Fundamental Concepts for Guidance and Navigation

AAE4203 – Guidance and Navigation

Dr Weisong Wen
Assistant Professor

Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University

Week 1

Dr Weisong Wen (Trustworthy Autonomous Systems)

Sept 2023-Now, Assistant Professor
2021-Aug 2023, Research Assistant Professor
2020-2021, Postdoc at AAE
2017-2020, PhD from Mechanical Engineering

2018, Visiting Researcher in UC Berkeley

2016-2017, Co-found Researcher in iDriverplus (estimated value RMB 5 billion)

2015, 2017, BSc, MSc of Mechanical Engineering from BISTU and CAU

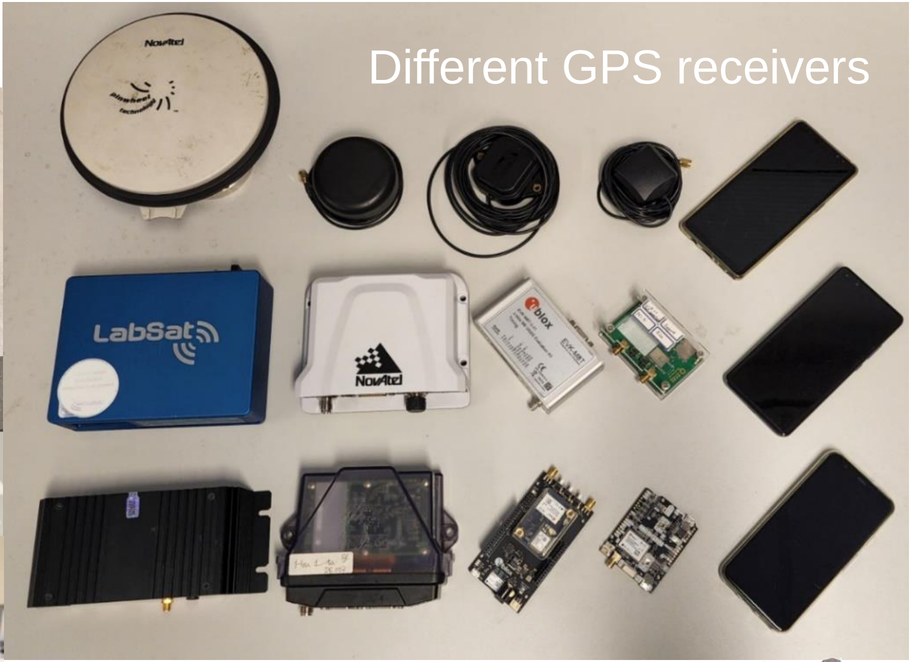
Selected academic contributions

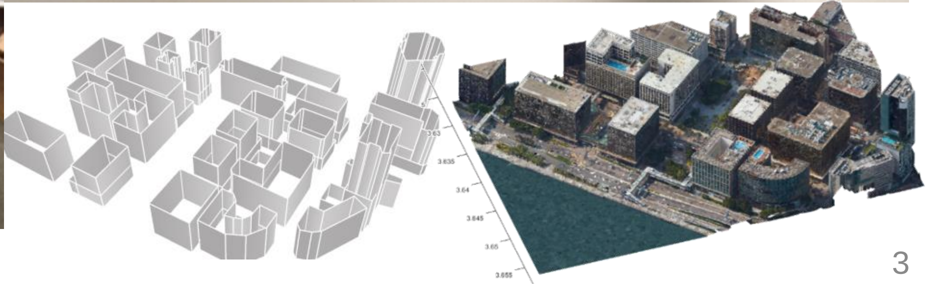
- **33 SCI journal (17 Q1, 11 Q2) and 40 conference papers** (highly cited papers, Best Presentation Award etc.)
- Session chair of ION GNSS+ 2022/2023, ICGNC 2022 and IEEE ITSC 2022/2023
- Over 19 millions HKD of research fund as PI

Funder (as PI since 2021)

Selected research collaborators

Research Facilities





Our Teaching Assistant

Ruijie XU

Xikun LIU

Yihan ZHONG

Yuan LI

TA: PhD students from AAE

Business Insurance

- Let’s get to know with each other
  - Short introduction about yourself (if we have enough time)? ☺
  - Who is your Final Year Project supervisor and what is your topic? ☺
  - Why you select this course? ☺

Ground Rules

For students:

Open mind; speak English; participate activities assigned; ask questions

For teachers:

Arrive on time; reply emails on time; answer questions related to the subject

Be curious, Be inspired, Be motivated, Study further by yourself.

Intended Learning Outcomes

- Understand and explain the working principles of navigation and guidance systems for unmanned autonomous systems (UAS); and
- Competently apply the fundamental mathematical concepts of UAS navigation; and
- Critically evaluate the characteristics, purposes, and design procedures of UAS navigation and guidance systems; and
- Identify the technological and design trends of future UAS navigation.

Assessment and Basic Requirement

- Assessment:
  - Homework Assignment (strictly no late submission) (15%)
  - Mid-Term Quiz/Test (15%)
  - Group Project (Case study, 2-3 members in a group) (20%)
  - Final Exam (50%) (Open book)
- Basic requirement:
  - Mathematics on matrix and its calculation
  - Basic coding skills with Python (expect to learn yourself)
  - Extra time for finish the coding homework based on Python
  - Assurance on the attendance

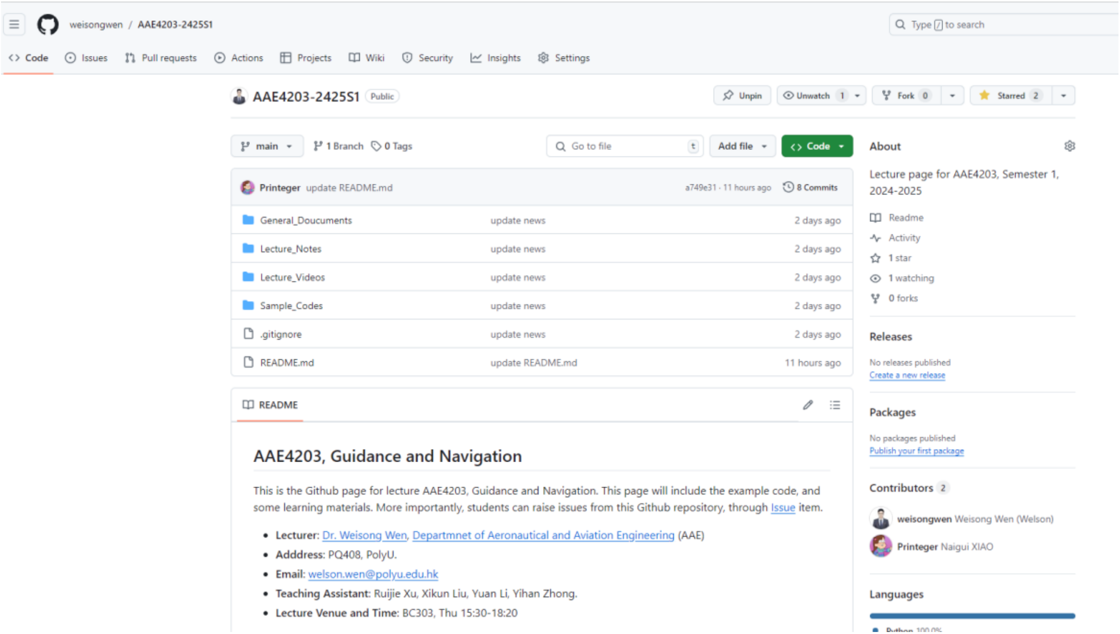
Teaching Plan (Tentative)

Sem. Week	Topics Taught	Assessment
Week1	Fundamental concepts	Assignment 1
Week 2	Satellite Navigation-I (GNSS SPP, linear least squares)	
Week 3	Satellite Navigation-II (GNSS Doppler, linear least squares for velocity estimation)	
Week 4	**No Lectures**	
Week 5	Satellite Navigation-III (GNSS RTK, DGNSS, linear least squares for RTK estimation)	
Week 6	Tutorial on Coding with Python for GNSS positioning	
Week 7	Visual positioning I: Visual feature detection and tracking	
Week 8	Mid-term	Assignment 2
Week 9	Visual positioning II: visual pose estimation	
Week 10	Kalman filtering for GNSS positioning and sensor integration: part 1	
Week 11	Kalman filtering for GNSS positioning and sensor integration: part 1I	
Week 12	Case Study Presentation	Group Presentation & Report
Week 13	Case Study Presentation	Group Presentation & Report

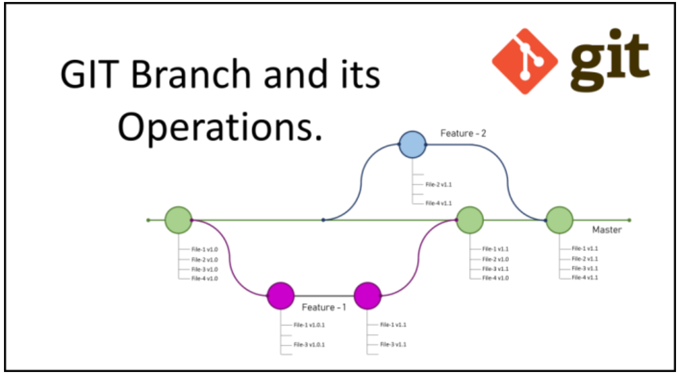
Department of Aeronautical and Aviation Engineering 航空及民航工程學系 The Hong Kong Polytechnic University 香港理工大學

GitHub Repository

https://github.com/weisongwen/AAE4203-2425S1



GitHub:
GitHub is a developer platform that allows developers to create, store, manage and share their code, and wikis for every project.

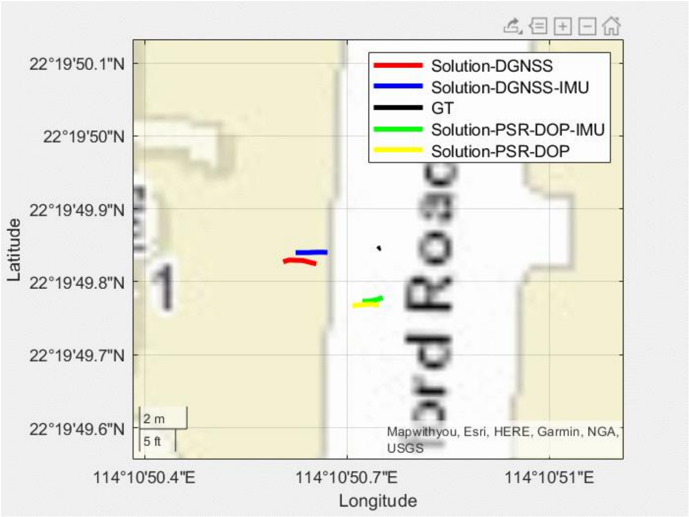


Outline

- Guidelines on the use of Generative AI in Learning
- Applications of navigation technology:
  - What is the role of the navigation in UAS?
  - Typical sensors for navigation
  - Pros and Cons of different navigation sensors/technology
  - Q&A
- Coordinates and transformation:
  - Why do we need coordinates?
  - ENU, ECEF, body frame
  - Transformation between coordinates
  - Q&A

Definition of Navigation

Definition: "any of several methods of determining or planning a ship's or aircraft's position and course by geometry, astronomy, radio signals, etc."



Examples of navigation devices that you are using daily.

- Smartphone
- Smart watch
- Vehicle navigator
- …

* Example of smartphone positioning in Hong Kong (~5 meters of accuracy)

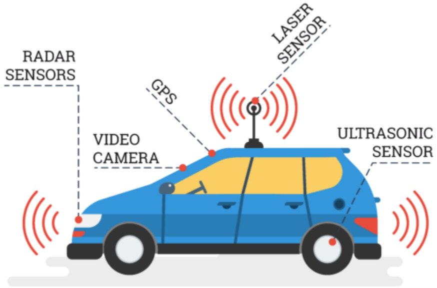
Navigation in Unmanned Autonomous System (UAS)

Tesla Autonomous Driving Car

Integration of cameras, maps, vehicle sensors and GNSS for robust and accurate navigation.

Our PolyU Autonomous Driving Car

Integration of LiDAR, maps, vehicle sensors and GNSS for robust and accurate navigation.



UAS rely on onboard sensor to determine the absolute position.

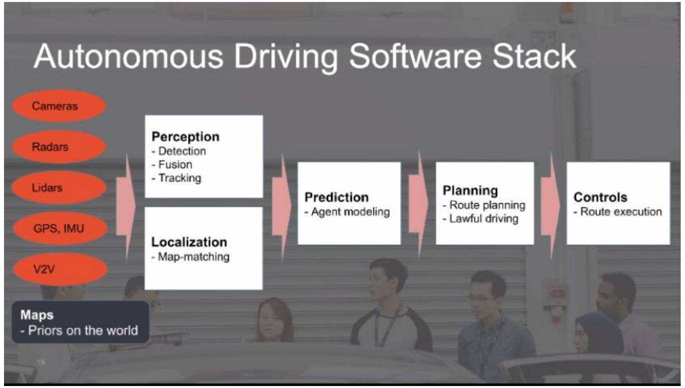
Department of Aeronautical and Aviation Engineering
航空及民航工程學系
THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Navigation in Unmanned Autonomous System (UAS)

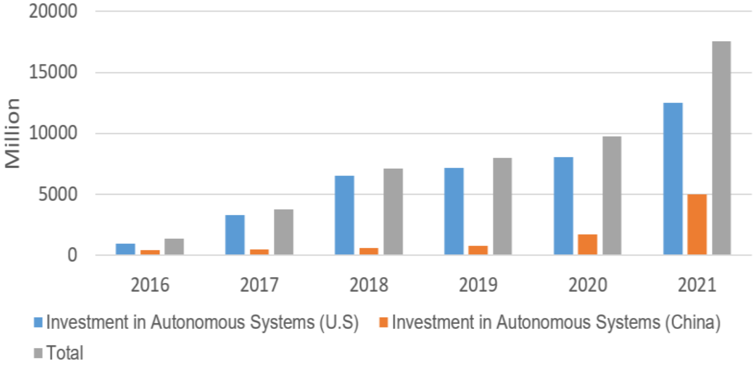
Scene understanding and planning
(what is the optimal path to there?)

