모바일 시스템 프로그래밍

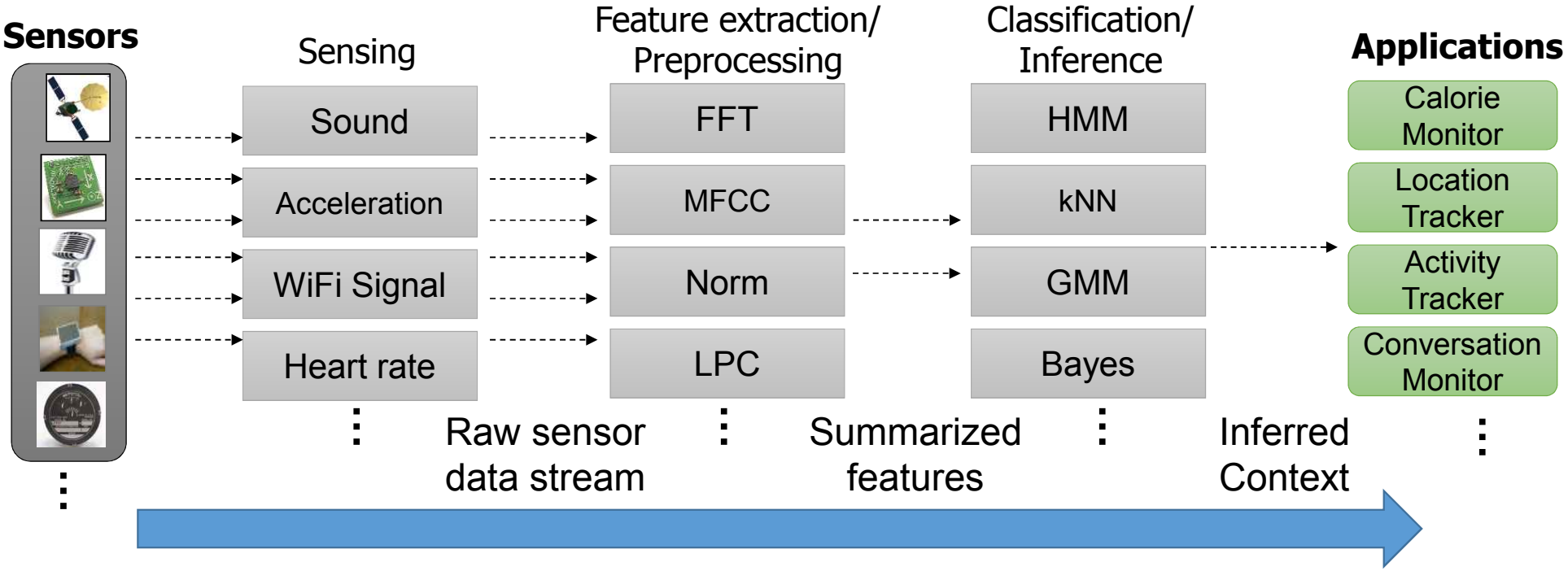
10 Energy-efficient Mobile Sensing System Design 1

2017 1학기

강승우

A Key Challenge of Mobile Sensing Systems

- **Continuous execution** of sensing and processing operations



→ Continuous **resource consumption** (CPU, Sensors, GPS, Mic. …)

