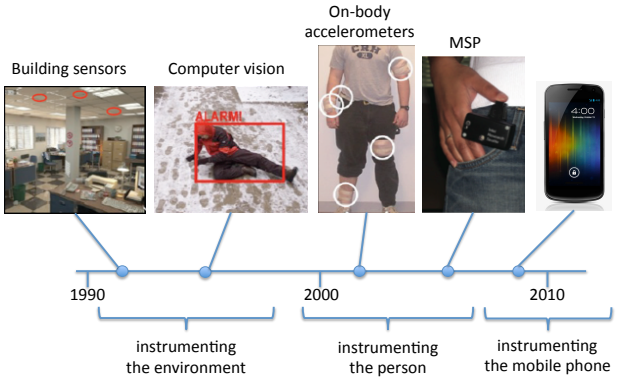
Mobile Phone Sensing

Dr. Christos Efstratiou

UNIVERSITY OF CAMBRIDGE

History of Sensing Platforms



UNIVERSITY OF CAMBRIDGE

History
Mobile Phone Sensing

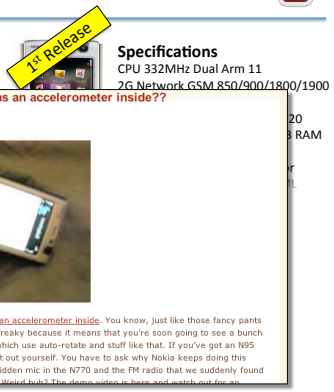
- Phone manufacturers never intended their devices to act as general purpose sensing devices
- Sensing components were only considered as tools to facilitate interaction with the phone
  - Accelerometer: Screen rotation
  - Gyro: games
  - Microphone: making calls ☺

Specifications
CPU 332MHz Dual Arm 11
2G Network GSM 850/900/1800/1900
3G Network HSDPA 2100
Display TFT, 16M colors, 240x320
Memory 160MB storage, 64MB RAM
GPS
GPU 3D Graphics HW acclerator
Browser WAP 2.0/xHTML, HTML

UNIVERSITY OF CAMBRIDGE

History
Mobile Phone Sensing

- Phone manufacturers never intended their devices to act as general purpose se
- Sensing componen considered as tools interaction with th
  - Accelerometer: S
  - Gyro: games
  - Microphone: ma



Specifications
CPU 332MHz Dual Arm 11
2G Network GSM 850/900/1800/1900

UNIVERSITY OF CAMBRIDGE

History
Mobile Phone Sensing

- Phone manufacturers never intended their devices to act as general purpose sensing devices
- Sensing components were only considered as tools to facilitate interaction with the phone
  - Accelerometer: Screen rotation
  - Gyro: games
  - Microphone: making calls ☺

Specifications
CPU 332MHz Dual Arm 11
2G Network GSM 850/900/1800/1900
3G Network HSDPA 2100
Display TFT, 16M colors, 240x320
Memory 160MB storage, 64MB RAM
GPS
GPU 3D Graphics HW acclerator
Browser WAP 2.0/xHTML, HTML

Specifications
CPU 332MHz Dual Arm 11
2G Network GSM 850/900/1800/1900
3G Network HSDPA 2100
Display TFT, 16M colors, 240x320
Memory 160MB storage, 64MB RAM
GPS
Accelerometer
GPU 3D Graphics HW acclerator
Browser WAP 2.0/xHTML, HTML

UNIVERSITY OF CAMBRIDGE

History
Mobile Phone Sensing

- The mobile phone sensing domain is filled with "hacks", and imaginative techniques that were used to circumvent the limitations of a platform that was **designed for a different purpose**.
- However, manufacturers have started to change direction
  - In the near future we expect the release of
    - New hardware platforms that facilitate back-ground sensing
    - New OS frameworks that incorporate a general purpose sensing middleware

UNIVERSITY OF CAMBRIDGE

Phone Sensing vs Sensor Networks

Sensor Networks

- Well suited for sensing the environment
- Specialized hardware designed to accurately monitor specific phenomena
- All resources dedicated to sensing
- High cost of deployment and maintenance (regular recharging thousands of sensor nodes)

Phone Sensing

- Well suited for sensing human activities
- General purpose hardware, often not well suited for accurate sensing of the target phenomena
- Multi-tasking OS. Main purposed of the device is to support other applications
- Low cost of deployment and maintenance (millions of potential users where each user charges their own phone)

But not sure if users will keep you app on their device!