Perception (Localization and mapping)
(where are you?)

System control by wire
(how can you to go there?)



Lots of industry investment in China and U.S

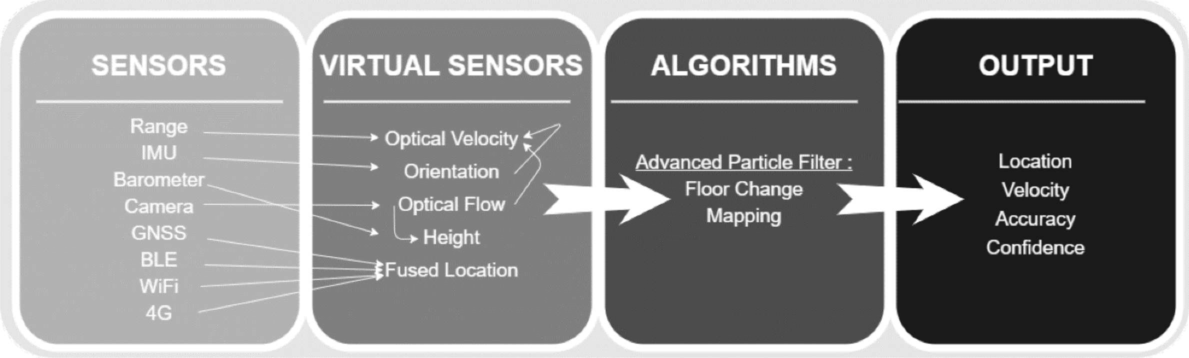


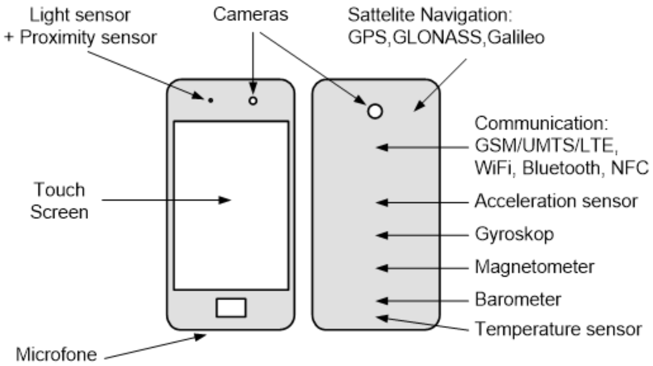
Source*: CMU

Navigation in Smartphone

Definition: “any of several methods of determining or planning a ship’s or aircraft’s position and course by geometry, astronomy, radio signals, etc.”

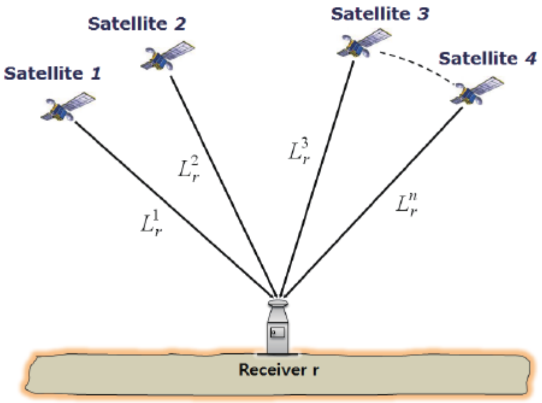
What sensors do we have in a smartphone for navigation?

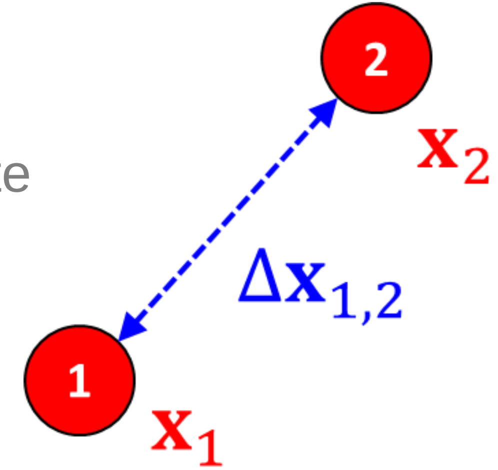




Sensors for Navigation

- Radio navigation (**point-positioning**)
  - Ultra wideband (UWB)
  - Wi-Fi
  - Cell communication 3G/4G/5G
  - Satellite navigation
  - etc
- Robotics navigation* **(dead-reckoning)**
  - Inertial sensors, accelerometers/gyroscope/magnetometer
  - Visual sensors
  - LiDAR sensors
  - etc





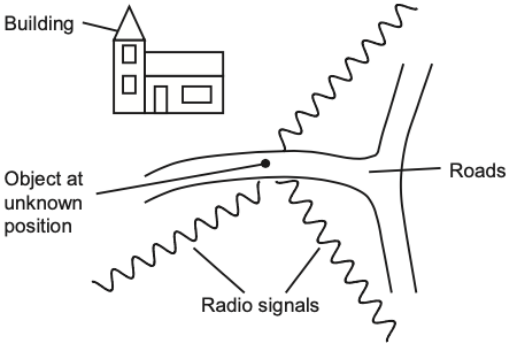
Framework of Sensors to Navigation

	Sensing	**Measuring**	**Measurement**	**Model**	**Estimate**
GNSS	GNSS Signal	Signal Defined Receiver	Pseudorange	Least Square Estimation	Point Position
INS	Raw measurements (voltage, delays, etc)	Accelerometer/ Gyroscope Models	Acceleration & Angular Velocity	INS Mechanization	Relative Position
Visual Navigation	Images	Features Detection & Tracking	Associated Features	Least Square Estimation	Relative Position

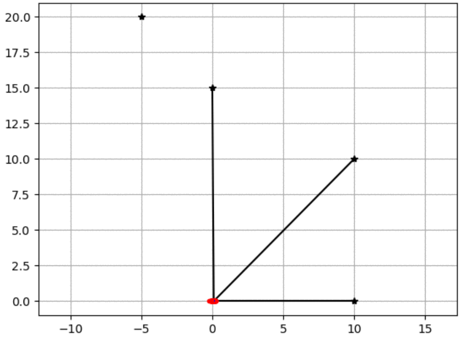
Point Positioning Methods

- Position may be inferred directly by ***matching the signals receivable and/or features*** observable at a given location with a database. (such as Wi-Fi fingerprinting)
- Alternatively, more ***distant landmarks*** at known positions may be selected and their distance and/or direction from the user measured. A landmark may be a transmitter (or receiver) of signals or an environmental feature. A landmark installed specifically for navigation is known as an aid to navigation. (such as GPS positioning)

Wi-Fi based positioning

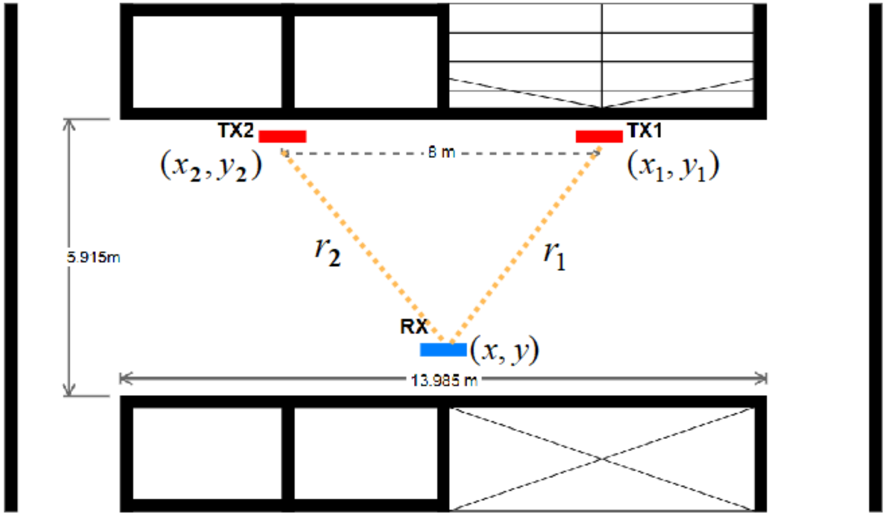


Radio based positioning



Landmark based positioning

Example: TDOA (Time Difference of Arrival) Model



$$(x_1 - x)^2 + (y_1 - y)^2 = r_1^2$$
$$(x_2 - x)^2 + (y_2 - y)^2 = r_2^2$$

$$x_i^2 + y_i^2 + x^2 + y^2 - 2x_i x - 2y_i y = r_i^2$$
$$K_i = x_i^2 + y_i^2, \quad R = x^2 + y^2$$
$$r_i^2 - K_i = -2x_i x - 2y_i y + R$$
$$\begin{bmatrix} r_1^2 - K_1 \\ r_2^2 - K_2 \end{bmatrix} = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ R \end{bmatrix}$$

$$\mathbf{Ax} = \mathbf{b}$$

Time Difference of Arrival (TDOA): also known as Time of Arrival (ToA) difference, is a method used for geolocating RF signals. The principle behind TDOA is based on measuring the difference in time at which a signal arrives at multiple receivers located at different points.

How many signals are required to calculate the position of the receiver?

Wi-Fi Round-trip Timing (RTT) Positioning

Wi-Fi Round Trip Time (Wi-Fi RTT) allows computing devices to measure the distance to nearby Wi-Fi access points (APs) and determine their indoor location with a precision of 1–2 metres using round-trip delay.

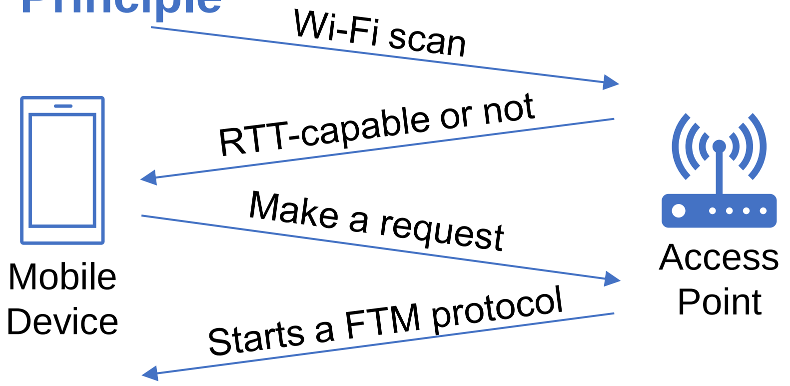
With a single Wi-Fi access point (AP), only a distance measurement is available. With three or more nearby APs, an app can trilaterate a device's location with an accuracy of one to two meters.

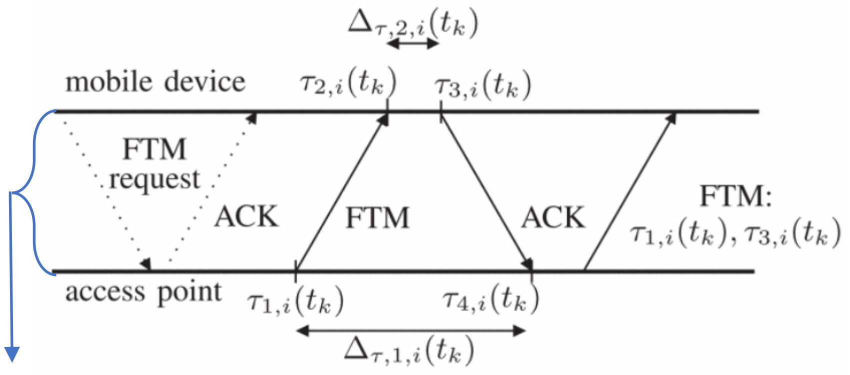
- The position of the Wi-Fi AP are known variables.
- The distance between the Wi-Fi AP and the user (e.g. smartphone) are known variables (measurements).
- The location of the user are unknow variables to be calculated.
- At least 3-4 AP are required to calculate the location of the user.

