Smart Health – CSE 40816

University of Notre Dame

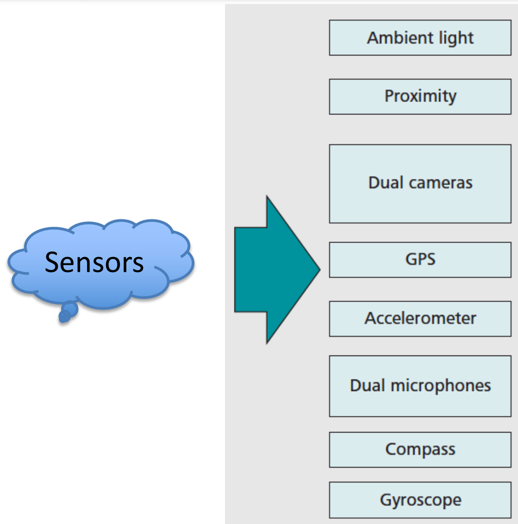
Spring 2020

UNIVERSITY OF NOTRE DAME

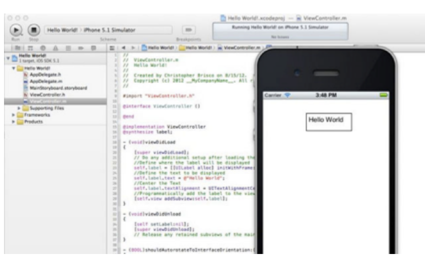
From Smart Devices to Crowdsensing

Crowdsensing refers to the process of collecting sensor data using smart devices from a crowd of contributing users

Technical Enabler 1: Powerful Embedded Sensors in Smartphones



Technical Enabler 2: Open and Programmable



Technical Enabler 3: App Store

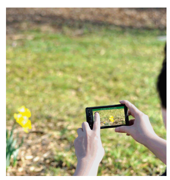
Apple Store

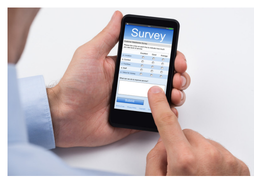
ANDROID APP ON Google play

Technical Enabler 4: Mobile Computing Cloud

Mobile Crowdsensing (MCS) Paradigm

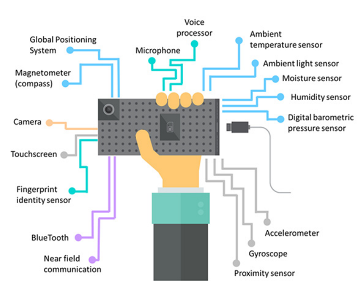
- Participatory Sensing:
  - Users actively engage in the “sensing process”
  - Users determine when/how/what data are collected
  - Human intelligence can be leveraged for complex tasks
  - More costs or incentives are needed to keep humans involved
  - Phone context issues can be resolved





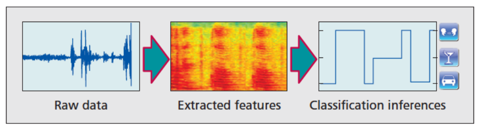
Mobile Crowdsensing (MCS) Paradigm

- Opportunistic Sensing:
  - Fully automated and no user involvement
  - Difficult to determine when/how/what data are collected (phone/user context)
  - Less burden and “costs” on the user
  - Energy issues
  - Activation issues



Mobile Crowdsensing (MCS) Paradigm

- Concerns and issues with both
  - Compliance/incentives
    - Collect data or keep data collection software running
  - Privacy/security
    - What is being collected and shared
  - Data quality (incl. false data)
    - Sufficient amount of data, "good" data, timing of data, times or geographic areas with lower quantity/quality
  - Data labeling/annotations
    - Providing ground truth for collected data
  - Device diversity; resource limitations
  - Turning data into information