UNIVERSITY OF CAMBRIDGE

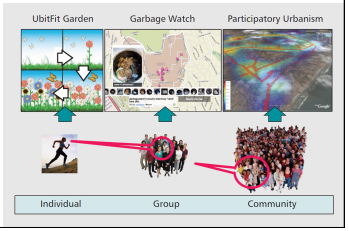
Sensors

- Microphone
- Camera
- GPS
- Accelerometer
- Compass
- Gyroscope
- WiFi
- Bluetooth
- Proximity
- Light
- NFC (near field communication)

UNIVERSITY OF CAMBRIDGE

Applications

- **Individual activity sensing**: fitness applications, behavioural suggestions.
- **Group activity sensing**: groups to sense common activities and help achieving group goals. Eg: assess neighbourhood safety, collective recycling efforts.
- **Community sensing**: large scale sensing, where large number of people have the same application installed. E.g., tracking speed of disease across a city, congestion in city.



Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, Andrew T. Campbell, A Survey of Mobile Phone Sensing, IEEE Communications Magazine, September, 2010.

UNIVERSITY OF CAMBRIDGE

Applications
Physical Activity

- Example Inferences:
  - {walking, running, up/down stairs}
- Sensors used: accelerometer, gyroscope, compass
- Applications:
  - Health – Calorie Tracking
  - "Presence sharing"

Source: "Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application" – SENSYS '08

Source: "Flowers or a Robot Army? Encouraging Awareness & Activity with Personal, Mobile Displays" – UBICOMP '08

Applications
Transportation Mode

- Example Inferences: {bike, bus, car}
- Sensors Used: accelerometer, gps, wifi, (location tech.)
- Applications:
  - Green Transportation
  - Smart Commuting

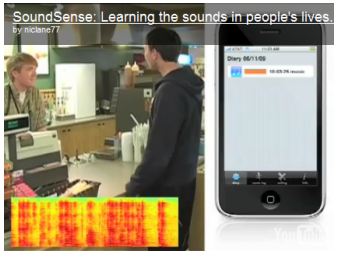
Example: "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research" –MOBISYS '09

UNIVERSITY OF CAMBRIDGE

Applications
Context and Environment

- Example Inferences:
  - {voicing, music, party, activity-related sounds}
- Sensors Used: microphone, camera
- Applications:
  - Automated Diary
  - Health & Wellness

Source: "SoundSense: scalable sound sensing for people-centric applications on mobile phones" –MOBISYS '09



UNIVERSITY OF CAMBRIDGE

Applications
Human Voice and Conversations

- Example Analysis:
  - Turn-taking, Stress, Speaker Dominance
- Sensors Used: microphone
- Applications:
  - Social network analysis
  - Stress

Conversation Network

UNIVERSITY OF CAMBRIDGE

Source: "Towards the Automated Social Analysis of Situated Speech Data" – UBICOMP '08

Applications
Detecting Emotions

- Example inference:
  - Emotional state, location and co-location with others
- Sensors used:
  - Microphone, bluetooth, GPS
  - Map speaking features to emotional state
- Application:
  - EmotionSense

Source: "EmotionSense: A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research" –Ubicomp'10

UNIVERSITY OF CAMBRIDGE

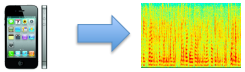
Mobile Phone Sensing Design

- Typical mobile phone sensing applications follow a common design pattern
  - Collect raw data using the sensors of the mobile phone
  - Infer a particular activity of interest using the sensor values (e.g. physical activity: is the user running? Context detection: is the user in a place full of other people?)
  - Expose the high-level result to the user or use that result to adapt the behaviour of the application

UNIVERSITY OF CAMBRIDGE

Development Design Patterns

- Collect data (labelled or unlabelled)



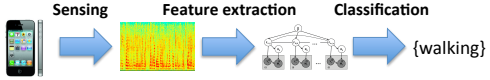
- Inference pipeline

- Mobile Sensing App
  - Extras: storage, networking, sharing, privacy

UNIVERSITY OF CAMBRIDGE

Development Design Patterns

- Collect data
  - High sampling rate
  - Label with ground truth (e.g. user walking → data set)
- Inference pipeline
  - Use collected data for training



