URBAN COMPUTING

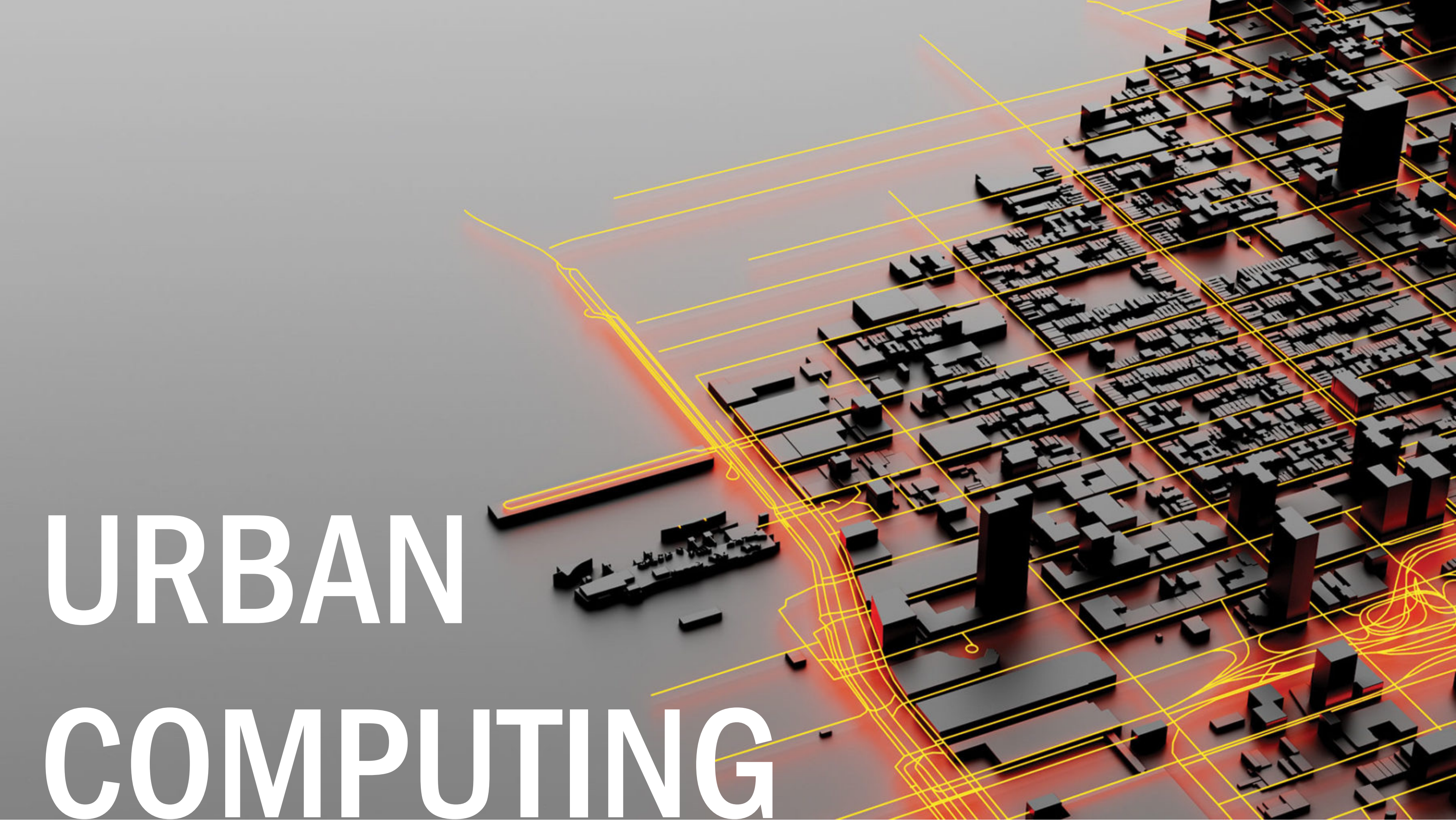


Image credit: The Linux Foundation

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

AI6128 Urban Computing

Course Project 1 Tutorial

A real-world case study: smartphone-based indoor localization

Course Project 1

- Topic
  - Use a publicly available dataset to study indoor localization for smartphone
- Objective
  - Reinforce understanding on various sensors
  - Get familiar with spatiotemporal data
  - Able to pre-process and visualize spatiotemporal data
  - Understand challenges of indoor localization