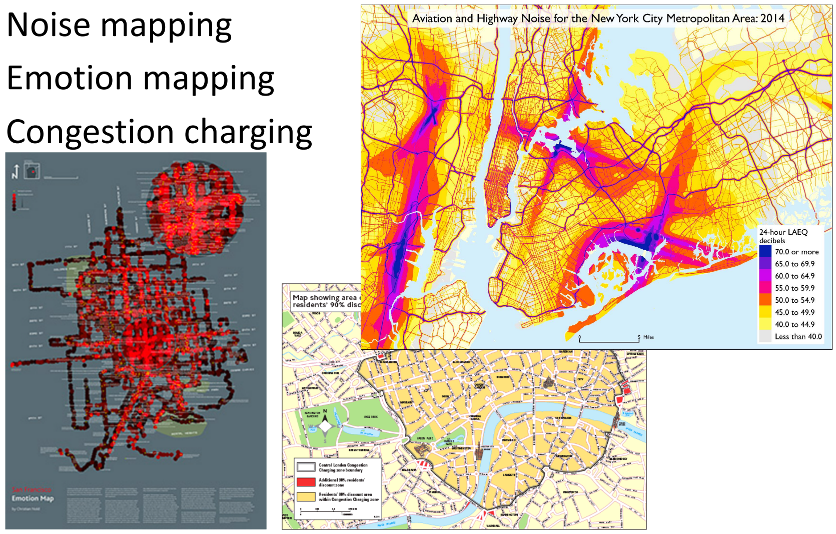
Examples: Participatory/Opportunistic/Both?

- City of South Bend:
  - Collect pollution data in all parts of town
  - Study the routes cyclists take through the city
  - Find graffiti across town
  - Find invasive species in rivers and lakes
  - Find potholes
  - Study activity/mobility patterns of citizens
  - Improve public transportation system

- What type of data? What types of sensors?
- Always-on sensing, context-triggered, human-triggered?
- Challenges: data quality, privacy, compliance, resources?

Examples: Urban Sensing Apps

- Noise mapping
- Emotion mapping
- Congestion charging



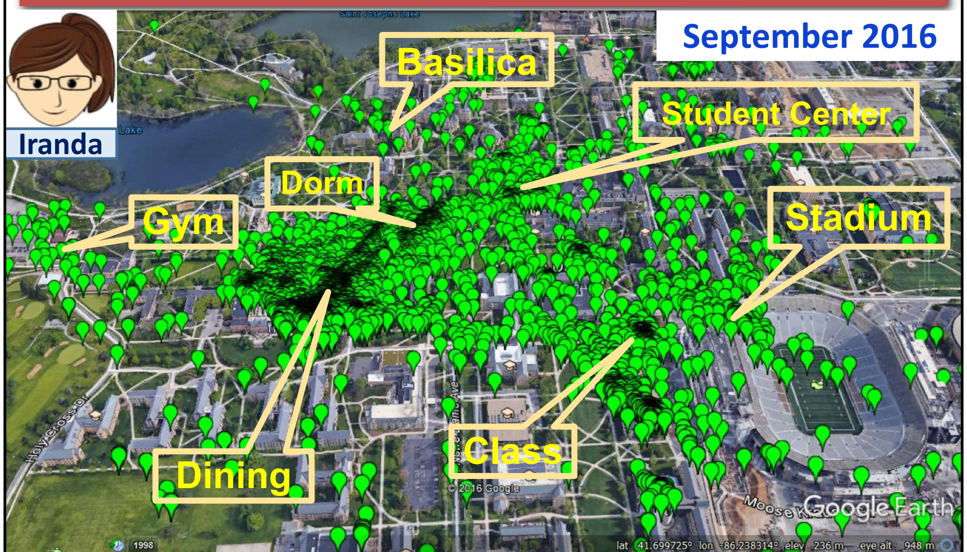
Example: NetHealth Study

- Smartphone Sensor Data

Device	Data Type	Sampling Period (Min.)
iPhone	Location (Latitude, Longitude, Accuracy)	2.75
Fitbit	Step Counts, activity levels (sedentary, light, fair, high), Calorie burn, Heart rate	1