→ Huge impact on **Battery**

Energy-efficient Mobile Sensing

- 가능한 한 자원 사용을 줄인다
  - Sensor, GPS, WiFi, BT 센싱 시간/횟수
  - CPU 연산량 / CPU on 시간

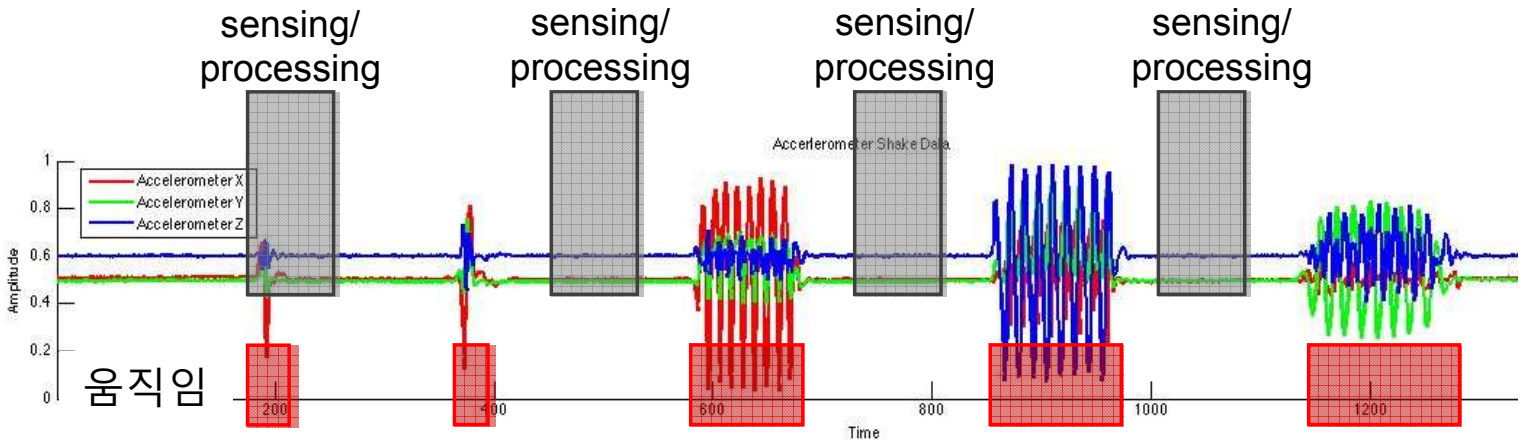
- 그러나 …

Energy-efficient Mobile Sensing

- 가능한 한 자원 사용을 줄인다
  - Sensor, GPS, WiFi, BT 센싱 시간/횟수
  - CPU 연산량 / CPU on 시간

- 그러나 무조건 줄인다고 되는 게 아니다
  - 알아내고자 하는 context 정보(예: 사용자의 위치, 활동, 같이 있는 사람, 등)를 적절한 정확도로 inference 할 수 있는 범위 내에서 줄여야 함

✓ 움직임을 감지하는 로직을 만드는데 자원 사용을 줄이기 위해서 다음과 같이 센싱을 하면 움직임 감지 정확도가 낮아질 것임



Approach to Energy-efficient Mobile Sensing

- H/W 측면의 방법

  → 센싱, 데이터 처리에 소모되는 파워를 줄 일 수 있도록 H/W를 설계

  - 저전력 H/W device (CPU, sensors, …)
  - 센싱 및 데이터 처리 전용 저전력 프로세서 탑재
    - iPhone에 탑재된 M7(M8, M9) motion co-processor
- S/W 측면의 방법

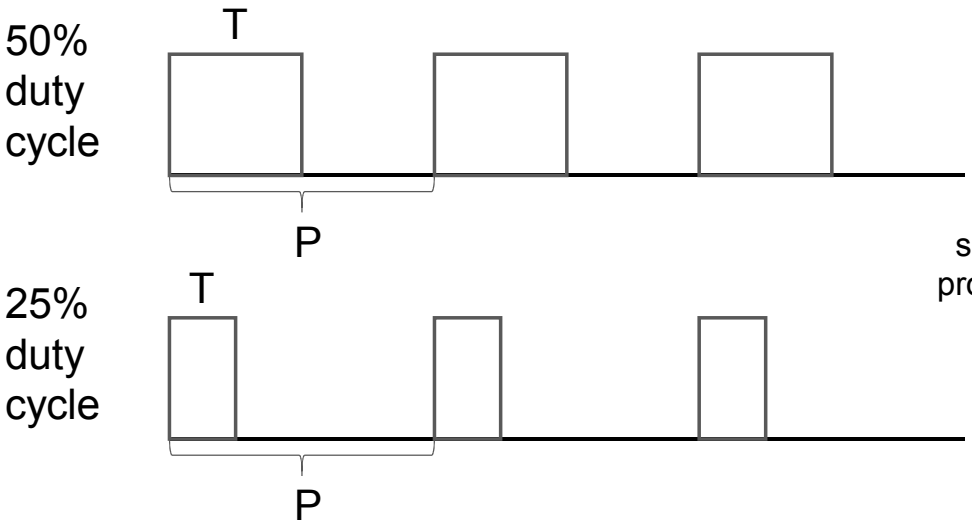
  → 센싱, 데이터 처리 시간/양을 줄여 소모되는 파워를 줄 일 수 있도록 Mobile Sensing Pipeline을 설계