Overview of this tutorial

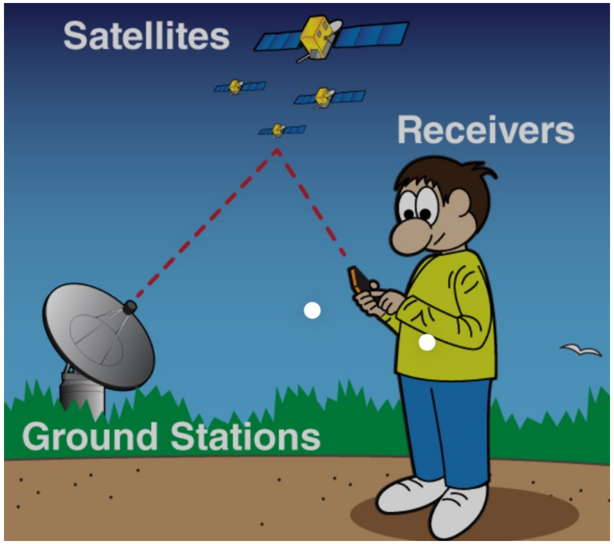
- Background
- Modalities & sensors
- A general workflow
- Tasks & report
- Dataset & sample code

Smartphone & localization

- Why “smart”? – sensors integrated
- Ubiquitous and accessible to everyone.
- Locate your phone → Locate you!
- Why do “localization”?
  - Navigation
  - Emergency
  - Advertisement
  - Entertainment
  - ……

Outdoor vs. Indoor

- Outdoor localization
  - ✓ Global positioning system (GPS)



https://spaceplace.nasa.gov/gps/en/

- Indoor localization
  - ✗ GPS
  - ✓ Smartphone sensors

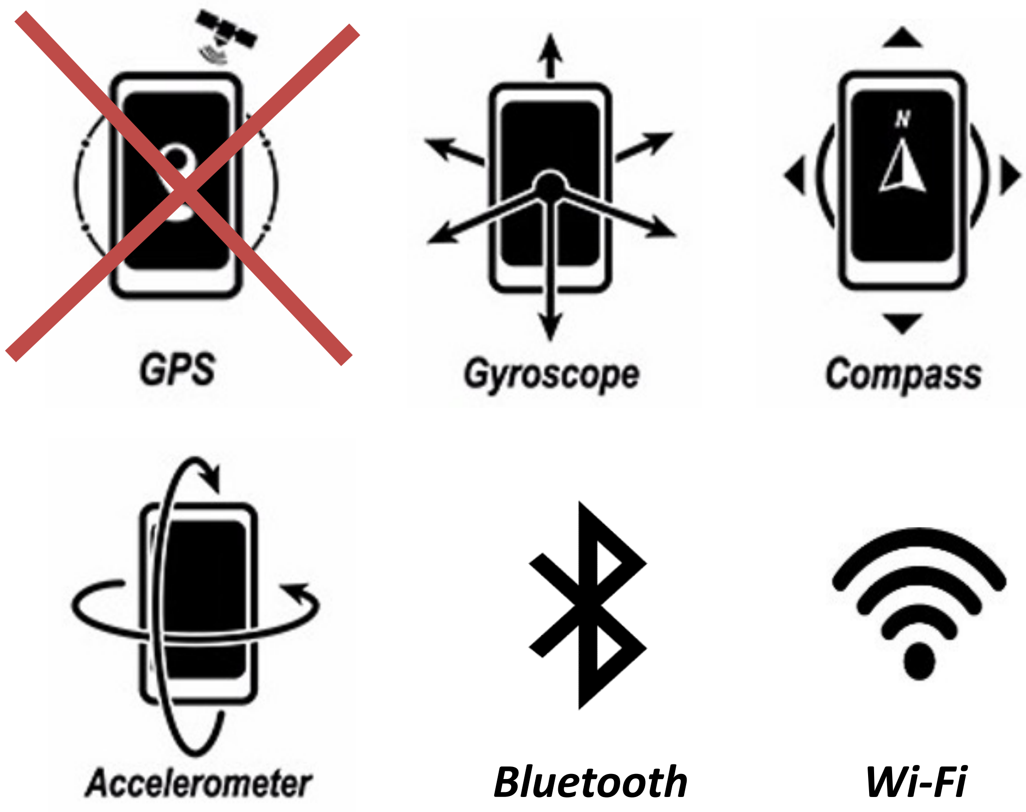
https://www.redpointpositioning.com/blog-gps/