Wi-Fi Round-trip Timing (RTT) Positioning

Ranging Principle

FTM:
ACK:





Repeats from 1 to 32 times, typically 8

$$\tilde{d}_i = \frac{c}{2n} \cdot \sum_{k=1}^{n} \left(\tau_4(t_k) - \tau_1(t_k)\right) - \left(\tau_3(t_k) - \tau_2(t_k)\right)$$

$$\hat{d}_i = d_i + b_i + n_i$$

d_i: the actual distance between i^{th} Wi-Fi Access Point (AP) and a smartphone

b_i: measurement error between i^{th} AP and a smartphone due to MAC address transfer delay or clock error.

n_i: zero mean Gaussian noise with a standard deviation σ_i

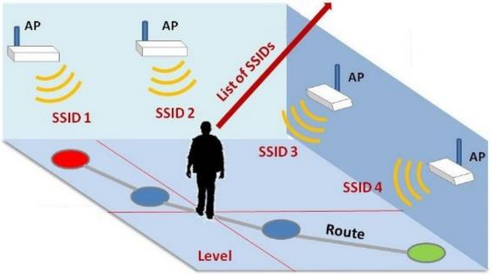
Privacy

- No need to connect to AP during ranging;
- Only requesting device can determine the distance to APs, APs do not have the information;

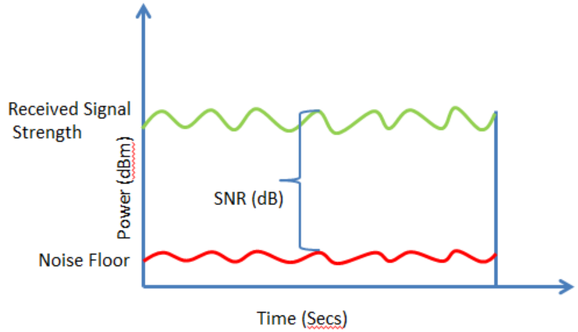
Wi-Fi Round-trip Timing (RTT) Positioning

Limitation of the Wi-Fi Round-trip Timing (RTT) Positioning:

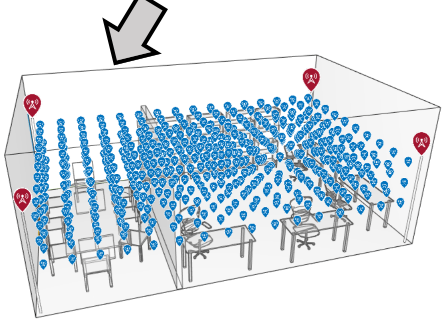
- Limited Range: The effective range of Wi-Fi RTT is limited by the range of Wi-Fi signals, which typically do not extend beyond a few hundred meters.
- Line-of-Sight Requirement: RTT positioning works best when there is a clear line of sight between the device and the access points. Obstacles like walls, furniture, and other objects can cause signal reflections and multipath effects, which can degrade accuracy.

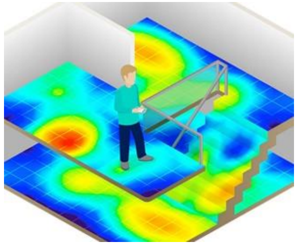


Received Signal Strength Indicator (RSSI)



Received signal strength (RSS)





Fingerprinting

a process of signal collection and association with indoor locations

The equation for measuring path loss:

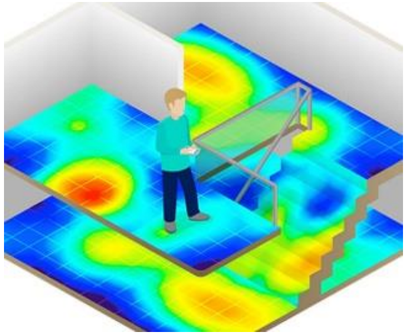
$$P_t - P_{rssi} = 20log10(f) + Nlog10(d) + L_f\ (n) - 28dB$$

- P_t = Transmitted signal strength.
- P_{rssi} = received signal strength
- f = frequency 2.4 GHz
- N = power loss coefficient
- $L_f(n)$ = floor penetration loss
- n = difference of the floor between transmitter and receiver (n = 0 if the rooms were on the same floor)
- d = distance.

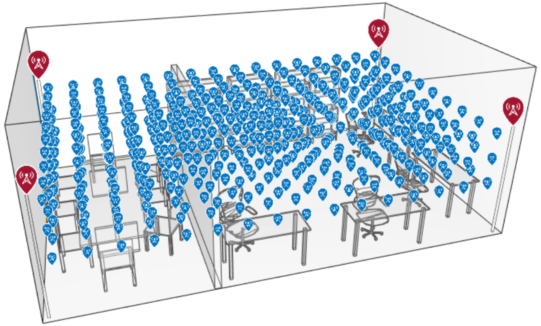
The value of P_t is given in the range 19 - 32 dBm depending on frequency, floor differences and different materials in the environment.

Fingerprinting Model based positioning

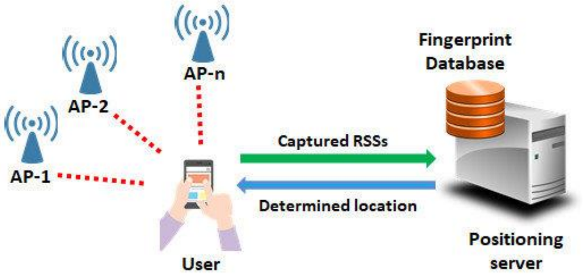
- Wi-Fi RTT (Round-Trip Time) fingerprinting is a technique used to improve indoor positioning accuracy by creating a database of signal characteristics at various locations within an environment.
- This database, or "fingerprint," is then used to estimate the position of a device based on real-time measurements. Here are the steps involved in Wi-Fi RTT fingerprinting:



Fingerprinting Model

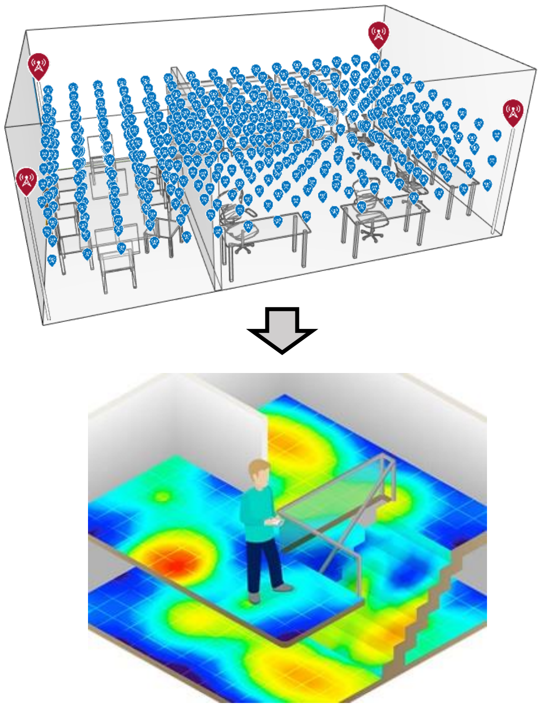


Grid setup

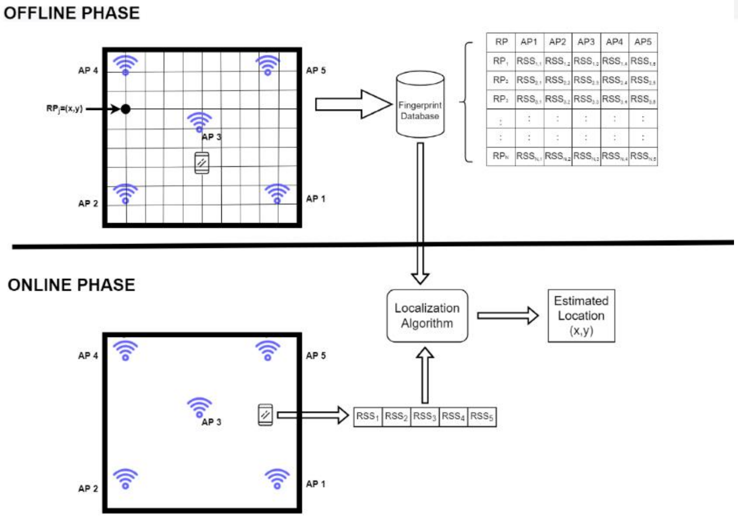


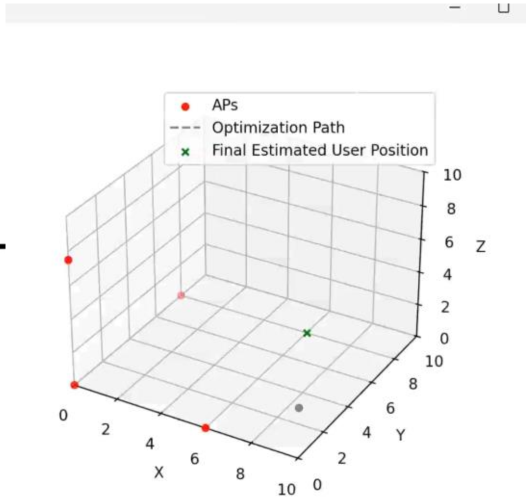
Positioning engine

Fingerprinting Model



Fingerprinting Model





- The Wi-Fi RTT positioning accuracy is around 1-2 meters.
- The accuracy is limited by the scenarios, for example, the signal reflections from buildings.

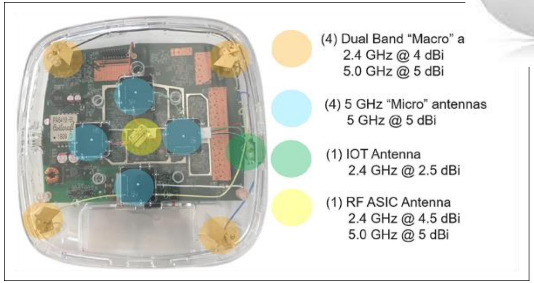
Wi-Fi Round-trip Timing (RTT) Positioning

Devices That Support Wi-Fi RTT

Nest Wifi Pro (Wi-Fi 6E)

Compulab WILD AP

Aruba AP-635



Cisco Series

Wi-Fi Access Point

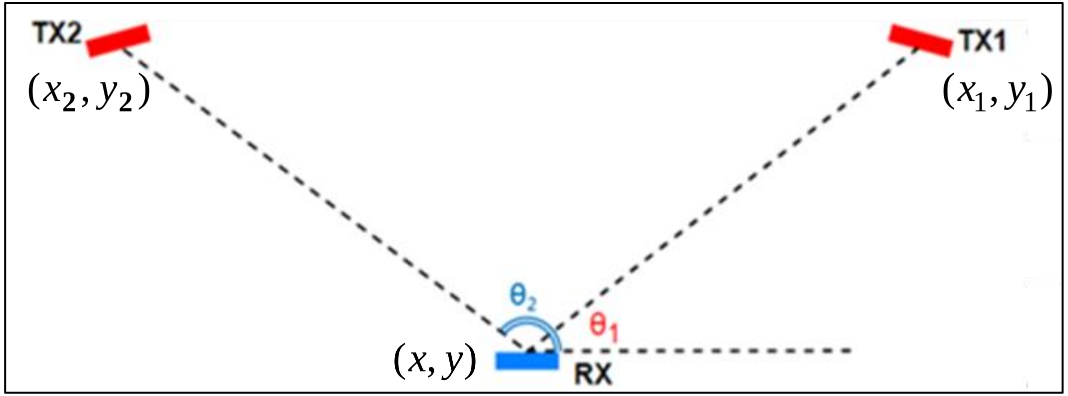
All Google Pixel Series

Xiaomi CC9 and Above Versions

Samsung Galaxy Note 10 and Above Versions

Phones

Example: AOA (Angle of Arrival) Model



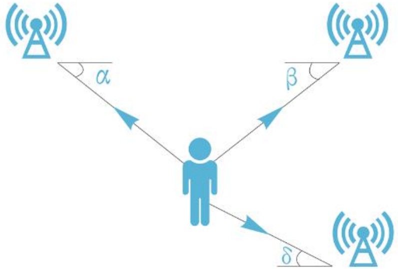
$$\frac{y_i - y}{x_i - x} = \tan\theta_i$$

$$\begin{cases} (y_1 - y) = (x_1 - x)\tan\theta_1 \\ (y_2 - y) = (x_2 - x)\tan\theta_2 \end{cases}$$

$$\begin{bmatrix} \tan\theta_1 & -1 \\ \tan\theta_2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_1\tan\theta_1 - y_1 \\ x_2\tan\theta_2 - y_2 \end{bmatrix}$$

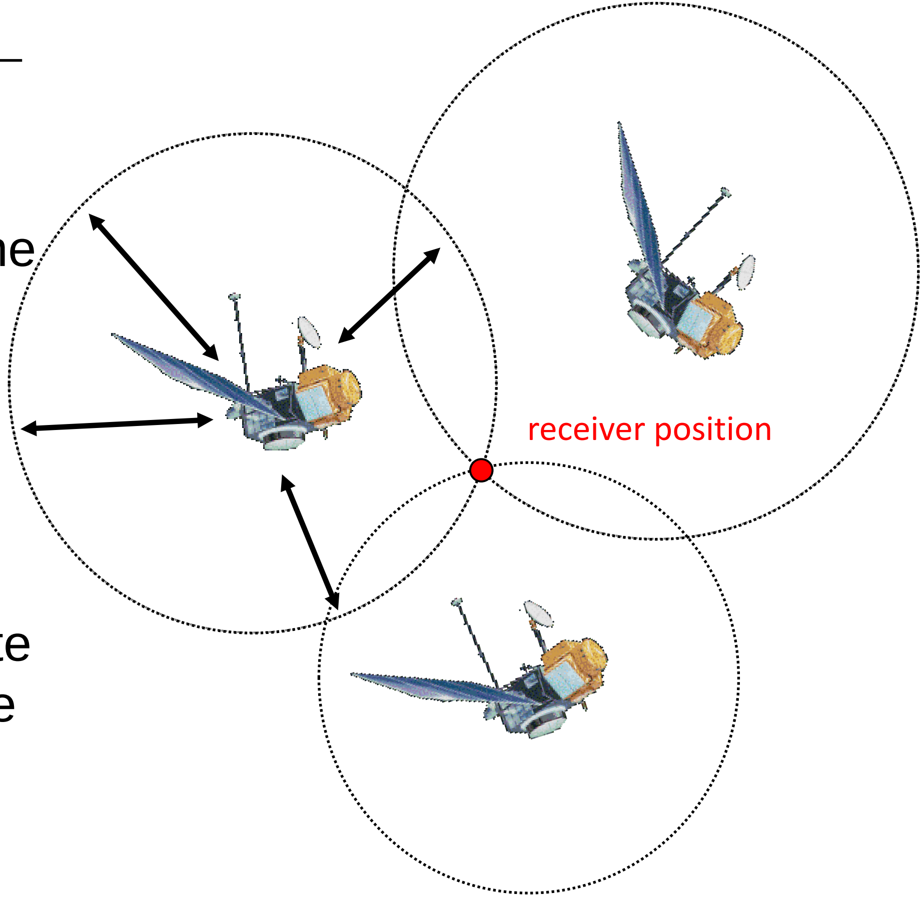
$$\mathbf{Ax} = \mathbf{b}$$

<u>Angle of Arrival (AoA)</u> is a method used to determine the direction from which a signal is being received. It is commonly used in navigation, surveillance, and wireless communication systems to locate the position of a transmitter or to track an object. AoA is based on measuring the angle at which the incoming signal arrives at a receiver or an array of receivers.