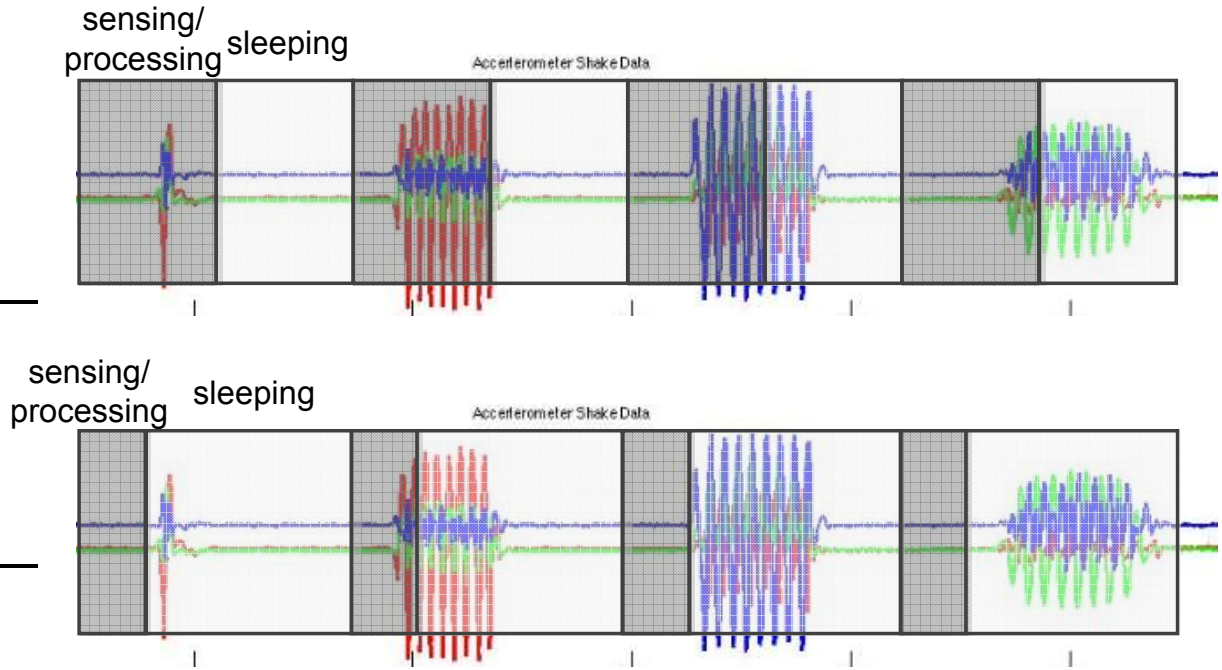
  - Duty Cycling / Adaptive Duty Cycling
  - Hierarchical(Conditional) Sensing
  - Cloud Offloading
  - ……

Duty Cycling

- **주기적으로** 센싱을 하고 데이터 처리를(inference 로직을 수행) 하는 방법

- Duty cycle
  - 신호나 시스템이 일정한 주기를 기준으로 active 상태에 있는 비율
  - D = T/P * 100%
    - T: active 상태에 있는 시간
    - P: 신호의 주기





Duty Cycling

- Inactive(sleeping) time 동안 자원 사용을 줄일 수 있으므로 에너지 소모 감소
- 하지만, 일반적으로 duty cycle이 작아질 수록 정확도는 낮아짐

✓그렇다면 적절한 duty cycle은 어떻게 정할 수 있을까?

- Step Monitor를 만들 때
- Location Tracker를 만들 때

Duty Cycling

- 사람의 context의 특성이나 행동 패턴에 따라서 적절한 수준의 duty cycle은 달리 정해질 수 있음
  - 예를 들어 Step Monitor를 만든다고 할 때,
    10초마다 1초씩 가속도 센서 데이터를 수집하고 처리를 한다고 해도, count 되지 않는 step 수가 많아질 것임 → 걸을 때는 10초에도 10 걸음 이상 걸을 수 있으니까
  - 그러나 Location Tracker를 만든다고 할 때는
    1분에 1번씩만 WiFi scan을 하더라도 큰 오차가 나지 않을 수 있음 → 이동하는 경우를 제외하면, 보통 사람이 어느 장소에 있으면 수 분, 수십 분 동안 그 장소에 있을 확률이 높으니까

