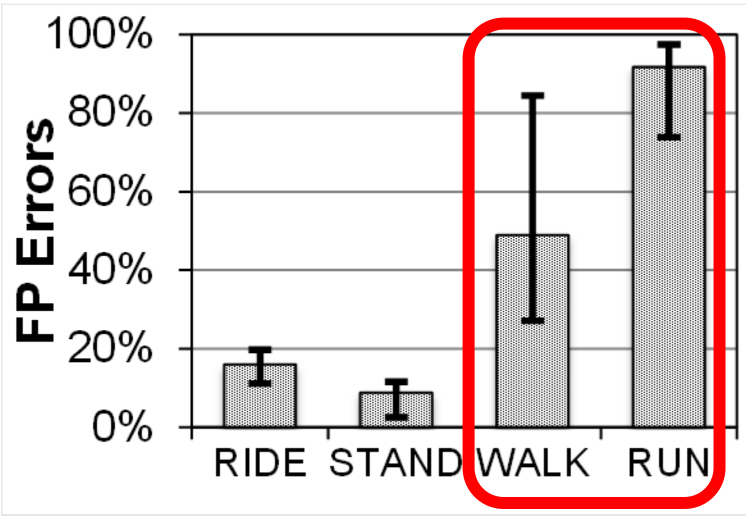


Walking

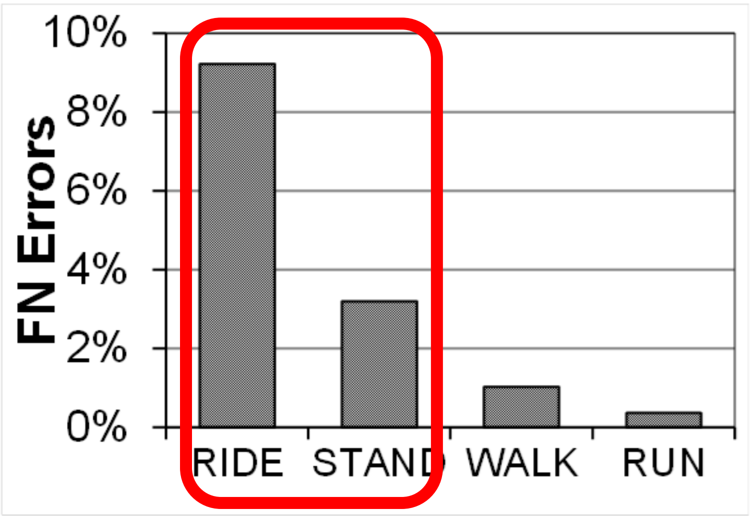


Running

Gesture Segmentation: Accel



Lower fixed threshold
➔ False-positives
on high mobility



Higher fixed threshold
➔ False-negatives
on low mobility

Gesture Segmentation: Gyro

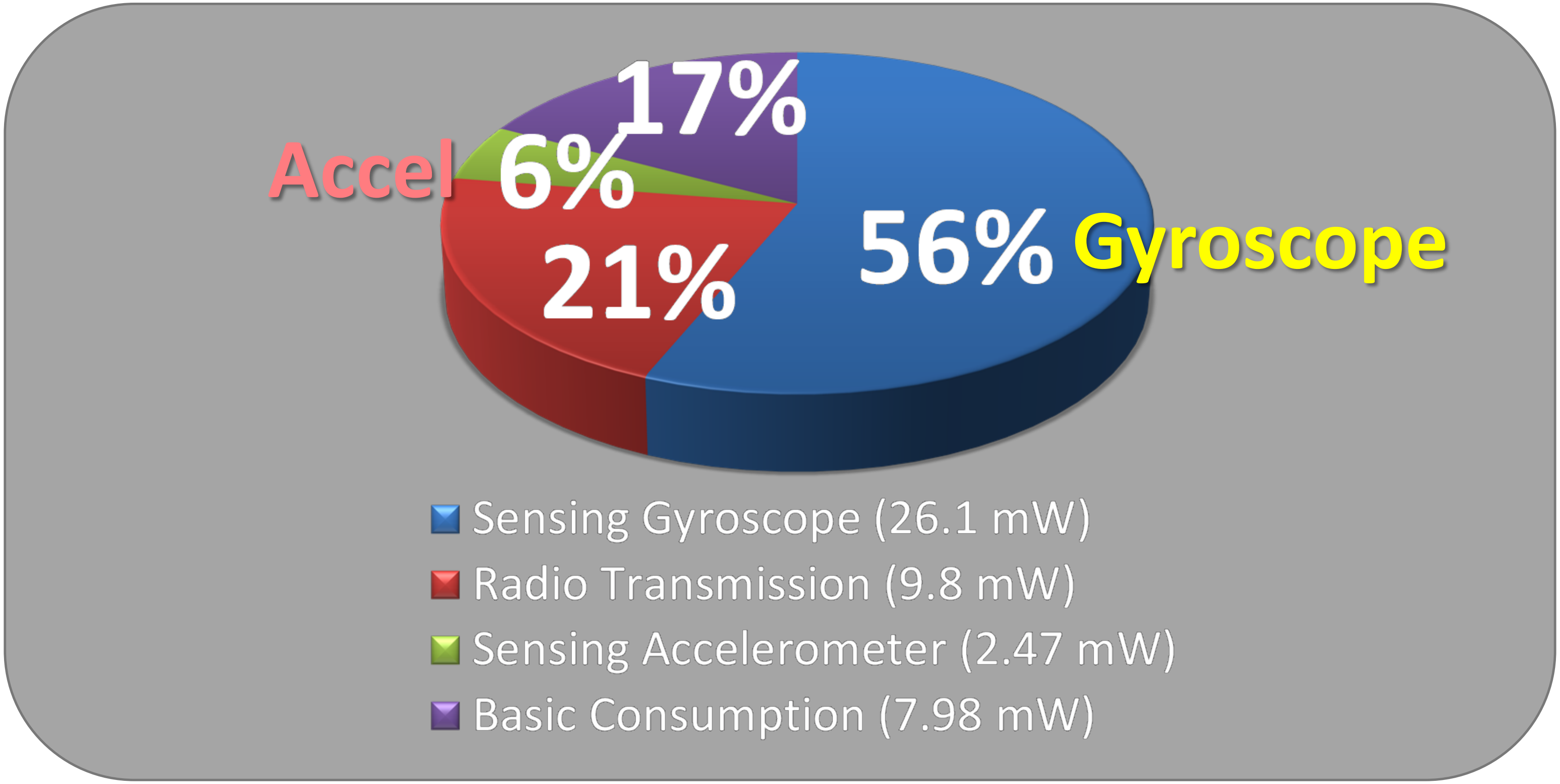
- Accelerometer is more sensitive to mobility
- Gyroscope is more robust to mobility

	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

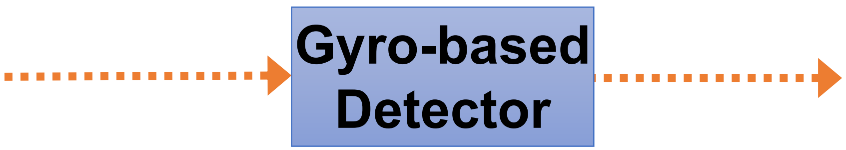
- Accelerometer: mobility-sensitive, energy-efficient
- Gyroscope: mobility-robust, energy-hungry

	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' mobility patterns (based on HMM)