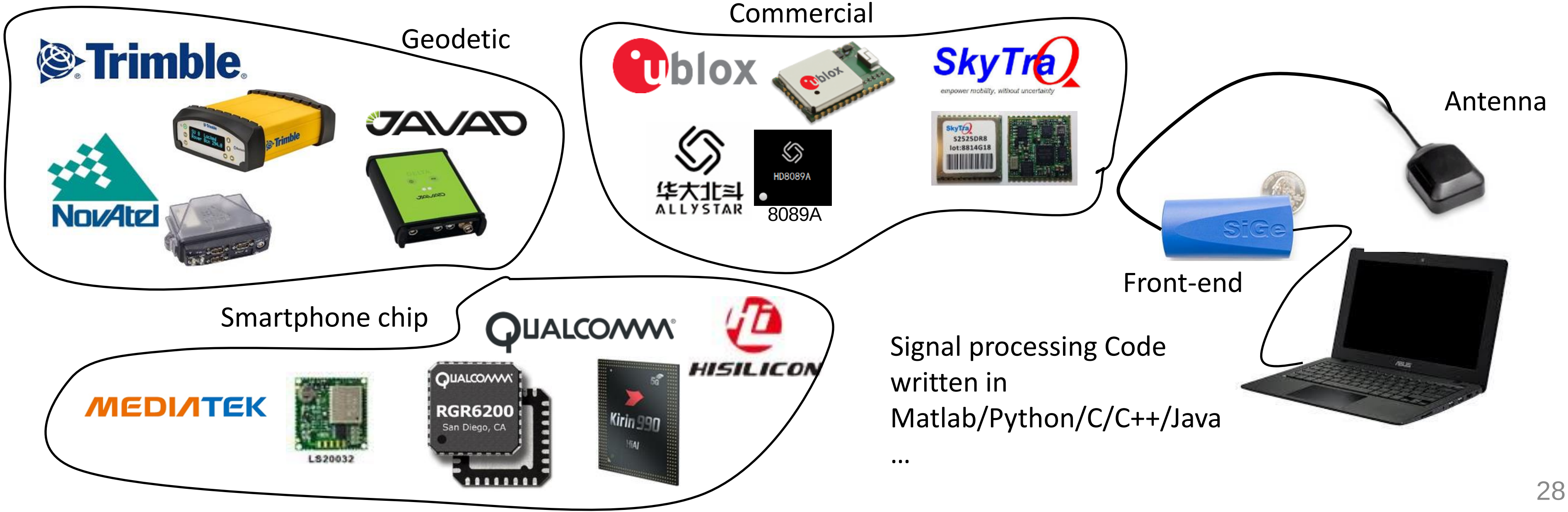
GPS Positioning

- GNSS Positioning is based on the triangulation method.
- Known information obtained from the signal processing
  - Position of satellites
  - Distance between satellites and receiver (Pseudoranges)
- The time difference between satellite and receiver is also estimated in the positioning process.
- At least 4 satellites are required.



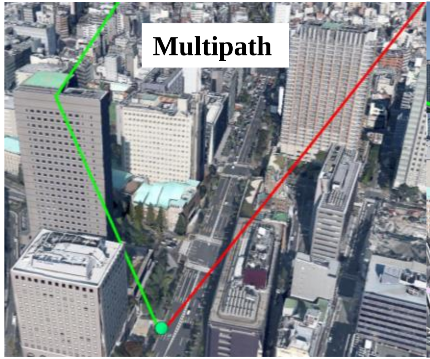
Examples of GPS Receivers

1. GPS receivers with quartz clocks can convert SV signals into position and time estimates and derive velocity.
2. Four satellites are required to compute the four dimensions of X, Y, Z (position) and Time.

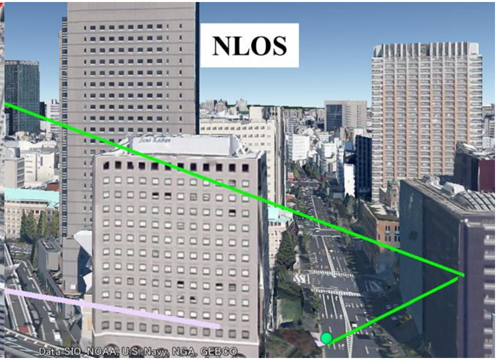


Background

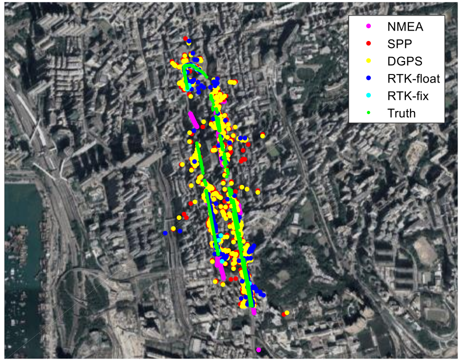
NLOS*: Non-line-of-sight

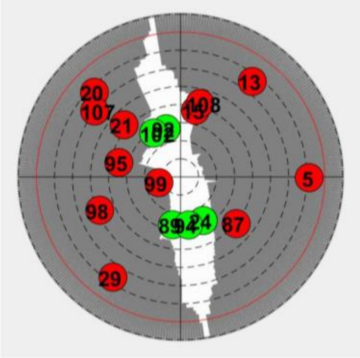


Hsu, 2016



Hsu, 2016





Satellite geometry in urban

Type	Availability	2D Error	X Error	Y Error	2D STD	X STD	Y STD
NMEA	91.43%	84.74	78.75	16.71	85.11	87.68	16.01
SPP	76.94%	51.49	31.71	32.28	61.28	49.48	43.73
DGNSS	72.07%	45.87	28.91	28.62	57.41	46.54	39.74

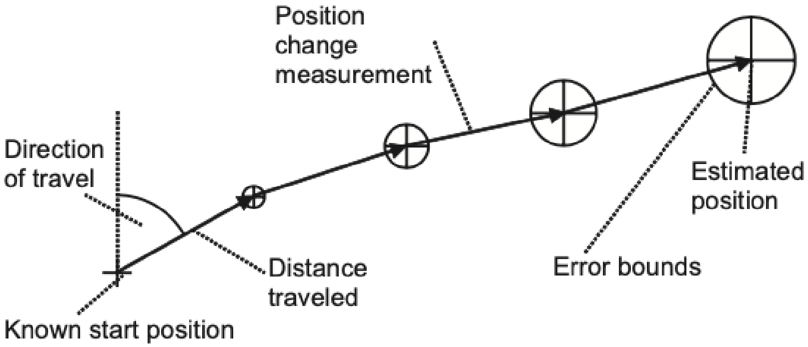
- GPS positioning accuracy is challenged in urban canyons.
- GPS positioning can provide absolute point positioning solution.
- More details about GPS model will be provided in the coming weeks.

[1]J. Breßler, etc, "GNSS positioning in non-line-of-sight context—A survey," *ITSC 2016*.
[2] Hsu, Li-Ta, etc. "3D building model-based pedestrian positioning method using GPS/GLONASS/QZSS and its reliability calculation." *GPS solutions*, 2016

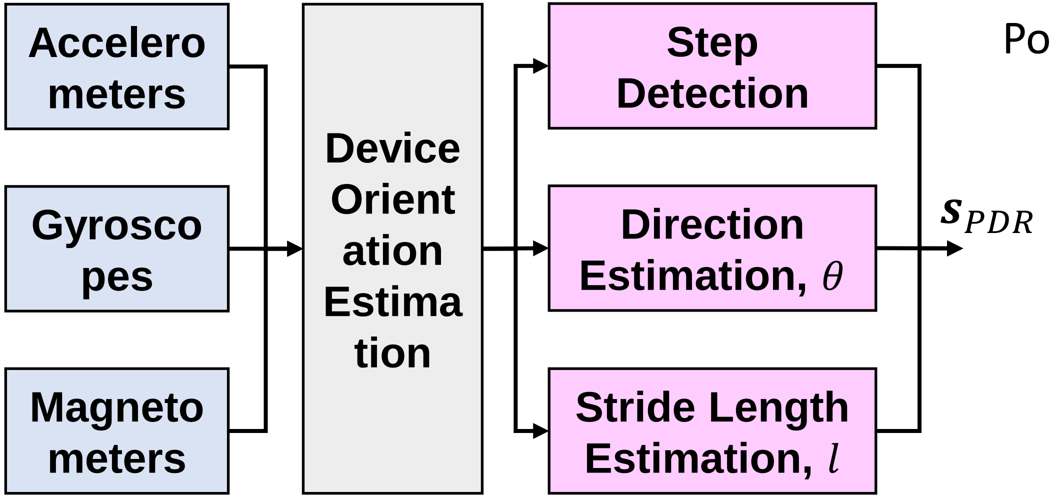
Dead Reckoning

- Dead reckoning either measures the change in position or measures the velocity and integrates it. Therefore, if the initial position is known, the current position may be determined as shown in the figure.
- For two-dimensional navigation, a heading measurement is sufficient, whereas for three-dimensional navigation, a full three-component attitude measurement is needed.

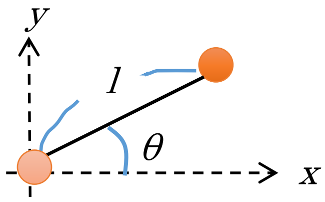
Heading may be measured using a *magnetic compass*. This is an ancient technology, although today magnetic compasses and magnetometers are available with electronic readouts.

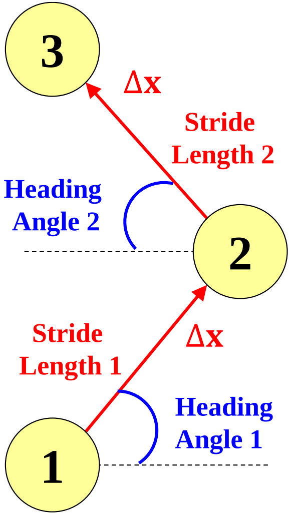


Pedestrian Dead Reckoning (PDR)



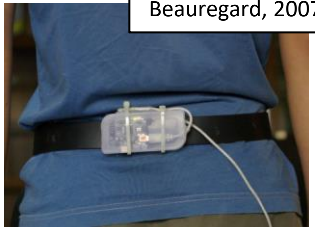
Positioning Update





$$\Delta\mathbf{x} = \begin{bmatrix} x_{east} \\ x_{north} \end{bmatrix} = f_{PDR}(\boldsymbol{s}_{PDR}) = step_{dct} \cdot \begin{bmatrix} l \cdot cos\theta \\ l \cdot sin\theta \end{bmatrix}$$

$$\boldsymbol{s}_{PDR} = \begin{bmatrix} l \\ \theta \end{bmatrix}, step_{dct} \in \langle 0,1 \rangle \qquad \mathbf{x}_n = \mathbf{x}_{n-1} + \Delta\mathbf{x}$$


Beauregard, 2007 [6]

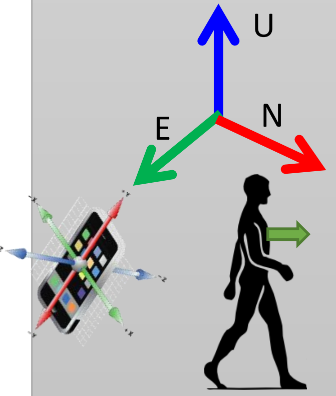

Goyal et al.,2011 [5]


Kamisaka et al., 2011 [8]


Steinhoff et al., 2010 [7]

Pedestrian Dead Reckoning (PDR)