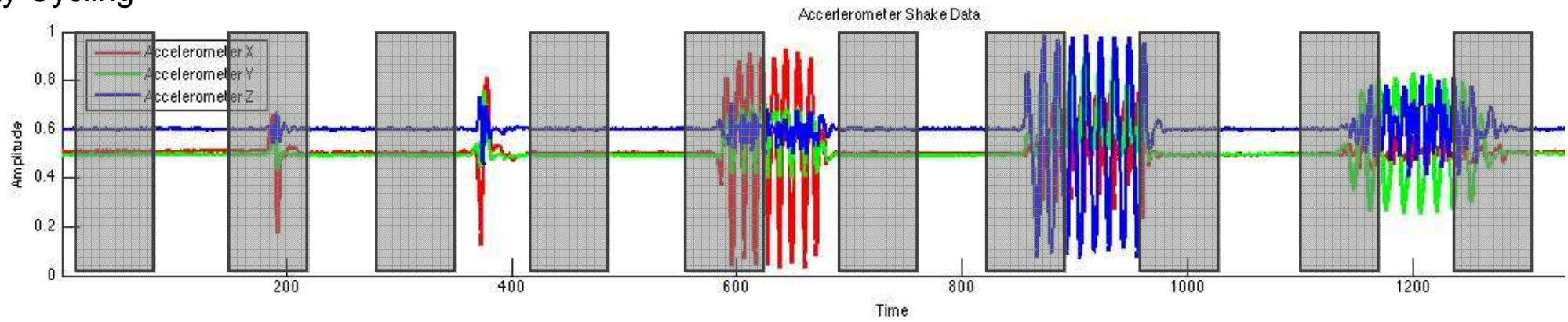
Adaptive Duty Cycling

- 고정된 하나의 duty cycle로만 실행하는 것이 아니라
  감지하고자 하는 이벤트 혹은 inference 결과에 따라서
  duty cycle을 동적으로 변경하여 센싱 및 데이터 처리를 하는 방법
  - 이벤트가 발생하지 않음 → sleeping time을 길게 함
  - 이벤트가 발생함 → sleeping time을 짧게 함

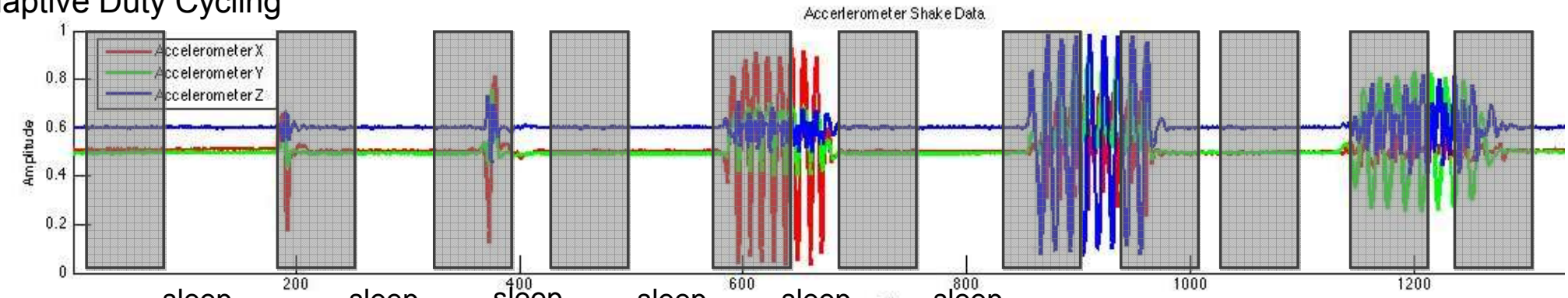
- 이런 방식이 단순한 duty cycling보다 좋을까? 좋다면 왜?