Indoor is challenging

- Complicated environment
- Various scenarios/requirements
  - Accuracy?
  - Scale?
  - Infrastructure deployment?
  - Cost?
  - Privacy?
  - …
- No dominant technology/solution for indoor localization

Survey link: https://arxiv.org/abs/2006.02251

Modalities & Sensors



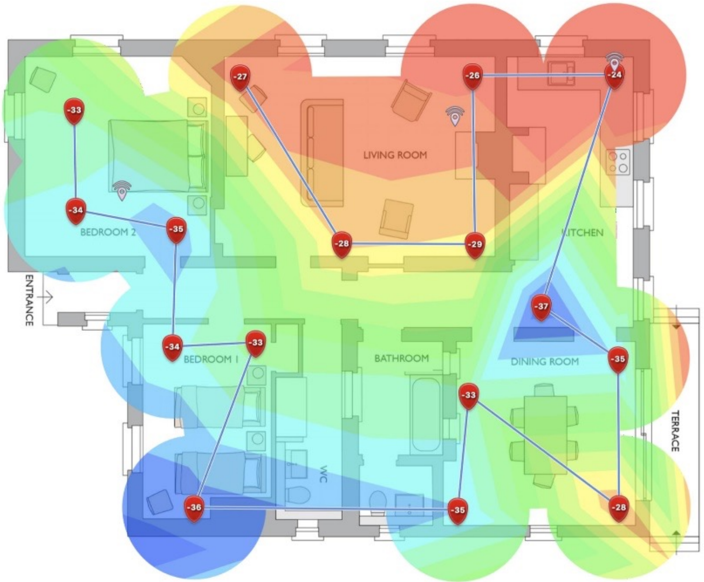
......

https://www.movisens.com/en/solutions/mobile-sensing/

Modalities & Sensors

Wi-Fi

- Received signal strength (RSS)
- Channel state information (CSI)
- [WifiManager API](#) (Android)



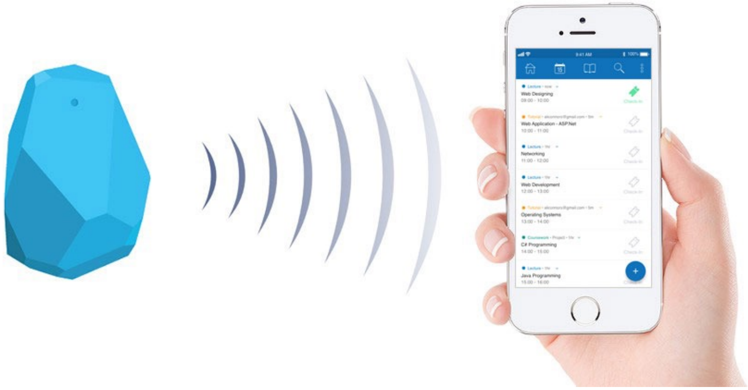
Heatmap

https://osx86project.org/blog/wifi-heatmaps/

Modalities & Sensors

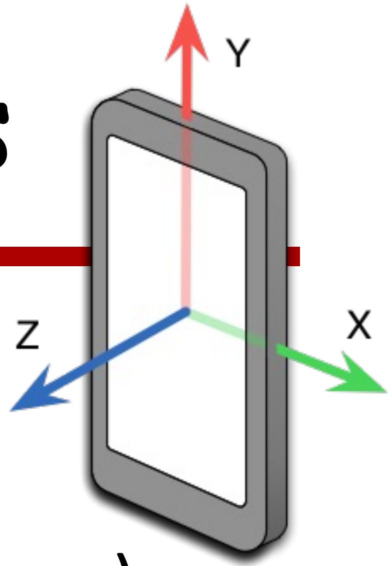
Bluetooth Low Energy (BLE)

- Received signal strength (RSS)
- Approximate location to the transmitter (beacon)
- Bluetooth API (Android)



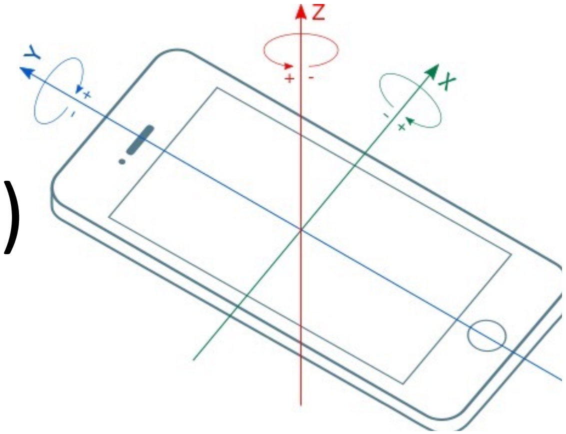
https://www.seatssoftware.com/ibeacon/

Modalities & Sensors



Motion (Inertial)