Device Orientation Estimation

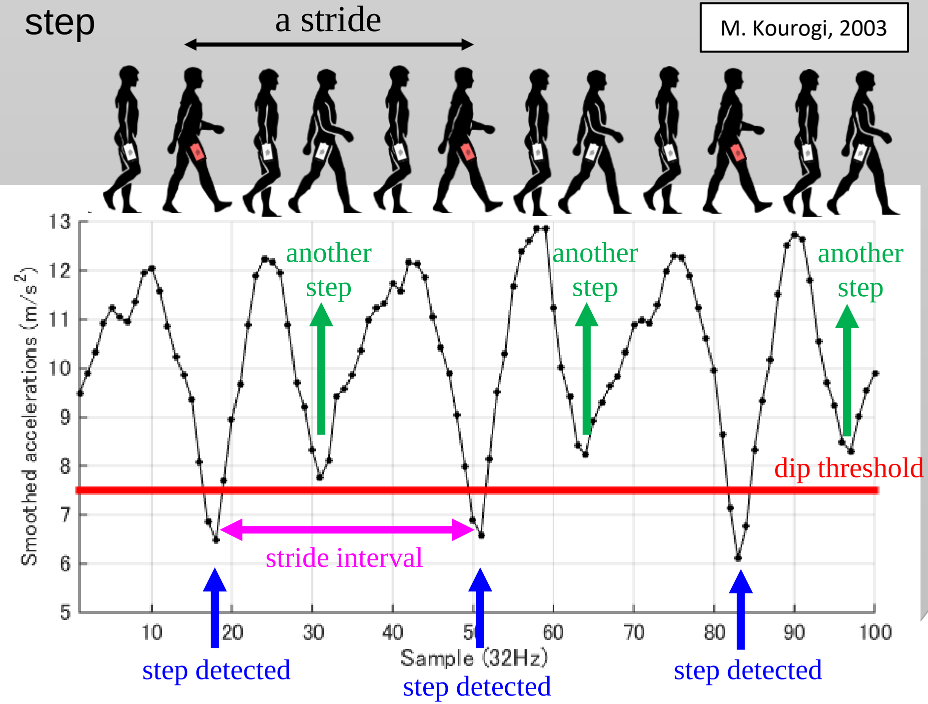


Transform The X-Y-Z (ECEF) measured data to E-N-U (Local) Coordinate

Step Detection

Detect the dip of the vertical acceleration data as one step

M. Kourogi, 2003

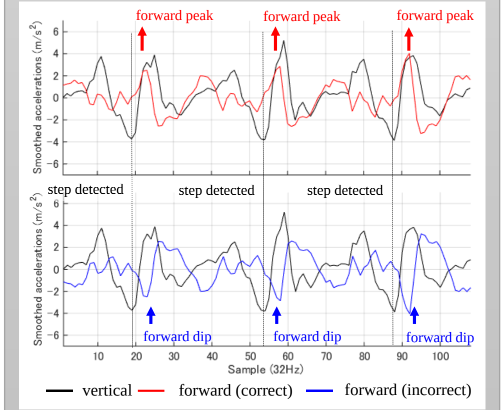


Stride Length Estimation

Using vertical acceleration (Weinberg equation [2])

$$l = K \sqrt[4]{a_{v,\max} - a_{v,\min}}$$

Moving Direction Estimation

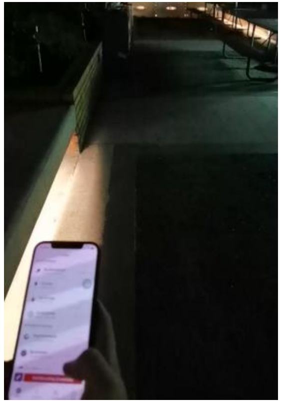


Applying PCA on acceleration in E-N plane (Steinhoff et al.'s method [1])

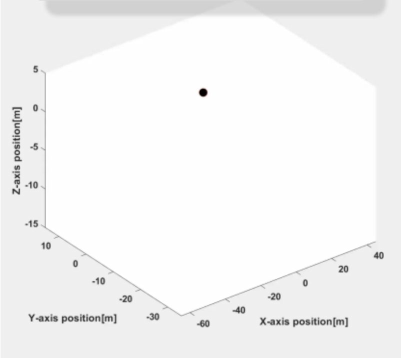
[1] H. Weinberg "Using the ADXL202 in Pedometer and Personal Navigation Applications," Analog Devices Inc. Application Note, 2002

[2] U. Steinhoff and B. Schiele "Dead Reckoning from the Pocket - An Experimental Study," *PerCom*, 2010

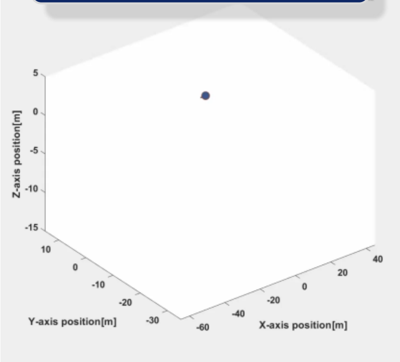
PDR: Multistorey Complex *Trajectory comparison*



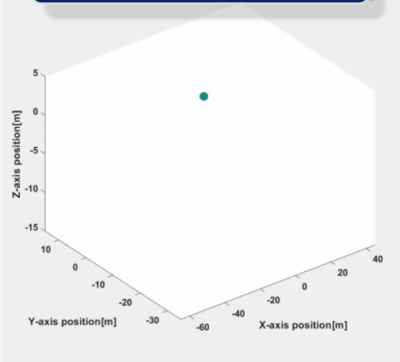
Ground Truth



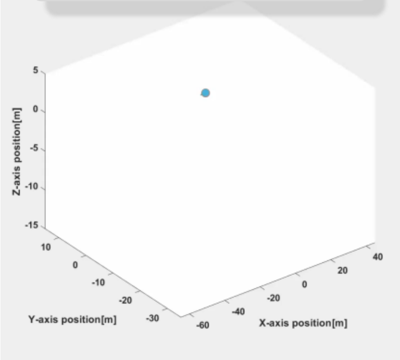
3D-PDR



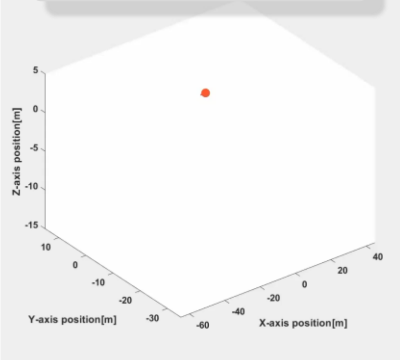
RoNIN



PF-SLAM

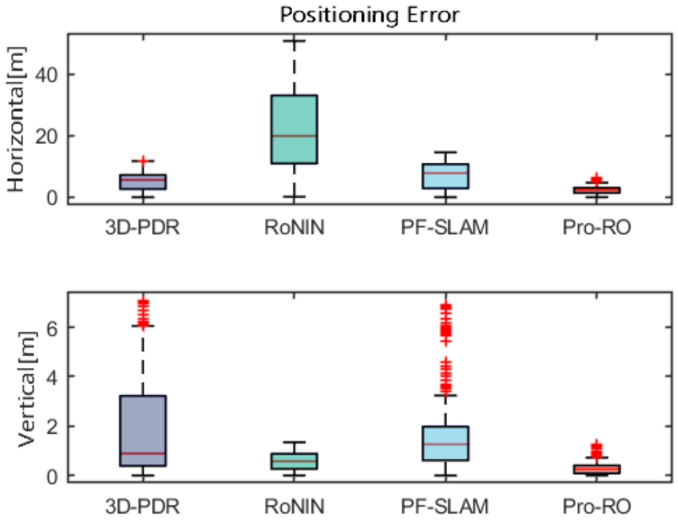


Pro-RO



PDR: Multistorey Complex *Trajectory comparison*

Position Error Comparison



ATE and RTE Comparison

Method	ATE (m)	RTE (m)
3D-PDR	4.51	3.61
RoNIN	18.62	11.60
PF-SLAM	6.07	5.95
Pro-RO	1.84	1.89

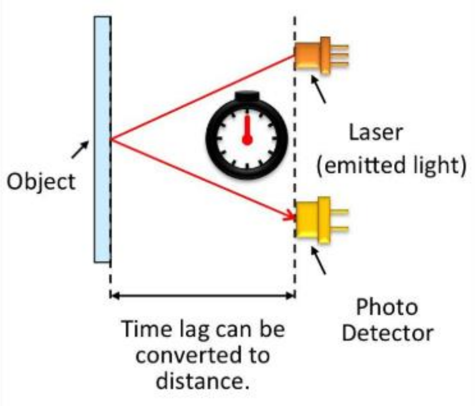
Light Detection and Ranging (LiDAR)

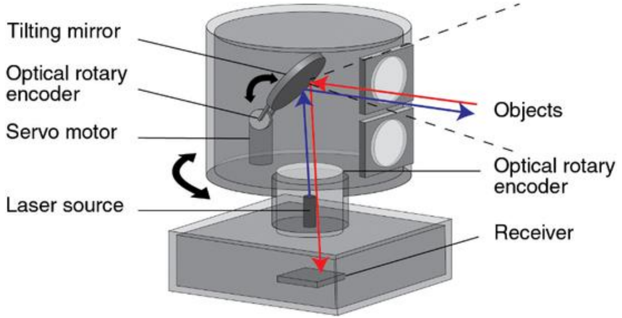
Laser signal emitted from a LiDAR reflect from objects both on and above the ground surface: vegetation, buildings, bridges, and so on. One emitted laser pulse can return to the LiDAR sensor as one or many returns.

The distance measurement's equation is given below,

Distance = (Speed of Light x Time of Flight) / 2

Multiple Channel LiDAR



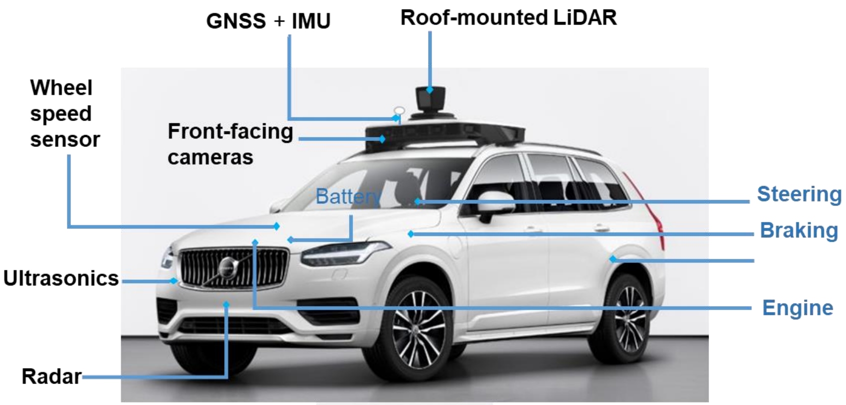


Source: Single Channel LiDAR

Source: Multiple Channel Rotating LiDAR

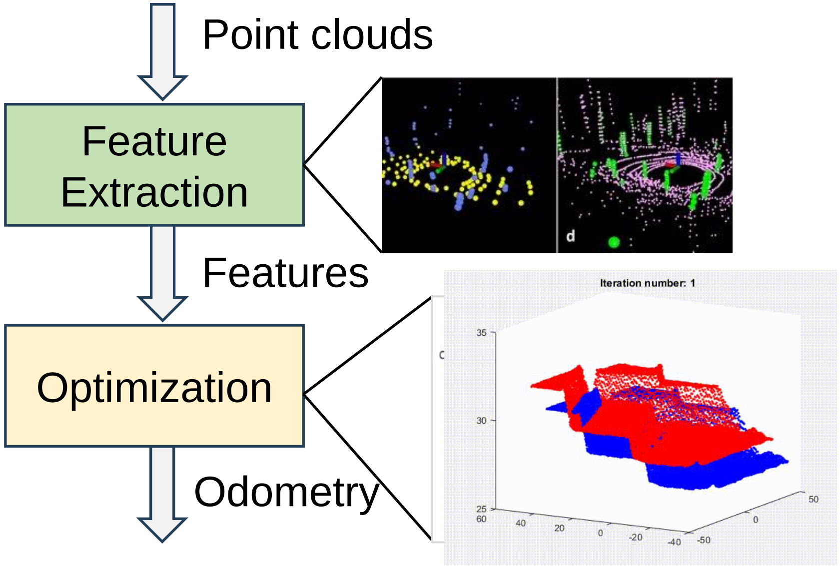
LiDAR based positioning (odometry)

<u>LiDAR odometry:</u> A technique used in the field of robotics, autonomous vehicles, and mapping to estimate the change in position over time of a device or vehicle, using data obtained from a Light Detection and Ranging (LiDAR) sensor.



Source: Volvo XC90

- Widely used in autonomous driving
- Limited by the high cost of the 3D LiDAR (~20K $HK).

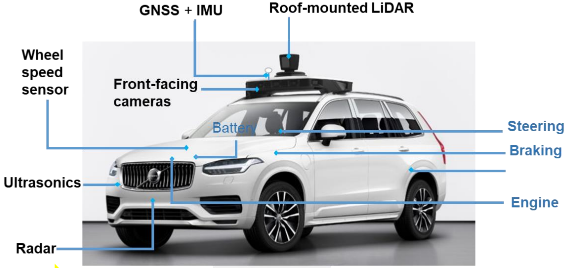


Light Detection and Ranging (LiDAR)

2007

The HDL-64E becomes the first commercially available
0.6 Million HK Dollar

Google (now Alphabet) begins testing self-driving cars on San Francisco Bay Area streets using Velodyne's lidar technology. Alphabet's first self-driving car prototype uses Velodyne's HDL-64E lidar sensor



Source: Volvo XC90

2014

Velodyne launched the Pucks series

QUANERGY

OUSTER

HESAI

History of mainstream 3D LiDAR sensors

2016

Velodyne expands the Puck family with the launches of three sensors: Puck Lite, Puck Hi-Res, and Ultra Puck.

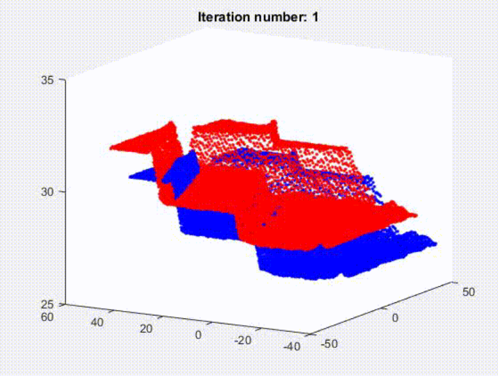
Solid-state LiDAR attracted increasing attention
10K HK Dollar

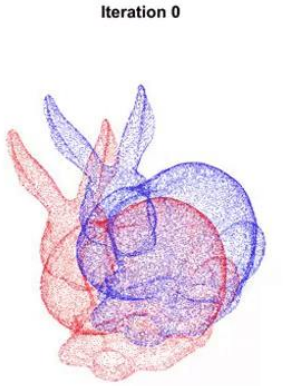
Prices of the Mainstream LiDAR

	Product	Inc	Type	Price
	HDL-64E	Velodyne	Mechanical LiDAR	**$75,000**
	HDL-32E	Velodyne	Mechanical LiDAR	**$30,000**
	VLP-16	Velodyne	Mechanical LiDAR	**$4,000**
	OS1	Ouster	Mechanical LiDAR	**$3,500**
	Horizon	Livox	Solid-State LiDAR	**$800**
	Velarray H800	Velodyne	Solid-State LiDAR	**$500**

LiDAR for autonomous driving is getting cheaper and cheaper!