Example of Adaptive Duty Cycling

Duty Cycling



Adaptive Duty Cycling



sleep time 증가 | sleep time 감소 | sleep time 감소 | sleep time 증가 | sleep time 감소 | sleep time 증가

Adaptive Duty Cycling

- Sleeping time 조정은 어떻게?
  - 예
    - Exponential increase – Linear decrease (예: 2t, 4t, 8t / t-a, t-2a, t-3a)
    - Linear increase – Exponential decrease (예: t+a, t+2a, t+3a / (1/2)t, (1/4)t, (1/8)t)

- Continuous sensing에 비하여 에너지 사용을 줄이면서 단순 duty cycling에 비하여 정확도를 높일 수 있다
- 그러나 duty cycle을 조정하는 적절한 방법을 결정하기 위해서는 application에 대한 이해가 필요하다
  - Inference 하려고 하는 사용자의 context가 무엇인가?
  - 어떤 센서를 사용하고 그 데이터의 특성이 어떤가?

Adaptive Duty Cycling

- Adaptive duty cycling이 단순 duty cycling보다 좋을 수 있는 것은 사람의 context (행동 패턴)에는 locality가 존재하기 때문
  - Locality란?

Adaptive Duty Cycling

- Adaptive duty cycling이 단순 duty cycling보다 좋을 수 있는 것은 사람의 context (행동 패턴)에는 locality가 존재하기 때문

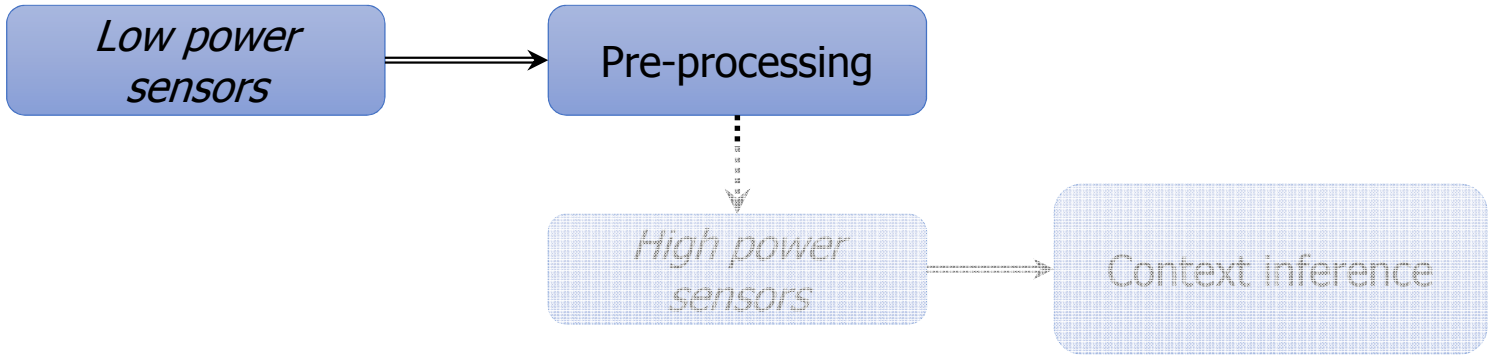
예를 들어, 회사원의 주중 오전 시간대의 활동을 생각해보면

출근(이동) - 30분	사무실 근무(정지) - 3시간	식당 가기(이동) - 10분	식사(정지) - 30분

- Step Monitor라면 이동 중일 때만 자주 sensing/processing을 하고 정지해 있을 때는 하지 않아도 됨
  - 이동, 정지 상황이 일정 시간 동안 지속이 되므로 duty cycle을 adaptation을 함으로써 sensing/processing에 드는 자원 사용을 줄일 수 있음