- Accelerometer
  - Acceleration in three dimension (x, y, z)
- Gyroscope
  - Angular velocity
- Inertial Measurement Unit (IMU)
- Motion sensor API(Android)



https://medium.com/@pivithuruamarasinghe/android-accelerometer-reorientation-c1d3867aa15b

Modalities & Sensors

Geo-magnetic field

- Magnetometer: Measure magnetic field: X, Y, Z
- Motion sensor API (Android)

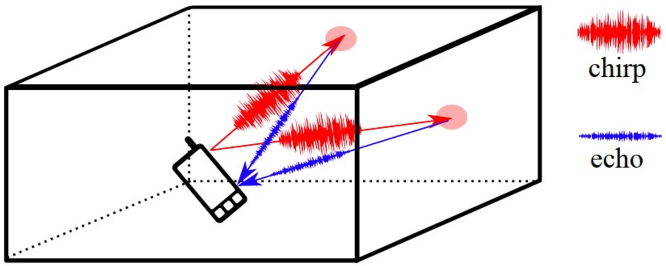
Sensor data from Android

- TYPE_ACCELEROMETER
- TYPE_GYROSCOPE
- TYPE_ROTATION_VECTOR
- TYPE_MAGNETIC_FIELD
- TYPE_MAGNETIC_FIELD_UNCALIBRATED
- TYPE_GYROSCOPE_UNCALIBRATED
- TYPE_ACCELEROMETER_UNCALIBRATED
- TYPE_WIFI
- TYPE_BEACON
- **Details can be found in https://developer.android.com/reference/android/hardware/Sensor**

Modalities & Sensors

Acoustics

- Microphone
- Infrastructure-assisted
- Infrastructure-Free
  - Emit inaudible sound & capture the reverberations



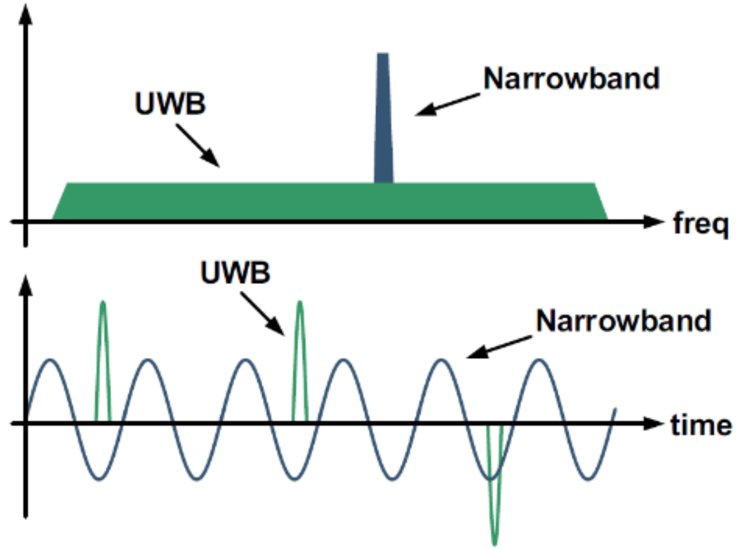
(Ubicomp'18) Deep Room Recognition Using Inaudible Echos.
(SenSys'21 Demo) Infrastructure-Free Smartphone Indoor Localization Using Room Acoustic Responses.
(SenSys'22) Indoor Smartphone SLAM with Learned Echoic Location Features.

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Modalities & Sensors

Ultra-Wideband (UWB):

- Precise timing
- Time-of-Flight (ToF) → Travelled distance



Source: https://www.embedded.com/

Modalities & Sensors

Visible light

- Camera/light sensor

Source: xiaomi

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Localization: modeling

raw data from single/multiple sensors (e.g., $x_{acc}(t, x_{acc}, y_{acc}, z_{acc})$, $x_{wifi}(t, \{ssid_1, RSSI_1\}, \{ssid_2, RSSI_2\}, \dots)$, ...)

ground truth location
(t, x, y)

data preprocessing

training data
$x(x_{acc}, y_{acc}, z_{acc}, \{ssid_2, RSSI_2\}, \dots.), y(x_{gt}, y_{gt})$

train

indoor localization model

Localization: inference

raw data from one sensor or multiple sensors
(e.g., $x_{acc}(t, x_{acc}, y_{acc}, z_{acc}), x_{wifi}(t, \{ssid_1, RSSI_1\}, \{ssid_2, RSSI_2\}, \dots), \dots$)

↓

data preprocessing

↓ input data
(e.g., $x(x_{acc}, y_{acc}, z_{acc}, \{ssid_2, RSSI_2\})$)

indoor localization model

↓ predict

location (x, y, z)

Dataset

- Microsoft Indoor Location Competition 2.0 Dataset (https://github.com/location-competition/indoor-location-competition-20)
- Data collected by smartphones in two multi-story commercial buildings

Site 1 has 5 floors

Site 2 has 9 floors.