- Mobile Sensing App
  - Extras: storage, networking, sharing, privacy

UNIVERSITY OF CAMBRIDGE

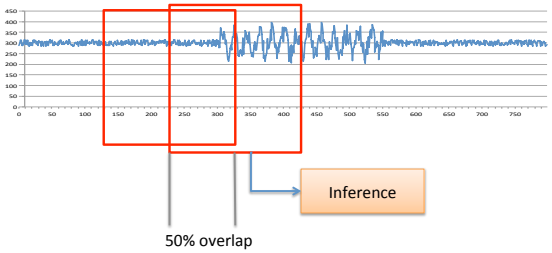
Sensing

- Sensing is resource intensive

- The mobile phone's purpose is to support multiple applications
- A mobile phone sensing application needs to maintain a balance between
  - The amount of resources needed to operate
  - The accuracy of the detection that is achieved

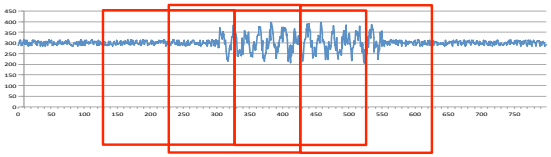
UNIVERSITY OF CAMBRIDGE

Sensing Modes
Continuous Sensing



UNIVERSITY OF CAMBRIDGE

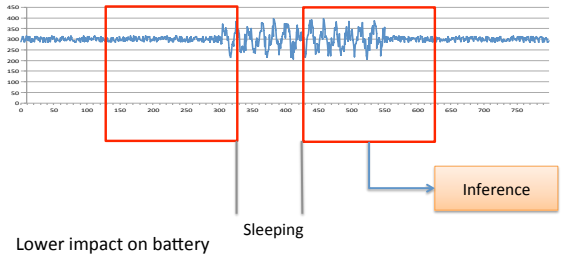
Sensing Modes
Continuous Sensing



- Highly accurate data
- Very costly in terms of battery and CPU usage
  - Continuous sensing on multiple sensors can reduce phone stand-by to 6 hours
  - May be used on "cheap" sensors e.g. accelerometer

UNIVERSITY OF CAMBRIDGE

Sensing Mode
Duty Cycling

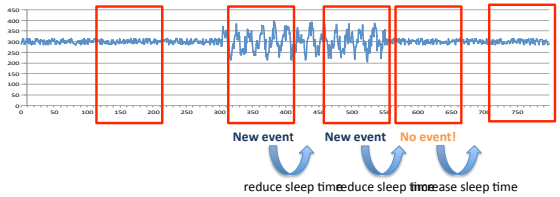


- Lower impact on battery
- Less accurate, interesting events may take place during sleeping

UNIVERSITY OF CAMBRIDGE

Adaptive Duty Cycling

- Adjust the duration of sleeping periods according the rate of events that are detected
- If no events are detected sleep for longer
- When new events are detected reduce the sleeping time



UNIVERSITY OF CAMBRIDGE

Adaptive Duty Cycling

- Typical approaches (depending on the type of events)
  - Exponential increase – linear decrease
  - Linear increase – exponential decrease
- Reduces the energy cost (compared to continuous sampling)
- Maintains high accuracy (compared to duty cycling)

But

- Requires a good understanding of the application domain
  - in conversation detection a new voice events may be followed immediately by more such events, so faster sleep-time decrease may be necessary
  - a location change event may not be followed by an immediate new event, so slower decrease may be applicable

UNIVERSITY OF CAMBRIDGE

Inference

- The process of mapping raw sensor data to meaningful high-level events
- Inference Pipeline

- Designing an Inference Engine
  - Collecting raw sensor data, typically labelled with ground truth information.
  - Data set should also cover states we are not trying to detect but look similar (e.g. detect *walking* : we need data also for *running* and *standing*)
  - Train the inference engine with the collected data
  - Applying the inference engine to the target application

UNIVERSITY OF CAMBRIDGE

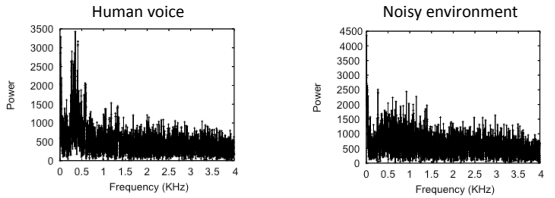
Feature Extraction

- Identifying features in a data set that can be used to infer a particular type of activity
- The set of selected features depends on the type of sensor and the type of activity that is detected
- The design process typically involves off-line analysis of training data to identify the right features for the particular inference engine
  - Usually an iterative process where different features are tested
- Examples:
  - Conversation detection
  - Physical activity detection