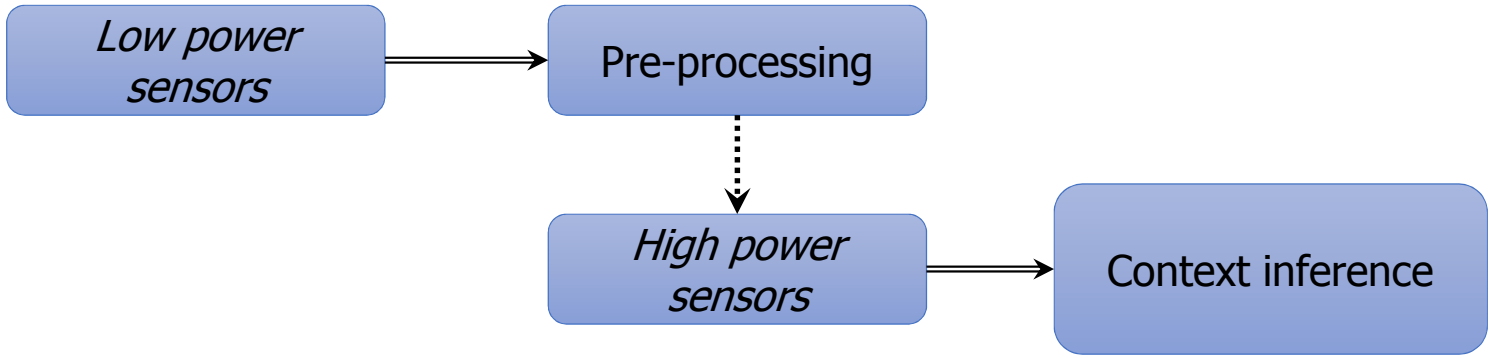
Hierarchical(Conditional) Sensing

- low power sensor (power 소모가 작은 sensor)를 이용하여 특정 조건이 만족되는지를 먼저 확인한 후에
  알고 싶은 context 정보를 inference 하는데 필요한 (보통 power 소모가 큰) sensor를 사용하는 방법



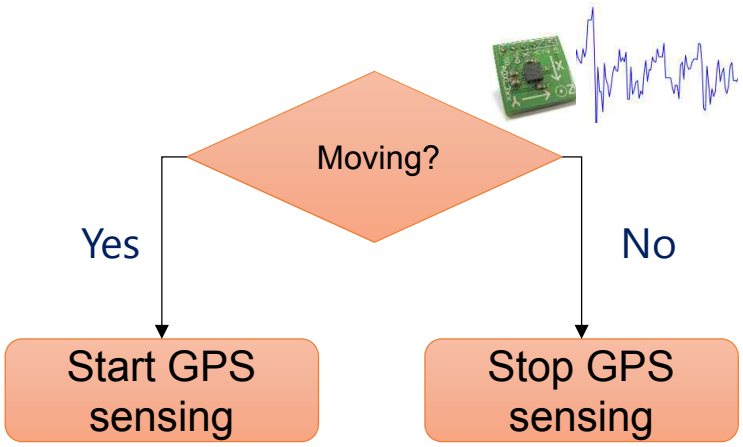
Hierarchical(Conditional) Sensing

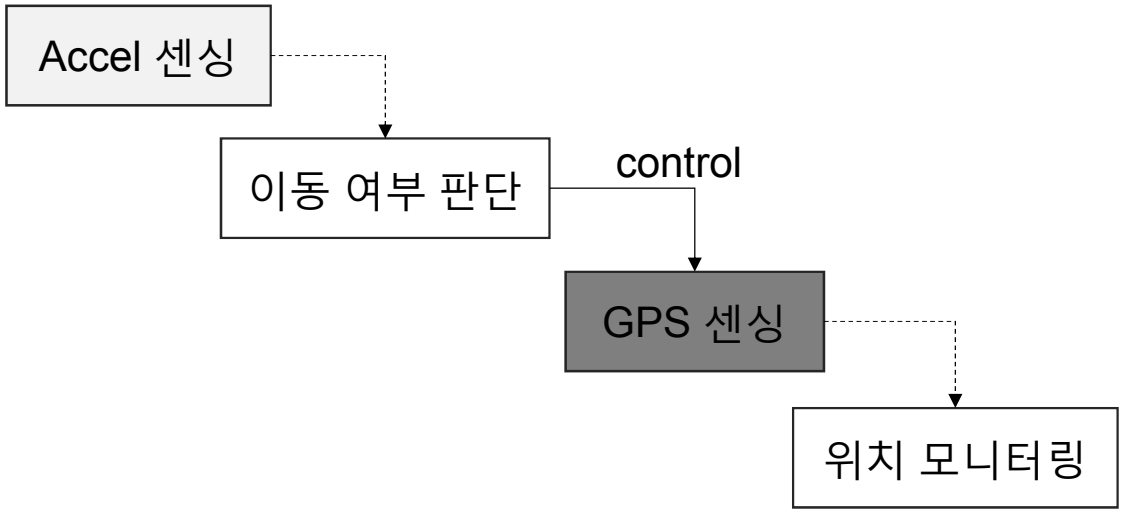
- low power sensor (power 소모가 작은 sensor)를 이용하여 특정 조건이 만족되는지를 먼저 확인한 후에
  알고 싶은 context 정보를 inference 하는데 필요한 (보통 power 소모가 큰) sensor를 사용하는 방법