Essential tasks (100%)

- Essential tasks (100%)
  - Visualize way points (ground-truth locations)
  - Visualize geomagnetic heat map
  - Visualize Wi-Fi RSS heat maps of 3 Wi-Fi APs
  - Visualize iBeacon RSS heat map
  - Requirements
    - You can choose any programming language
    - You can refer to the sample code in Python, write your own code to **pre-process the data** and use a basic plotting tool (e.g., matplotlib) to **visualize the data**
    - No need to superimpose your visualization onto map
    - 2-person group to cover 2 essential tasks
    - 3-person group to cover 3 essential tasks
    - 4-person group to cover 4 essential tasks

Data preprocessing

Essential tasks

- Visualize way points (ground-truth locations)
- Visualize geomagnetic heat map
- Visualize Wi-Fi RSS heat maps of 3 Wi-Fi Aps
- Visualize iBeacon RSS heat map

Ground-truth location collected by volunteers

Sensor data with timestamp
(Accelerometer, Gyroscope, Magnetic, Wifi RSSI, iBeacon RSSI)

sensor data with position
(ready for training)

Visualization

- Various plotting packages
  - E.g., Matplotlib for Python codes
  - Matlab
  - R
  - …

Bonus tasks

- Build a deep learning-based location fingerprint model
- Study the performance improvement brought by multi-modal machine learning
- Study the performance improvement brought by integrating temporal relationship via SLAM
- Any other you can claim

Project 1 Report

- Format
  - Use IEEE A4-size two-column conference templates https://www.ieee.org/conferences/publishing/templates.html
  - Don't change page margins and font sizes
- Submit the writeup in PDF format
  - Submission deadline: by the end of **Week 9 (Oct 20th 2024)**
  - Via NTULearn under Content folder
- One-week grace period for late submissions
  - No penalty if a valid excuse provided; otherwise, a penalty of 20% reduction will be applied to the mark of the late submission
  - Zero mark for submissions after the grace period
- Policy on plagiarism
  - Write by yourselves based on your own understanding
  - We will use a tool to check submissions against databases
  - Obvious plagiarism cases will have zero scores

Suggested Project 1 Report Content

- Section 1: Introduction (0.5 page)
- Section 2: Dataset (0.5 page)
- Section 3: Essential tasks (1 page each)
  - Subsection 3.1: Visualization of waypoints
  - …
- Section 4 (optional): Bonus tasks (1 page each)
- Section 5: Group member contributions (within 1 page)
- Appendix: source code

Introduction:

- Essential parts to be covered:
  - What topic is this report about?
  - What are the challenges/problems to be solved?
  - A brief introduction of used approaches
  - A short presentation of the results.
- Things to be noted:
  - An overview of the whole report.
  - The text shall be super concise and contain no technical details.
  - Can be understood by a non-technical reader.

For each task

- Approach description:
  - Contain enough details so that others can reproduce.
- Result presentation:
  - Each figure/result shall be:
    - a) described *(what do the points/lines mean?);*
    - b) explained *(why does it look like this? Possible reasons?)*

Project 1 Assessment

- Purely based on report
- Overall achievement and quality (70%)
  - Coverage of essential tasks
  - Pre-processing result quality
  - Depth of discussion on the results (e.g., what challenges experienced, **how they are addressed or why they cannot be addressed, etc.**)
- Individual contribution (30%)

Dataset & Sample code

- Raw data explanation
  - https://github.com/location-competition/indoor-location-competition-20
- Prerequisites:
  - Python 3
  - Required python packages: numpy, scipy, dataclasses, plotly, pillow.
- Run sample code on your own computer:

  pip3 install wheel
  pip3 install numpy scipy dataclasses plotly pillow
- Run sample code on Google Colab (cloud):

  https://colab.research.google.com/drive/1z3EhIBNwuZbqffw36hizQ1I4iiVOI6d2