UNIVERSITY OF CAMBRIDGE

Feature Extraction
Conversation Detection

- Applying FFT on the audio samples, and comparing training data that are labelled as "conversation" and "non-conversation noise"

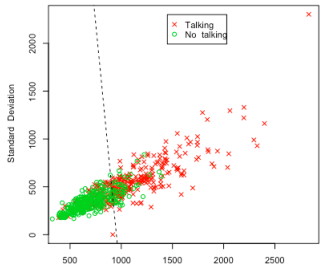


- Sound samples of human voice present most of their energy within the 0-4 KHz spectrum

UNIVERSITY OF CAMBRIDGE

Feature Extraction
Conversation Detection

- Selecting as Features the mean and standard deviation of the FFT power

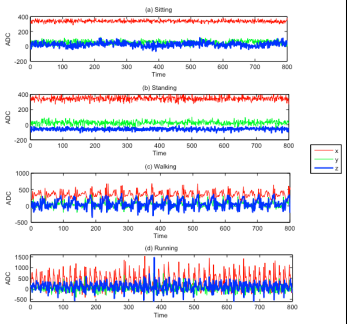


- Using a simple threshold line, could give a relatively accurate detection (with a high number of false positives, however)

UNIVERSITY OF CAMBRIDGE

Feature Extraction
Physical Activity using Accelerometer

- Sensor: accelerometer
- Activities: sitting, standing, waking, running
- Features
  - Mean (can help distinguish between standing and sitting)
  - Standard deviation
  - Number of peaks (can help distinguish between waking and running)



UNIVERSITY OF CAMBRIDGE

Classification

- Feature extraction produces a feature vector.
- The classification matches the feature vector to a pre-defined set of high-level classes.
- The classification engine is typically based on machine-learning techniques and is trained using labelled training data.
- Common classification algorithms include
  - Naive Bayes classifier
  - Decision Trees
  - Hidden Markov Models

UNIVERSITY OF CAMBRIDGE

Naive Bayes classifier

- Given a set of features $F_1,...,F_n$ and a classifier C estimate the probability

$$p(C|F_1,...,F_n)$$

- This can be approximated as

$$p(C|F_1,...,F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

Where Z is a constant (scaling factor) and can be ignored in comparisons

- Using the training dataset we estimate the distributions $p(F_i|C)$
- During runtime, given a set of values for the features $f_1,...,f_n$ we select a classifier that maximizes

$$p(C=c)\prod_{i=1}^{n} p(F_i = f_i|C=c)$$

UNIVERSITY OF CAMBRIDGE

Classification Example

- Trying to detect **walking** and **running** activities using accelerometer
- We collected 8 data sets labelled with the right class
- We select as features:
  - F1: mean acceleration
  - F2: standard deviation
- We need to calculate the distributions

$$p(F_i|C=c_j)$$

for each feature and class

Training data set

F1 Mean	F2 StdDev	Class
384.68	52.31	walking
410.24	114.39	running
392.21	71.26	walking
383.04	61.11	walking
375.32	91.01	running
399.52	109.32	running
377.36	83.01	walking
395.01	78.34	running

UNIVERSITY OF CAMBRIDGE

Classification Example

- We assume Gaussian distributions and therefore we can characterise the distributions using the mean and variance for all combinations

	Mean F1	Var F1	Mean F2	Var F2
walking	384.32	28.12	66.92	131.23
running	395.02	160.00	98.27	207.97

- With these calculations, given a new set of values for F1 and F2 we can estimate the probability that the user is walking or running
- Under the Gaussian distribution assumption this is given by

c_w:walking
μ_w:mean
σ_w^2:variance

$$P(F_1 = x|c_w) = \frac{1}{\sqrt{2\pi\sigma_w^2}} e^{-\frac{(x-\mu_w)^2}{2\sigma_w^2}}$$