Hierarchical(Conditional) Sensing

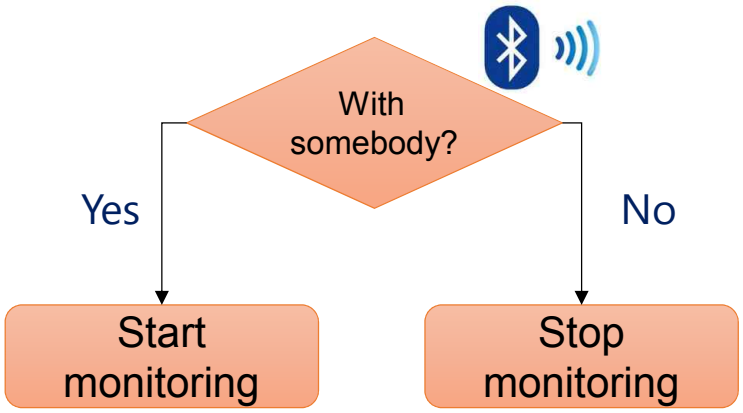
- 위치 모니터링 로직을 구현하는 경우
  - GPS를 계속 켜놓고 위치 데이터를 지속적으로 업데이트 받는 것이 아니라
  - 사용자가 움직이고 있는지 여부를 확인하여 이동 중일 때만 GPS 센싱
    - 이동 중이 아니라는 것은 한 장소에 머무르고 있다는 것이므로 위치 정보 업데이트를 위한 GPS 센싱을 계속 할 필요가 없음