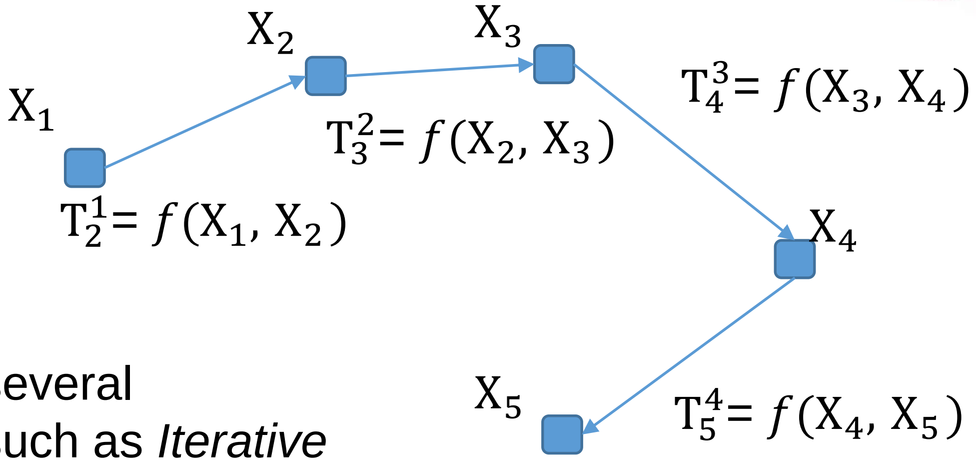
LiDAR Localization Method





X_{k+1} and X_k are two consecutive frames of point cloud

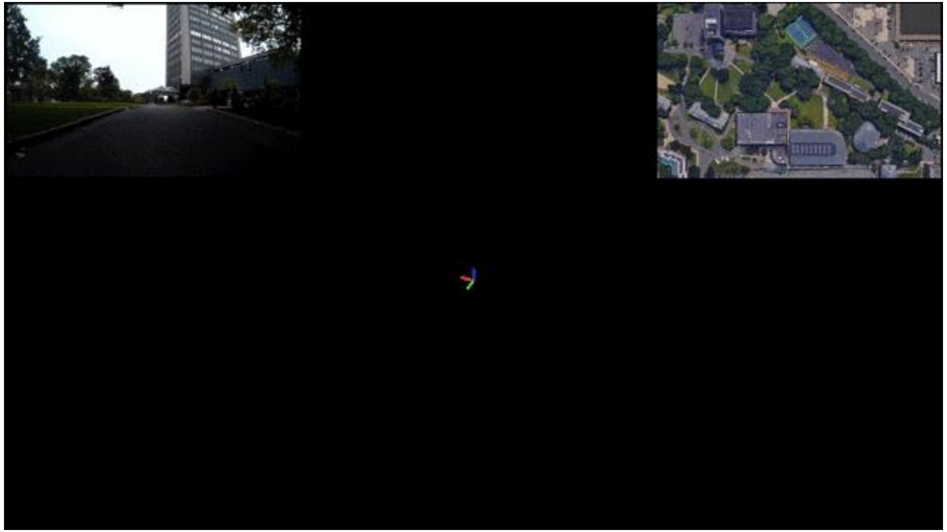
$$T_{k+1}^{k} = f(X_k, X_{k+1})$$

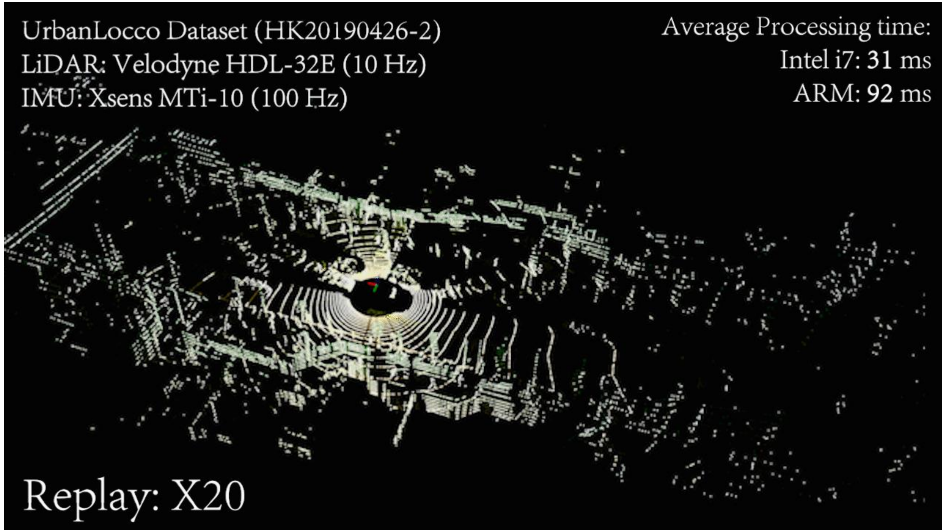


The function $f(X_k, X_{k+1})$ denotes several algorithms for LiDAR localization, such as *Iterative Closest Point*(***ICP***), *Normal Distributions Transform*(***NDT***),et al.

Huang, F., Wen, W., Zhang, J., & Hsu, L. T. (2021). Point wise or Feature wise? Benchmark Comparison of Public Available LiDAR Odometry Algorithms in Urban Canyons. *arXiv preprint arXiv:2104.05203*.

LiDAR Localization Method

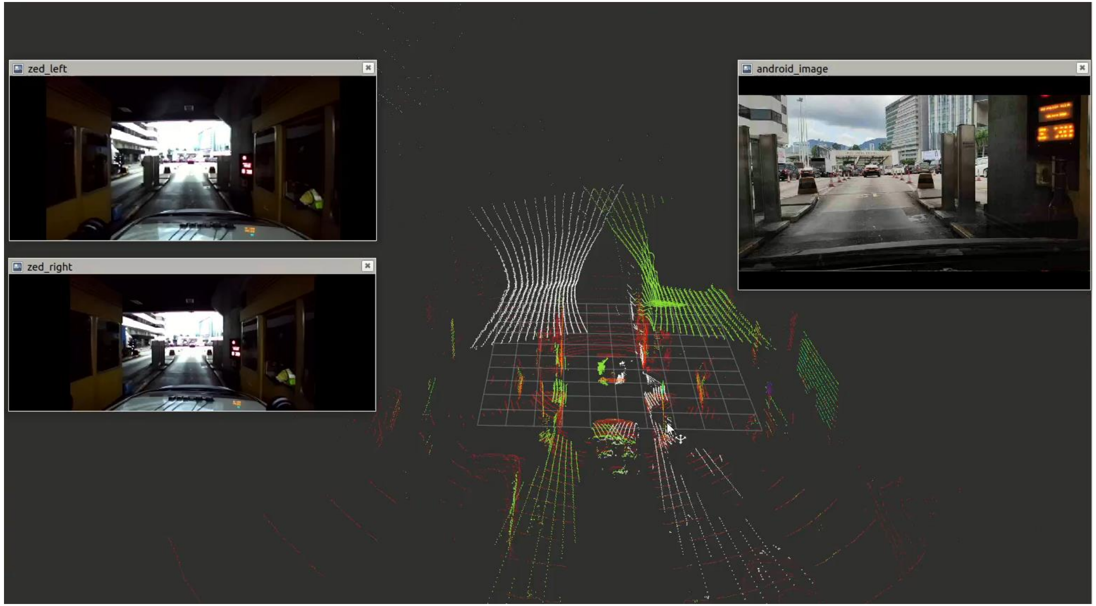




Pros of LiDAR localization: very accurate and high frequency pose estimation over time. The map of the environment can be generated simultaneously.

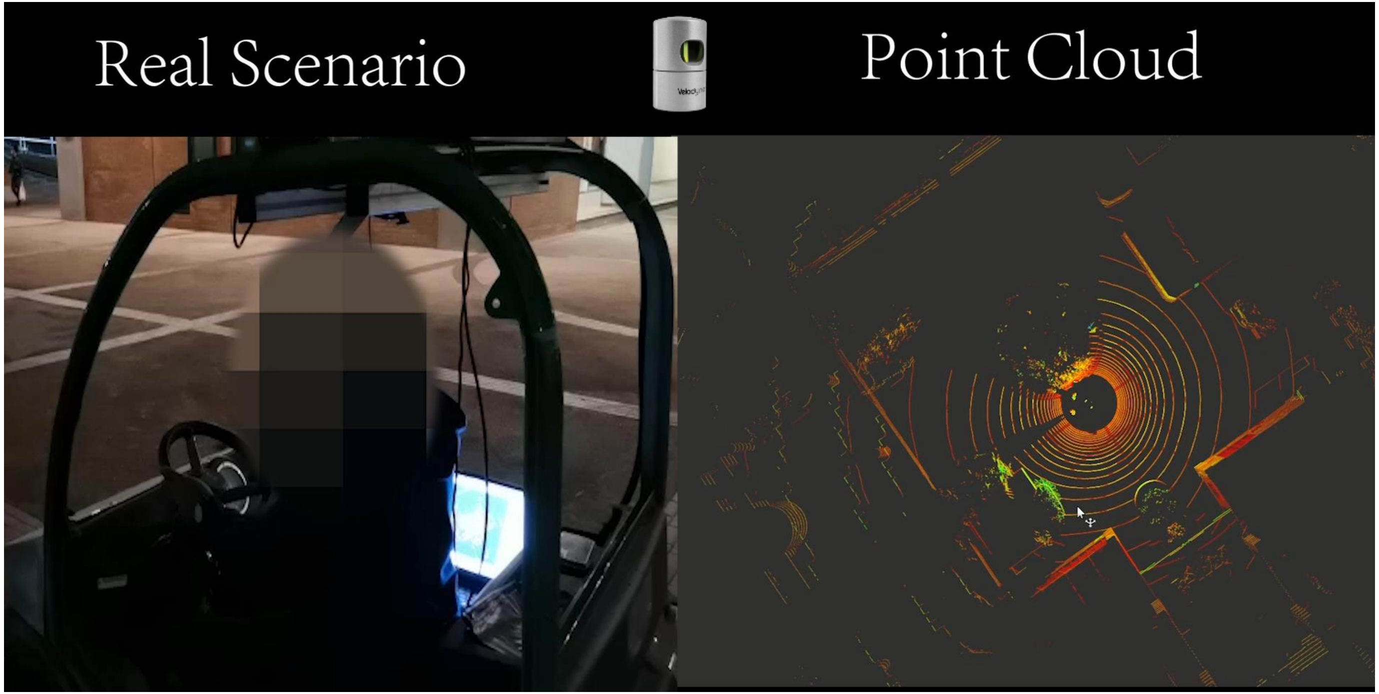
Cons of LiDAR localization: The localization result is subject to drift over time. The LiDAR localization can fail in feature insufficient environments.

Multiple LiDARs for Autonomous Vehicle Navigation



Testing of the multiple 3D LiDARs for typical autonomous vehicles setups.

LiDAR for Autonomous Vehicle Navigation



Testing of the autonomous driving vehicles in PolyU campus.
Only a 3D LiDAR is employed for navigation.