UNIVERSITY OF CAMBRIDGE

Classification Example

- The classifier is ready and we can run it in our application
- A new sensor sample is analysed and features are extracted
- Assume a new input with features F_1 = 391.2 and F_2=58.5
- The classifier calculates

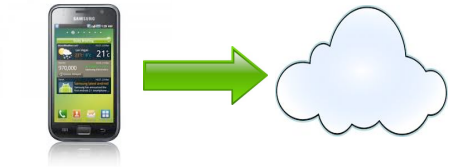
$$p(C = walking | f_1, f_2) = 1.21e-03$$
$$p(C = running | f_1, f_2) = 2.71e-05$$

and selects the class with the highest probability: **walking**

UNIVERSITY OF CAMBRIDGE

Inference Optimizations

- Adaptive sampling can bring down the energy cost but inference can also be costly
  - Example: running a speech recognition engine on the phone can have significant impact on the phone's battery life.
- Offloading parts of the energy cost to the cloud



UNIVERSITY OF CAMBRIDGE

Computation distribution

- Challenges
  - Balance computation energy cost versus network traffic cost
  - Balance local delay versus remote delay
- Traffic
  - Sending raw sensor data may cost more in network energy than what is saved
- Solution: Split computation
  - Perform feature extraction on the phone
  - Perform classification in the cloud
- Adaptive computation distribution
  - Decide best place to do computation dynamically
  - Estimate the cost of off-loading on the fly

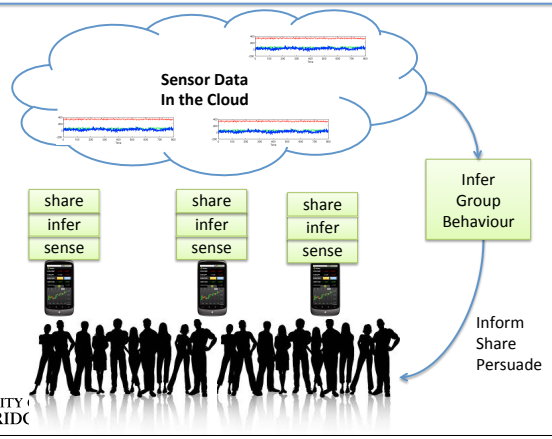
UNIVERSITY OF CAMBRIDGE

Whether to compute locally or remotely ?

- Divide a task into subtasks where each task can be distributed locally or remotely. So we have 2^n possible configurations
  - Voice recognition: FFT, feature extraction, classification
- We need to calculate the impact of each configuration regarding energy, latency, data transmitted over 3G.
- First calculate utility function for each dimension: $u_{e_i} = \frac{e_{\min} - e_i}{e_i}$
  (u_{ei}: utility for energy, e_{min}: energy for best case configuration, e_i energy for this configuration)
- Then calculate utility for each configuration: $u_{c_i} = w_e u_{e_i} + w_l u_{l_i} + w_d u_{d_i}$
  Where u_{ci} is the utility of configuration c_i, and w_e, w_l, w_d are weights for energy, latency and data transmission requirements.
- Select configuration that minimizes the utility function

UNIVERSITY OF CAMBRIDGE

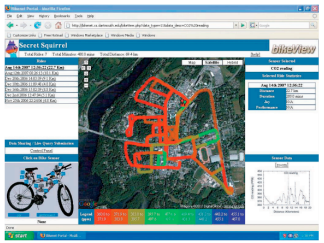
Participatory / Opportunistic Sensing



UNIVERSITY OF CAMBRIDGE

Participatory Sensing Applications

BikeNet



mappiness

UNIVERSITY OF CAMBRIDGE

Reading

- N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A. Campbell. "A survey of mobile phone sensing", IEEE Computer Magazine, vol 48, no 9, September 2010.
- E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, A. Campbell. "Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application", Sensys 2008
- K.K. Rachuri, M. Musolesi, C. Mascolo, P.J. Rentfrow, C. Longworth, A. Aucinas. "EmotionSense: A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research", Ubicomp 2010, September 2010.

- *Further reading, check the publications in the applications section*

UNIVERSITY OF CAMBRIDGE

Summary

- We have seen how mobile phones are being used as a new sensing platform
- We discussed the general design pattern that is used for designing mobile phone sensing applications
- We identified as the major challenge the balance between energy consumption and accuracy, and saw some techniques that are applied to reduce energy consumption.

UNIVERSITY OF CAMBRIDGE