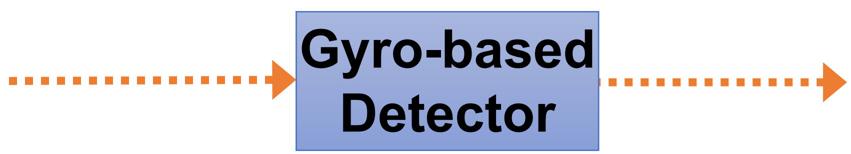
Closed-loop Collaborative Segmentation



Accurate, High energy

Gyro-based Detector

Closed-loop Collaborative Segmentation



Accurate, High energy

Gyro-based Detector

Accurate, High energy
because of mobility

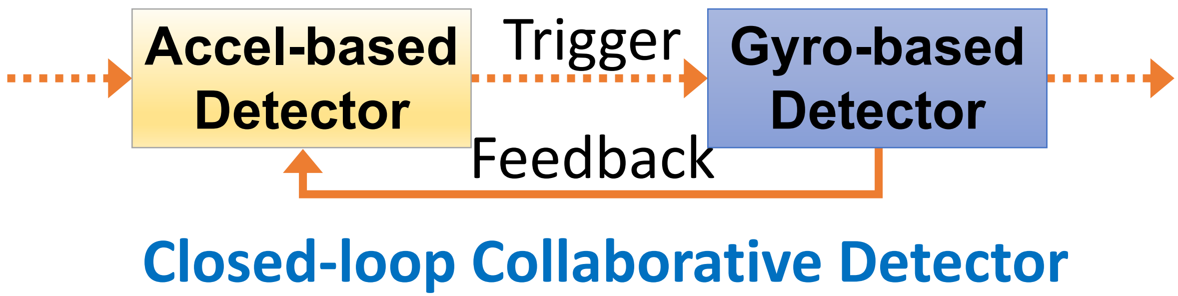
Open-loop Detector

Closed-loop Collaborative Segmentation

Closed-loop Collaborative Detector

Accurate, Low energy

Closed-loop Collaborative Segmentation



Accurate,
Low energy

Performance-preserving, Energy-saving Collaborative Sensor Fusion

Approaches

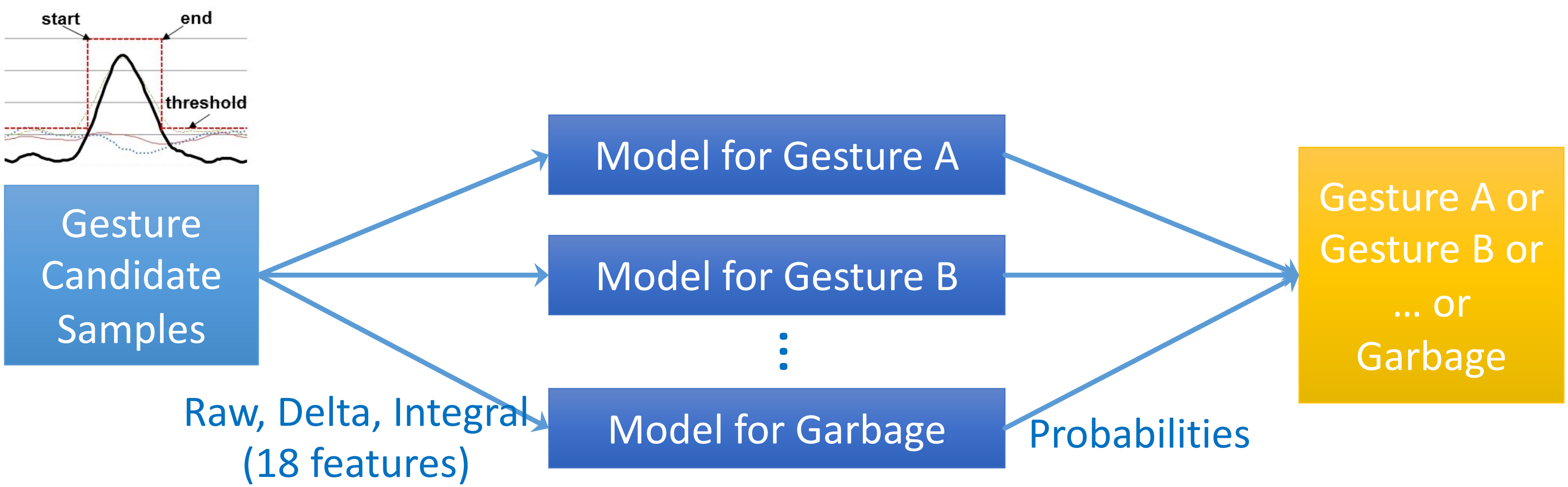
- Investigated characteristics of Accel and Gyro
  - ✓ Accelerometer: Mobility-Sensitive, Energy-Efficient
  - ✓ Gyroscope: Mobility-Robust, Energy-Hungry