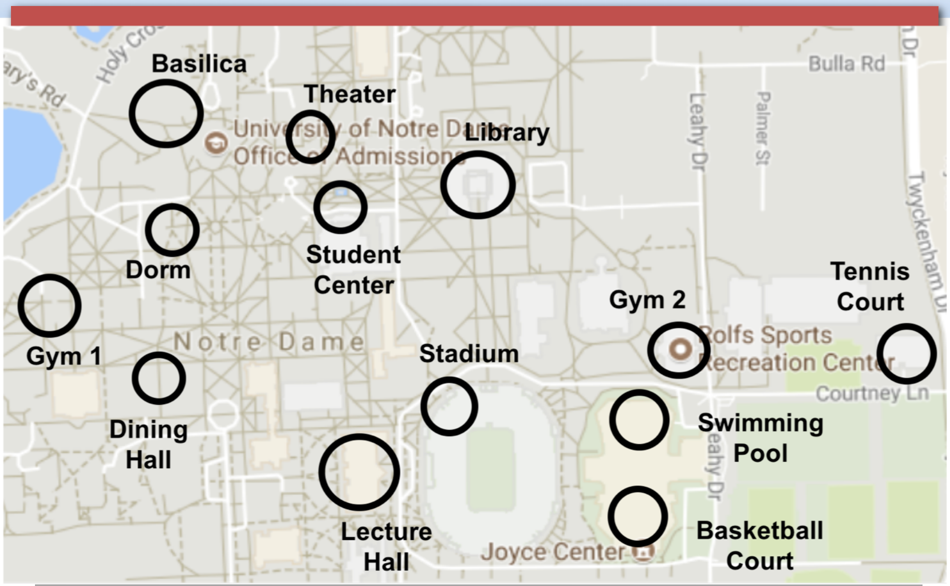
- Subjects
  - 467 iPhone users (on-campus freshmen)
  - Avg. age ~17y 11m (SD = 11m)
  - 2015-2019

Years of Location Data



Computer Science and Engineering - University of Notre Dame

Personal Places of Interest



Computer Science and Engineering - University of Notre Dame

Personal Places of Interest

- GPS represents location as <longitude,latitude>
  - Semantic location is better for reasoning about locations
  - E.g.: street address (140 Park Avenue, Worcester, MA) or (building, floor, room)
  - Geocoding: addresses into longitude/latitude
  - Reverse geocoding: longitude/latitude into addresses
  - Example: Google Geocoding API

Latitude: 37.422005 Longitude: -122.084095

Address:
1600 Amphitheatre Pkwy
Mountain View, CA 94043
Mountain View
94043
United States

Personal Places of Interest

- Place: physical space that has a name (e.g., local businesses, points of interest, geographic locations)
  - Fitzpatrick Hall, South Bend Airport, Studebaker Museum, Martin's Supermarket on Ironwood Rd., etc.
- Location APIs (e.g., Google Places API) often provide contextual information about places near device
  - Name of place, address, geographical location, place ID, phone number, place type, website URL, etc.

GeoFencing

- Sends alerts when user is within a certain radius to a location of interest
- Can be configured to send:
  - ENTER event when user enters circle
  - EXIT event when user exits circle
  - Can also specify a duration or DWELL user must be in circle before triggering event

Points of Interest

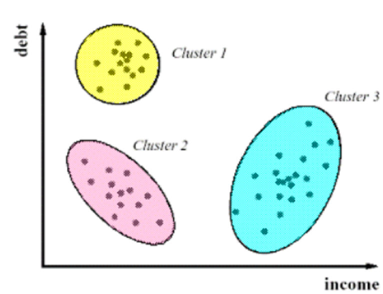
- Points of Interest (PoI): places where a person spends lots of time (e.g., home, work, café, etc.)
  - Given a sequence GPS <longitude, latitude> points, how to infer points of interest?
  - General steps:
    - Pre-process sequence of GPS points (remove outliers, etc.)
    - Form cluster of points
    - Convert cluster to a semantic location

Pre-Processing GPS Points (Remove Noise and Outliers)

- Remove low density points (few neighbors):
  - Places where little time was spent
    - E.g., radius of 20 meters, keep only clusters with at least 50 points
  - If GPS coordinates retrieved every minute, only considering places where you spent at least 20, 30, 50, … minutes
    - What might this time depend on?
  - Remove points with movement:
    - GPS returns speed as well as <longitude, latitude> coordinates
    - If speed value indicates that user is moving, discard that GPS point
  - Reduce data for stationary locations:
    - When user is stationary at same location for very long time, too many points generated (e.g., sitting at at chair)
    - Remove some points to speed up processing

Clustering

- Cluster analysis: group points
- Two main clustering approaches:
  - K-means clustering
  - DBSCAN



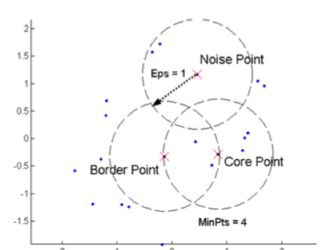
K-Means Clustering

- Each cluster has a center point (centroid)
  - Each point associated to cluster with closest centroid
  - Number of clusters (K) must be specified
  - Algorithm:

Select K points as the initial centroids
repeat
    Form K clusters by assigning all points to the closest centroid
    Recompute the centroid of each cluster
until *The centroids don't change*

DBSCAN Clustering

- Density-based clustering
  - Density: number of points within specified radius (**Eps**)
  - Core points: have > **minPoints** density
  - Border points: have < minPoints density, but are within neighborhood of core point
  - Noise point: not core point or border point



DBSCAN Clustering

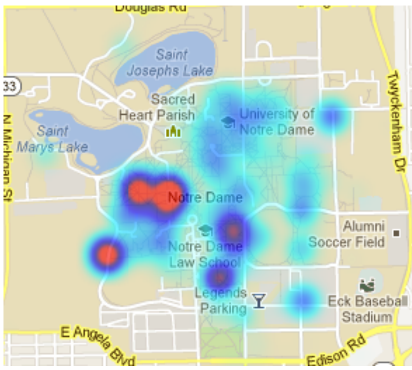
- Eliminate noise points & cluster remaining points

```
current_cluster_label ← 1
for all core points do
  if the core point has no cluster label then
    current_cluster_label ← current_cluster_label + 1
    Label the current core point with cluster label current_cluster_label
  end if
  for all points in the Eps-neighborhood, except i^th the point itself do
    if the point does not have a cluster label then
      Label the point with cluster label current_cluster_label
    end if
  end for
end for
```

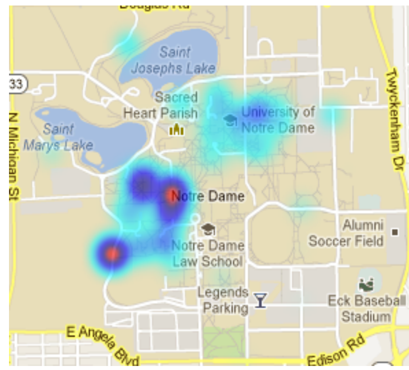

Converting Clusters to Semantic Locations

- Can simply call reverse geocoding or Google Places API on the centroid of the clusters
  - Determining work? Cluster where user spends most time during 9am and 5pm
  - Determining home? Cluster where user spends most time during 6pm and 6am

Sensing Example: Location Hotspots



Subjects' locations during daytime hours



Subjects' locations during nighttime hours

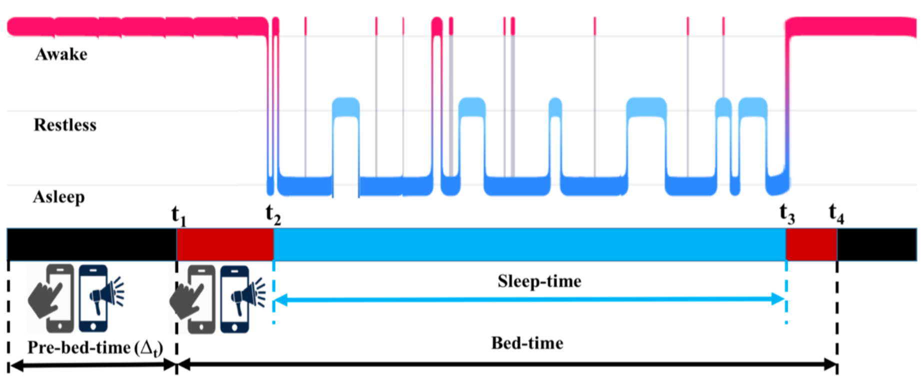
Crowdsensing for Health/Wellness

- Assess a user's **quality of life** through analysis of
  - Place visits and mobility patterns, social interactions, and levels of physical activity
- Researchers and healthcare providers can monitor **patient behavior** remotely
  - E.g., assess the effectiveness of stroke therapy
- Deliver place-specific mobile **health interventions**
  - E.g., encourage individuals to work out when near gyms or parks
- Deliver **customized surveys** to an individual's phone
  - E.g., social interaction surveys, or mood surveys