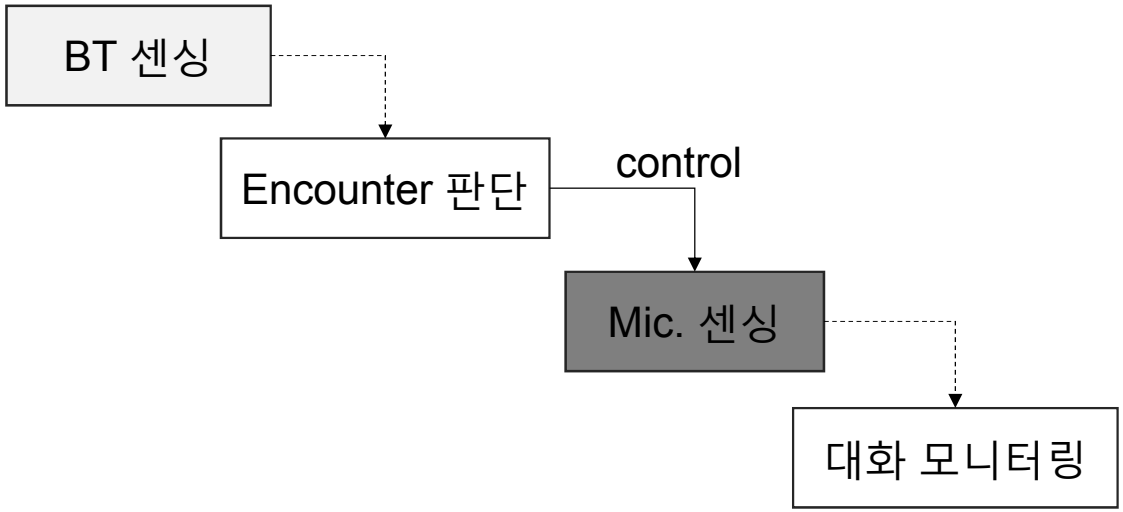




Hierarchical(Conditional) Sensing

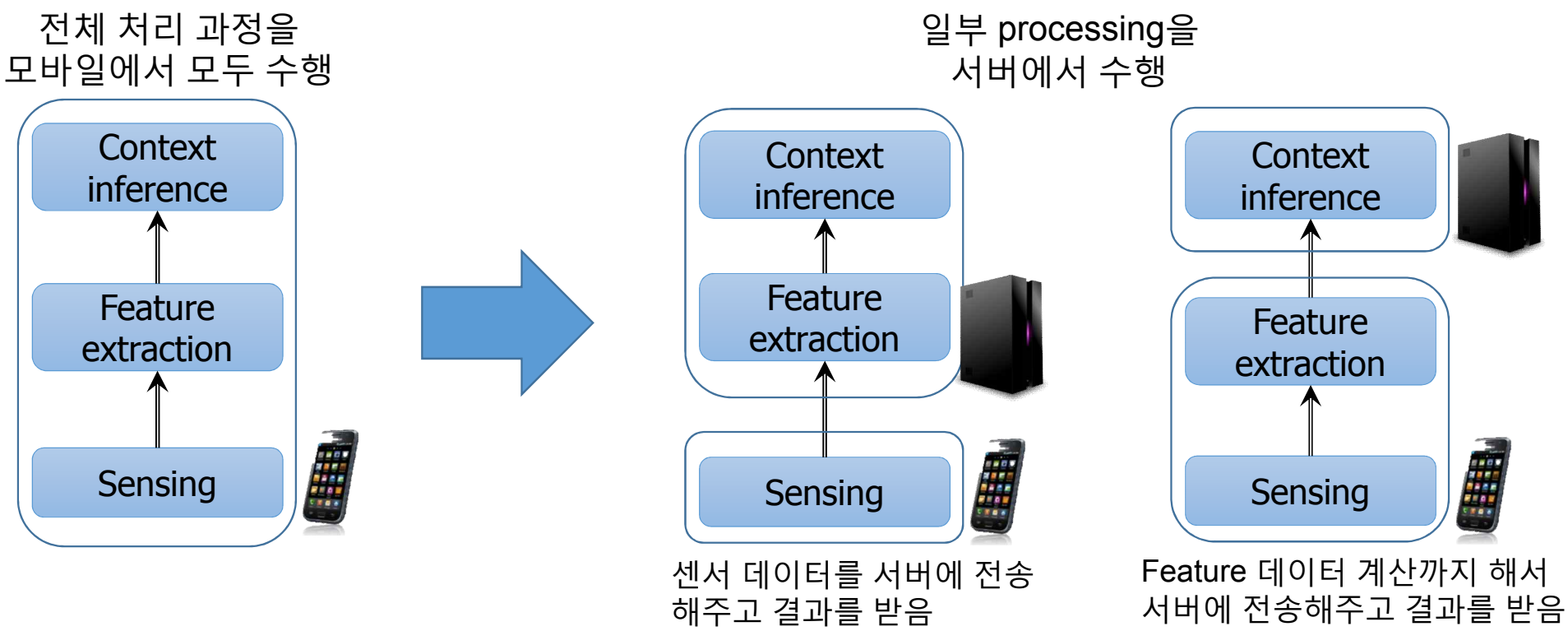
- 가족 간 대화 이벤트를 감지하는 로직을 구현하는 경우
  - 마이크를 이용해서 계속 audio 데이터를 센싱하고 처리하는 것이 아니라
  - 사용자가 등록해 놓은 가족 구성원과 같이 있는지 여부를 먼저 확인하여, 같이 있는 경우에만 audio 데이터를 센싱
    - 같이 있는지 여부는 BT 스캔을 통해 할 수 있음





Cloud Offloading

- Mobile sensing pipeline의 일부를 cloud 서버에서 처리하도록 하여 모바일 디바이스에서의 자원 사용을 줄이는 방법



Cloud Offloading

- Offloading을 하면 항상 에너지가 절약될까?

- 어떤 부분을 언제 offloading을 하는 것이 효율성을 높일 수 있을지 판단을 내릴 수 있어야 함
  - 에너지 측면만 본다면,
    offloading을 하기 위해 발생하는 에너지 소모보다
    offloading을 함으로써 절약할 수 있는 에너지가 커야 의미가 있음

Cloud Offloading

- Offloading energy cost
  - 모바일에서 서버로 데이터를 전송하거나 서버에서 결과를 받으려면 3G/LTE 혹은 WiFi로 통신을 해야 하므로 여기서 에너지를 소모하게 됨
- 절약되는 energy
  - 복잡한 inference 로직 수행을 하지 않음으로써, CPU 연산량을 줄여 소모되는 에너지를 줄임
- 복잡한 연산이 요구되는 음성 인식 같은 작업을 처리할 때 적용 가능