- 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' mobility patterns (based on HMM)

Basic HMM

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



8-state left-right HMMs

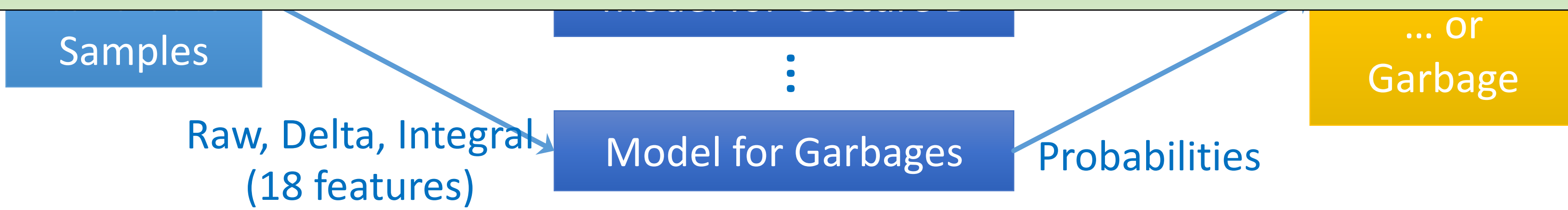
Basic HMM

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

Design alternatives:

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

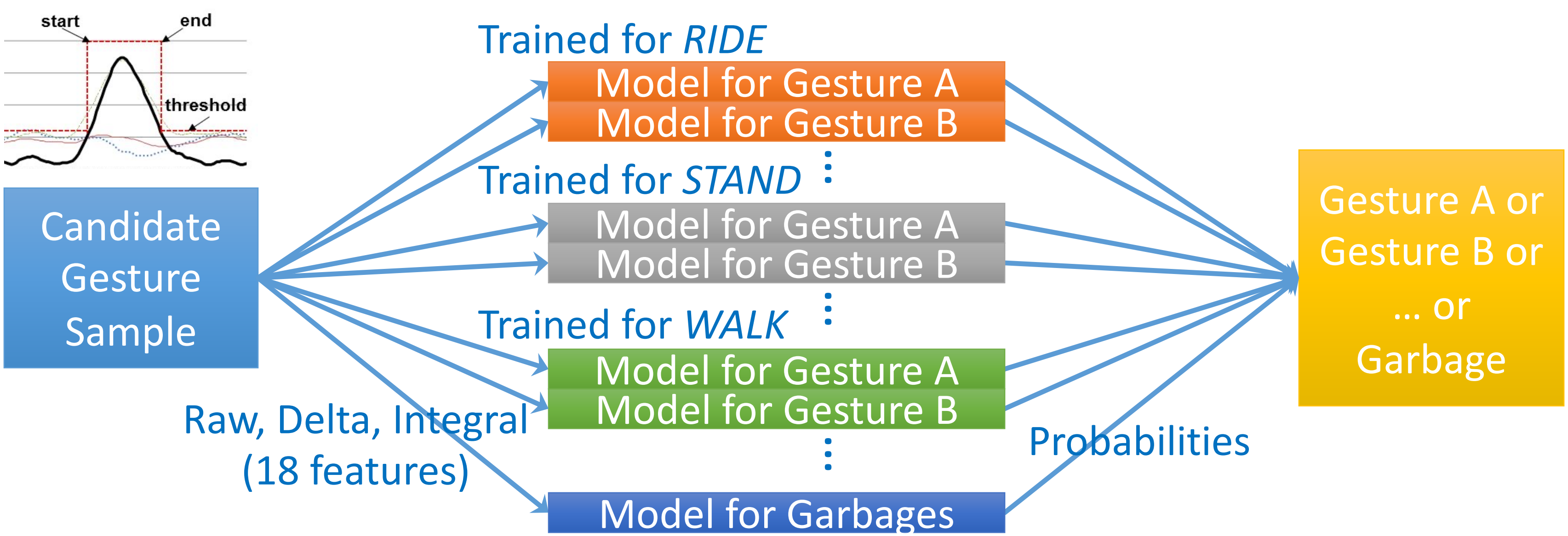
2) Train several different models for predefined set of mobility situations