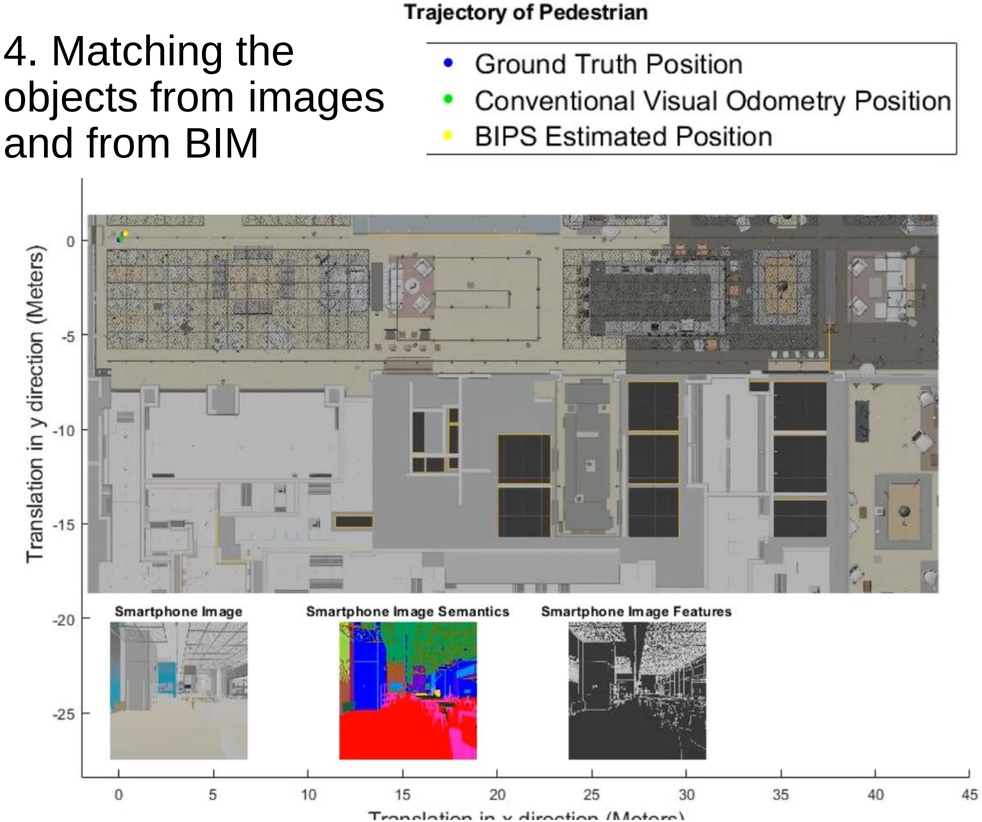
Visual indoor positioning using semantic information

1. Taking images to get ready for object recognition

2 Object recognition and classification based on deep learning

3. Object extraction from building information model (BIM) from grids

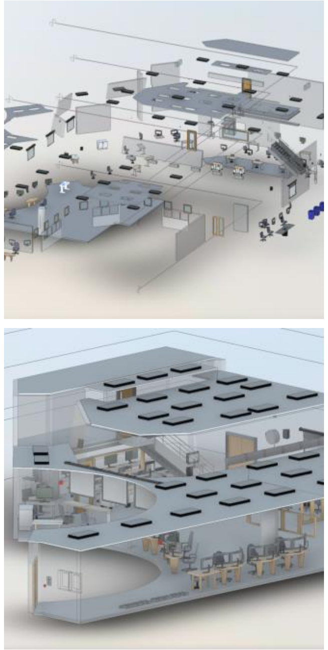
4. Matching the objects from images and from BIM

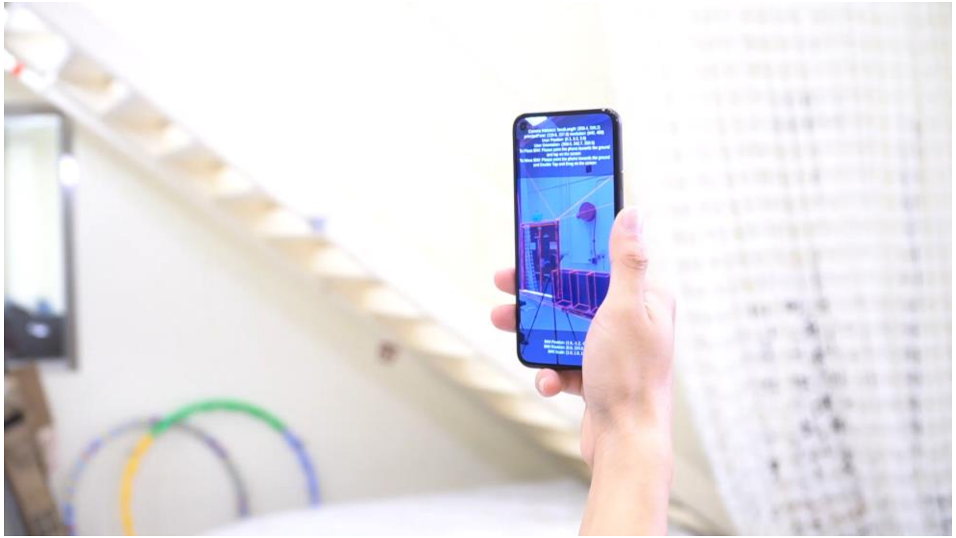


M. Lee, H.-Y. Ho, **Li-Ta Hsu***, S. Au, BIPS: Building Information Positioning System, *IEEE IPIN 2021*

Visual indoor positioning using semantic information

Server setup with BIM → Turn on Camera → Point at Asset → Instant asset position and informatics!





Coordinate Systems

Coordinate Systems

- A significant problem to overcome when using a navigation system is the fact that there are a great number of different coordinate systems worldwide.
- As a result, the position measured and calculated does not always correspond with one's supposed position.

Big Ben, London, UK

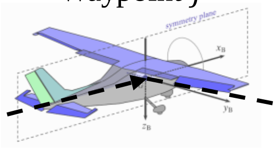
Lat./Long./Alt.	51.5007° N	0.1246° W	72m
X/Y/Z-WGS84	3978622.9m	8652.2m	4968467.2m
E/N/U relative to Westminster Station	45.3m	-32.5m	68.9m
front/right/up (relative to your body)	-30.8m	20.3m	69.9m

Why we need the navigation coordinate system?

Scenario: A planned flight from Hong Kong airport to the Los Angeles.
Question: How can I know the distance from current (e.g., waypoint j) to the destination?

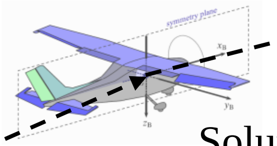
Waypoint j: (x_j, y_j, z_j)

Waypoint j



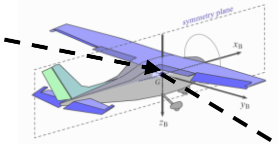
Waypoint i: (x_i, y_i, z_i)

Waypoint i



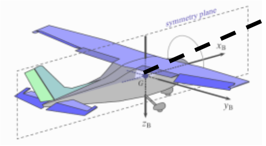
Waypoint k: (x_k, y_k, z_k)

Waypoint k



Waypoint 0: (x_0, y_0, z_0)

Waypoint 0

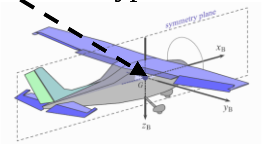


Source: Euler Angles, Tait–Bryan angle

Solution: define a global navigation coordinate system to denote the position at each waypoint!

Waypoint n: (x_n, y_n, z_n)

Waypoint n



Hong Kong

$$\text{Distance: } D_{jn} = \sqrt{\left((x_j - x_n)^2 + (y_j - y_n)^2 + (z_j - z_n)^2\right)}$$

Question: how to define the global coordinate system applicable for both HK and Los Angeles?

Coordinate Systems

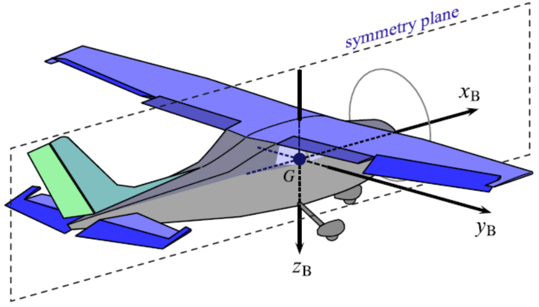
Any navigation problem thus involves at least two coordinate frames. One describing the body frame and one as the reference frame. Any two coordinate frames may have any relative orientation, known as **attitude**.

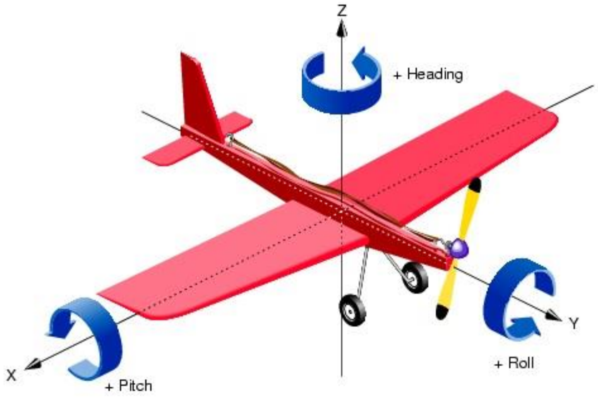
The main coordinate systems used in navigation: *Earth-centered inertial (ECI), Earth-centered Earth-fixed (ECEF) and body frames*

In physics, any coordinate frame that does not accelerate or rotate with respect to the rest of the Universe is an *inertial frame*.

Body Navigation Frame - Body

- It represents the orientation of the body to which it is connected.
- Its origin is described by the aerial vehicle (sometime is the center gravity of sensors)
- The axes are
- X^b –axis: pointing towards the right to the direction of motion
- Y^b –axis: pointing towards the front (in the direction of motion)
- Z^b–axis: pointing up to complete the orthogonal right-hand
  Applications
  - Attitude derivation
  - Flight control system
  - Simultaneous localization and mapping (SLAM)

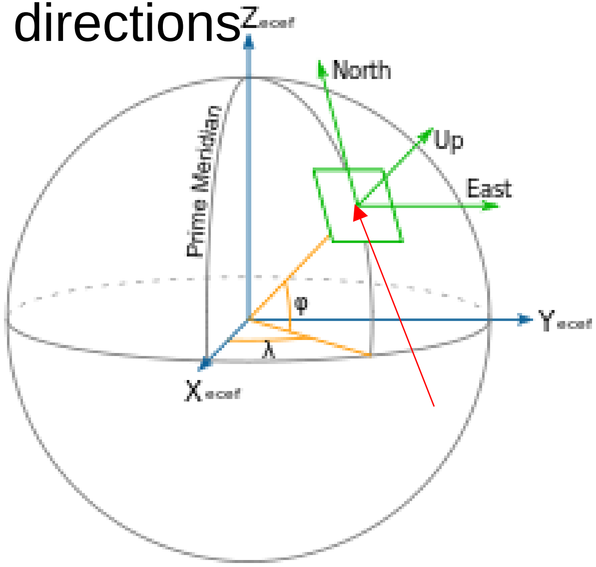




Local Navigation Frame - ENU

Origin defined by User

- Local (Relative) Frame.
- Its origin is described by a navigation object (usually in ECEF).
- The axes are aligned with topographic directions North, East and Vertical (Up).

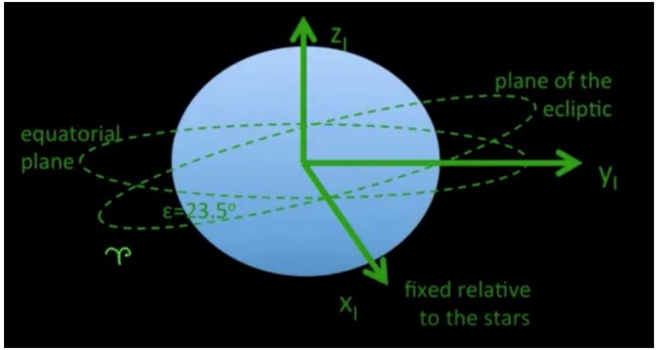


- (E, N, U) is used to denote position
- Applications
  - Robotic Navigation
    Since the user wants to know his/her position/attitude relative to the north, east and vertical direction.

https://en.wikipedia.org/wiki/Axes_conventions

Earth Centered Inertial (ECI)

The origin is at the center of the mass of the Earth and whose axes are pointing in **fixed directions** with respect to the stars, which does not rotates with the earth.

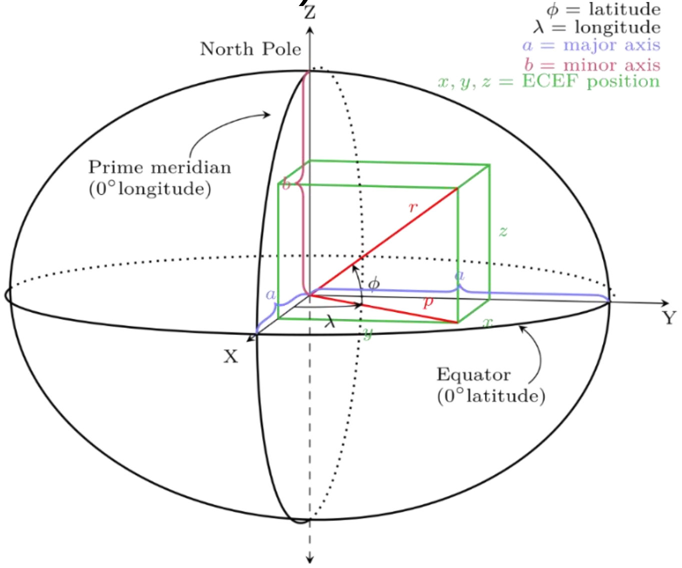


- The *z*-axis always points along the Earth's axis of rotation from the frame's origin at the center of mass to the true north pole (not the magnetic pole).
- The *x*- and *y*-axes lie within the equatorial plane, but do not rotate with the Earth. +*x*-axis is permanently fixed in a particular direction relative to the celestial sphere.
- The *y*-axis points 90° ahead of the *x*-axis in the direction of the Earth's rotation.