NetHealth: Continuous Health Monitoring

- Opportunities of continuous monitoring:
  - Identify mobility patterns
    - Time spent indoors/outdoors; type of transportation; locations visited
  - Recognize social interactions
    - Electronic communications (email, phone, SMS, chat)
    - In-person meetings (individual/group, type of meeting, venue)
  - Identify activities
    - Healthy/unhealthy habits, routine household activities, physical activities
  - Other health-related information and events
    - Sleep times/quality, stress, moods, falls and other injuries

NetHealth: Continuous Health Monitoring

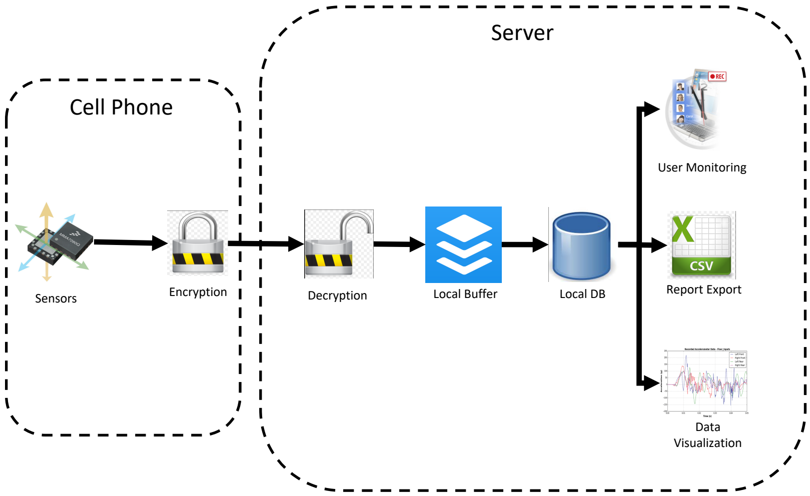


NetHealth: Continuous Health Monitoring

- **C**onfigurable **I**ntegrated **MON**itoring Toolkit
- Powerful smartphone monitoring system for Android-based devices:
  - Monitor **sensor** activities (e.g., GPS, accelerometer, gyroscope, microphone, barometer, magnetometer, etc.)
  - Monitor **user** activities (e.g., communications, apps, music, browsing, etc.)
  - Monitor **system** activities (e.g., resource usage, network connections and traffic, etc.)
- Configurable to meet sensing needs of specific (health) situation

Screenshots of Configuration Screens

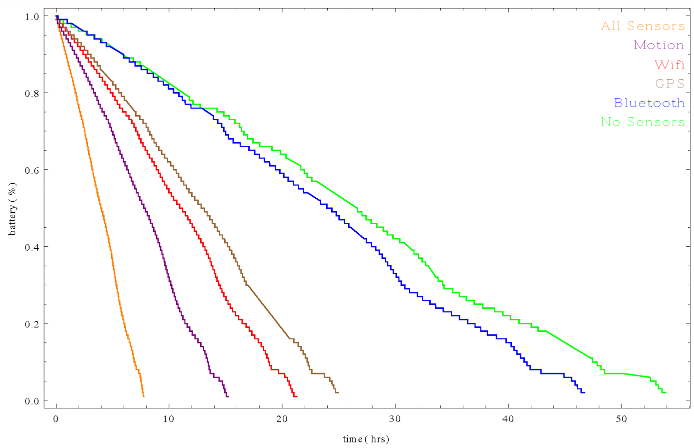
CIMON Framework



Examples of Sensing Capabilities

Sensor Type	Sensing Examples
GPS & Triangulation	Locations, routes, indoor/outdoor time
Accelerometer	Mode of transportation, activities, step counters
Gyroscope	Type of activities, unusual events (falls)
Wi-Fi Proximity	Locations, routes
Bluetooth Proximity	Proximity to friends, family, coworkers, etc.
Magnetometer	Type of activities
NFC/RFID	Locations (supermarket, library, etc.)
Barometer	Locations (floor of building)
Applications	Preferences, moods, interests/hobbies
Phone, EMail, SMS	Communication patterns, moods
Media (Music, Pictures, …)	Preferences, interests, moods

Technical Challenge: Battery Life



Current research focus: *collect maximum amount of data at highest quality possible, while making sure that device will last 14-16 hours* (typical time between recharging)

CIMON Sensing App: Labeling Interface

- Allows subjects to track common types of activities
- Used for development of activity detection algorithms
- In addition to pre-defined activities, subjects can add custom activities