8-state left-right HMMs

Multi-Situation HMM

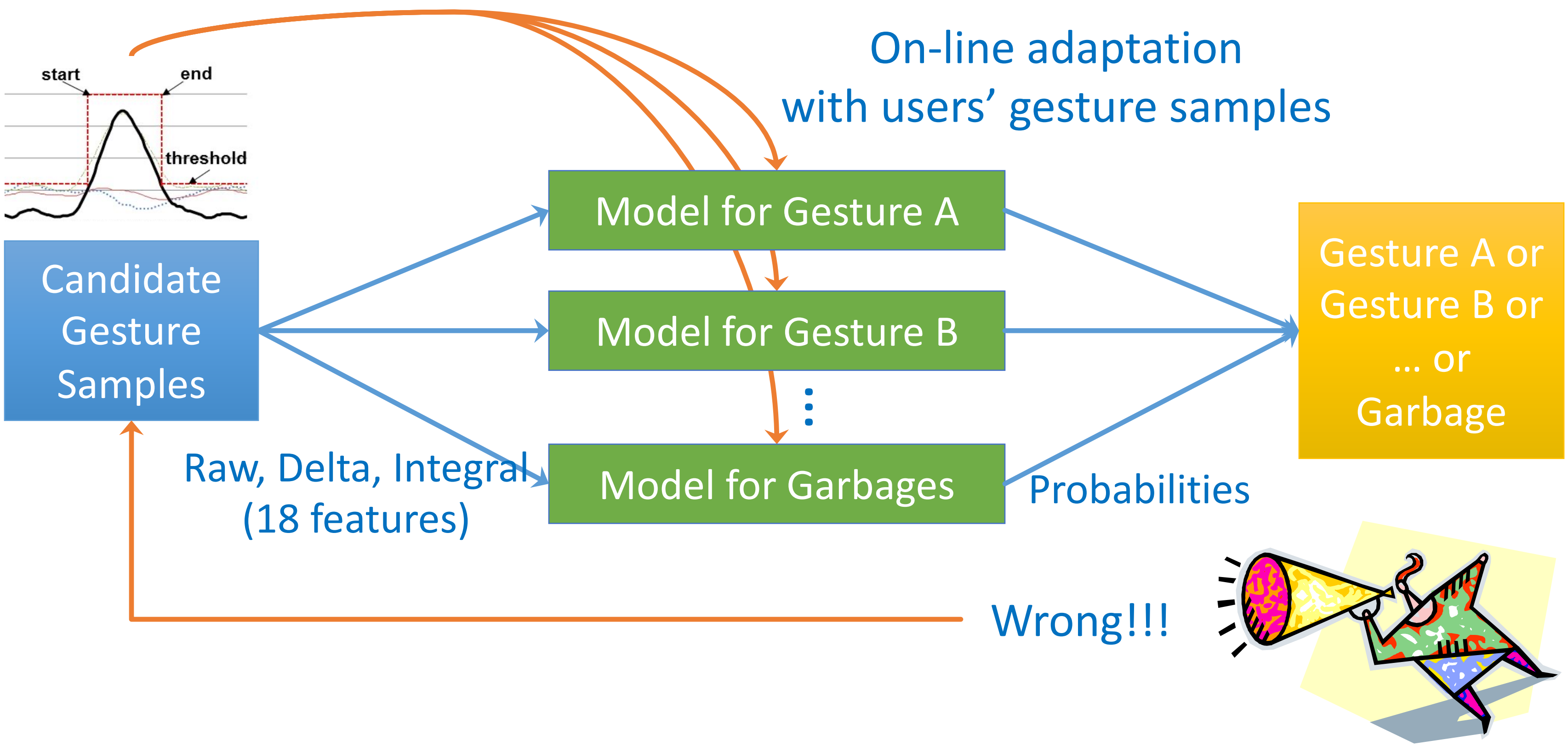
- Train models separately for representative mobility situations
  - ✓ e.g. Riding a car, Standing, Walking, Running
- Classify by evaluating all models



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

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