Earth-Centered, Earth-Fixed - ECEF

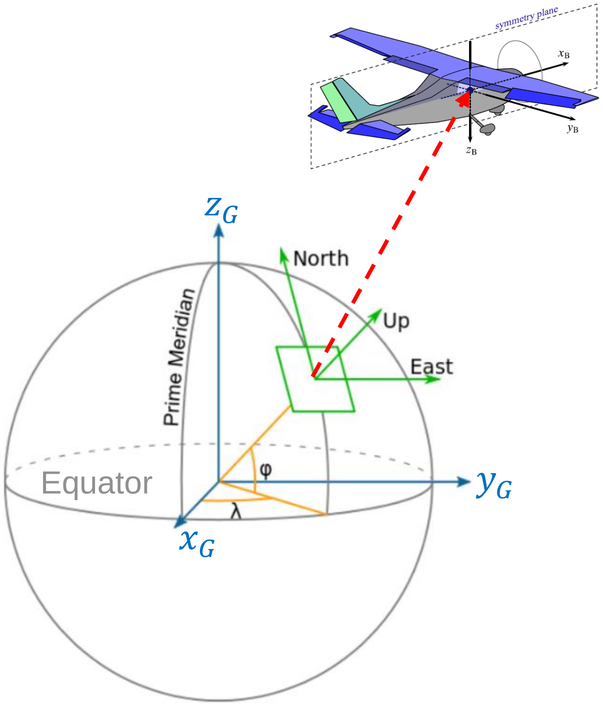
IERS*: Earth Rotation and Reference Systems Service

- Similar to ECI except all axes remain fixed w.r.t Earth.
- Z-axis is rotated with the Earth spin axis (same with ECI).
- X-axis points to 0º longitude defined by IERS.
- Y-axis is orthogonal with X-axis.
- (0,0,0) means Center of Earth Mass
- Both (X, Y, Z) and (Lat, Lon, Alt) can be used to denote positions
- Applications
  - GPS positioning



https://en.wikipedia.org/wiki/ECEF

Body && ENU && ECEF



λ: Longitude
φ: Latitude

Earth-centered earth-fixed (ECEF) coordinate system, $\mathbf{C}_G(x_G, y_G, z_G)$

- Origin at the center of mass of earth
- x_G extends through the intersection of the prime meridian Greenwich and the equator.
- z_G points towards geographical north
- y_G completes the right handset of coordinates axis

East-North-Up (ENU) coordinate system, $C_L(x_L, y_L, z_L)$

- Local tangent plane coordinates based on **a selected reference point**
- The east axis is labeled x, the north y and the up z.
- Rotate with the earth
- Useful to describe motion of objects on earth surface.

Representation of Position and Attitude

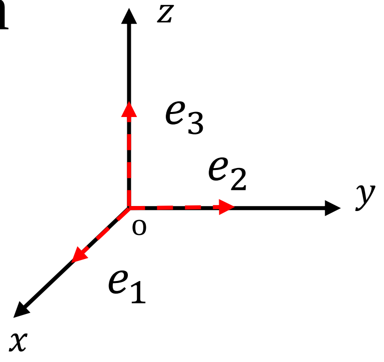
Prerequisite and Notation

Prerequisite :

- Matrix and vector calculation
- Unit orthogonal basis of coordinate system

Notation :

- Matrix with bold upper case (**R**)
- Vector with bold lower case (**v**)
- Scalar with low case italic (*k*)



Unit orthogonal basis $[\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3]$, which satisfying

$$[\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3]\begin{bmatrix}\mathbf{e}_1\\ \mathbf{e}_2\\ \mathbf{e}_3\end{bmatrix} = [1]$$

Mathematical Foundations- Matrix Calculations

$$\begin{cases} 3x + 4y + z = 9 \\ 2x - 6y + 3z = 8 \\ 7x + 5y - 7z = 1 \end{cases}$$

3 linear equations with 3 unknowns

Stack into matrix form

$$\underbrace{\begin{bmatrix} 3 & 4 & 1 \\ 2 & -6 & 3 \\ 7 & 5 & -7 \end{bmatrix}}_{\text{Matrix } \mathbf{A}} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\text{Vector } \boldsymbol{x}} = \underbrace{\begin{bmatrix} 9 \\ 8 \\ 1 \end{bmatrix}}_{\text{Vector } \boldsymbol{b}} \Rightarrow \mathbf{A}\boldsymbol{x} = \boldsymbol{b}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 & 4 & 1 \\ 2 & -6 & 3 \\ 7 & 5 & -7 \end{bmatrix}^{-1} \begin{bmatrix} 9 \\ 8 \\ 1 \end{bmatrix} \Rightarrow \boldsymbol{x} = \mathbf{A}^{-1}\boldsymbol{b}$$

Condition for matrix multiplication: $\mathbf{AB}$ → The columns of $\mathbf{A}$ should be the same as the row of $\mathbf{B}$

Condition for matrix inverse: $\mathbf{A}^{-1}$ → The $\mathbf{A}$ is an full rank matrix

Transpose of matrix: $\mathbf{A}^{\mathrm{T}}$

Identity matrix: $\mathbf{I}_{n\times n}$

Mathematical Foundations- Matrix Calculations

Add operation between matrix: $\mathbf{A} + \mathbf{B}$

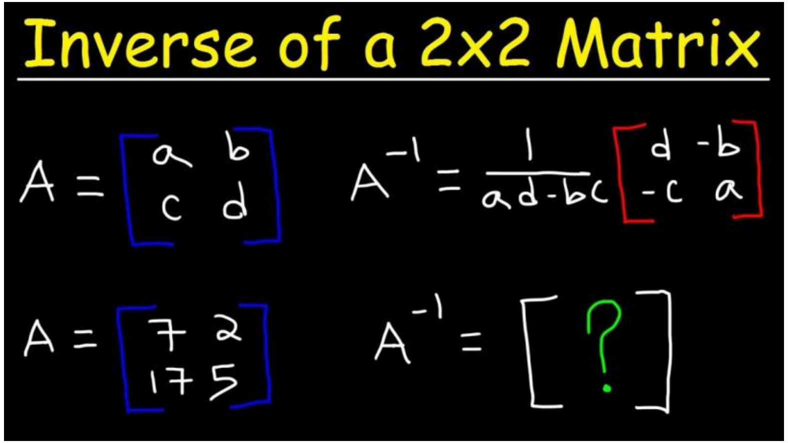
$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & -6 & 3 \\ 7 & 5 & -7 \end{bmatrix} + \begin{bmatrix} 0 & 4 & 1 \\ 2 & 1 & 1 \\ 7 & 5 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 8 & 2 \\ 4 & -5 & 4 \\ 14 & 10 & -5 \end{bmatrix}$$

Multiply operation between matrix: $\mathbf{A} * \mathbf{B}$

$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & -6 & 3 \\ 7 & 5 & -7 \end{bmatrix} \begin{bmatrix} 0 & 4 & 1 \\ 2 & 1 & 1 \\ 7 & 5 & 2 \end{bmatrix} = \begin{bmatrix} 15 & 21 & 9 \\ 9 & 17 & 2 \\ -39 & -2 & -2 \end{bmatrix}$$

Transpose of matrix: $\mathbf{A}^{\mathrm{T}}$

$$\begin{bmatrix} 3 & 4 & 1 \\ 2 & -6 & 3 \\ 7 & 5 & -7 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 3 & 2 & 7 \\ 4 & -6 & 5 \\ 1 & 3 & -7 \end{bmatrix}$$



Take a try on this ☺

Try Yourself

Add operation between matrix: $\mathbf{A}+\mathbf{B}$

$$\begin{bmatrix}1 & 3 & 2\\ 1 & -1 & 0\\ 2 & 6 & -2\end{bmatrix}+\begin{bmatrix}4 & 2 & 3\\ 2 & 1 & 5\\ 2 & 3 & 7\end{bmatrix}=?$$

Results:

$$\begin{bmatrix}5 & 5 & 5\\ 3 & 0 & 5\\ 4 & 9 & 5\end{bmatrix}$$

Multiply operation between matrix: $\mathbf{A}*\mathbf{B}$

$$\begin{bmatrix}1 & 2 & 3\\ 3 & -5 & 2\\ 6 & 2 & -1\end{bmatrix}\begin{bmatrix}0 & 2 & 1\\ 1 & 3 & -1\\ 5 & 6 & 2\end{bmatrix}=?$$

$$\begin{bmatrix}17 & 26 & 5\\ 5 & 3 & 12\\ -3 & 12 & 2\end{bmatrix}$$

Transpose of matrix: $\mathbf{A}^{\mathrm{T}}$

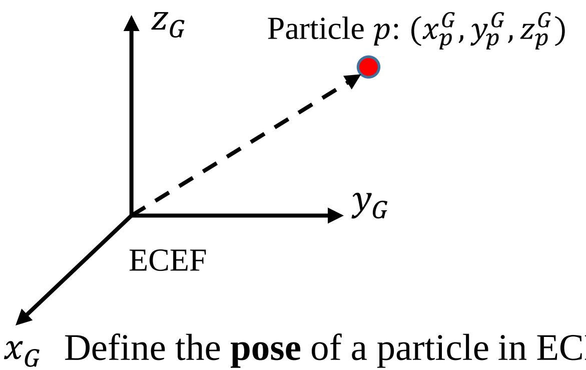
$$\begin{bmatrix}3 & 2 & 5\\ 2 & -1 & 4\\ 5 & 7 & -1\end{bmatrix}^{\mathrm{T}}=?$$

$$\begin{bmatrix}3 & 2 & 5\\ 2 & -1 & 7\\ 5 & 4 & -1\end{bmatrix}$$

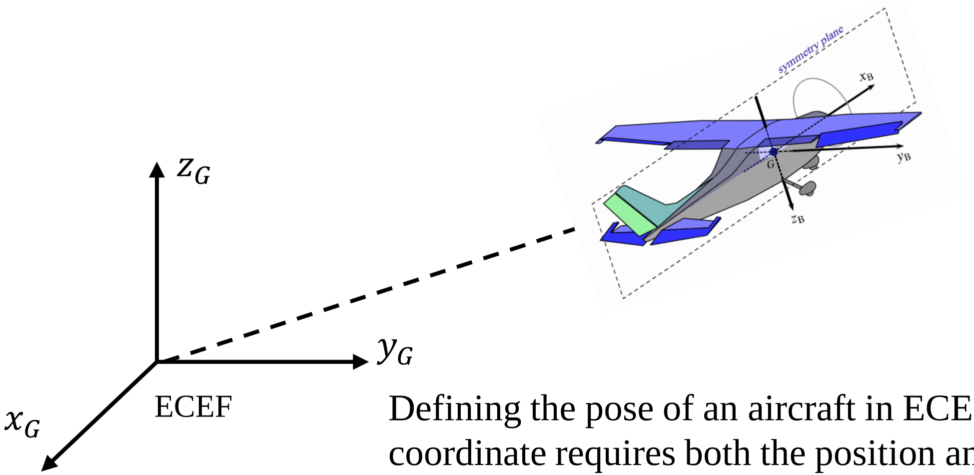
Pose Representation of UAV in Space

Scenario: A planned flight from Hong Kong airport to the Los Angeles.

Question: How to define the pose of a flight in the space?



Define the **pose** of a particle in ECEF coordinate as $\left(x_p^G, y_p^G, z_p^G\right)$. The G denotes the ECEF global coordinate system.



Defining the pose of an aircraft in ECEF coordinate requires both the position and orientation as $(\underbrace{x_p^G, y_p^G, z_p^G}_{\textbf{position}}, \underbrace{\phi_p^G, \theta_p^G, \psi_p^G}_{\textbf{orientation}})$

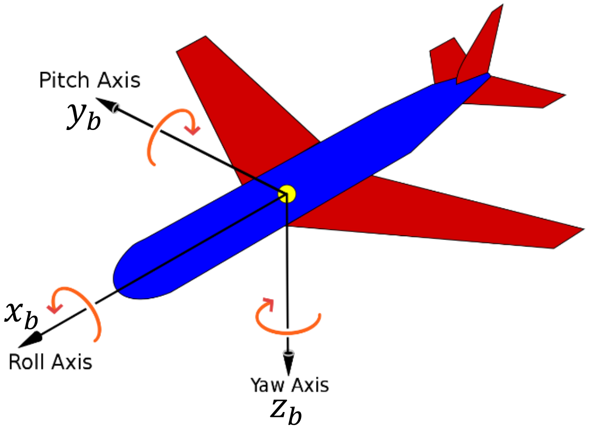
Pose

Pose Representation of UAV in Space

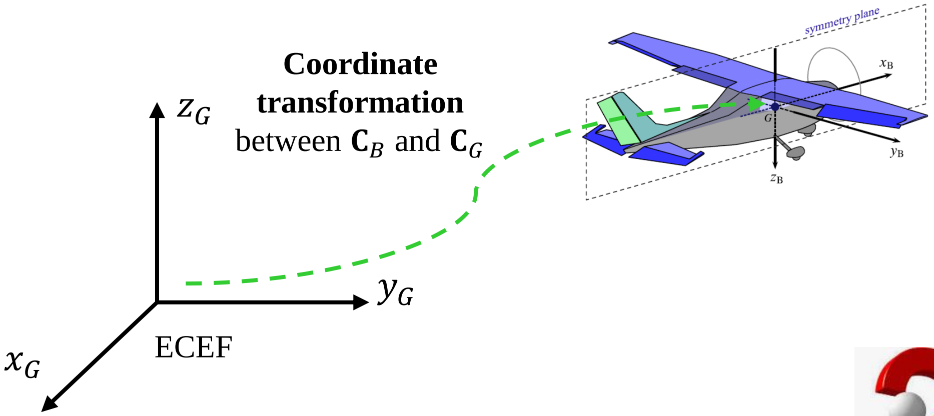
Pose of an aircraft in ECEF coordinate $(x_p^G, y_p^G, z_p^G, \phi_p^G, \theta_p^G, \psi_p^G)$:

- x_p^G, y_p^G, z_p^G, **position** in ECEF frame
- $\phi_p^G, \theta_p^G, \psi_p^G$, the **orientation** (*roll*, *pitch* and *yaw angle*).

Since the body-fixed coordinate $\mathbf{C}_B$ is fixed on the flight mechanics. The **coordinate transformation** between $\mathbf{C}_B$ and $\mathbf{C}_G$ represents the **pose of the flight mechanics in the ECEF frame**!



Source: Euler Angles, Tait–Bryan angle



How to denote the coordinate transformation?

Rotation Representation with Matrix: Derivation from Orthogonal Basis

Define the unit orthogonal basis of $\mathbf{C}_G$ as $[\mathbf{e}_x^G, \mathbf{e}_y^G, \mathbf{e}_z^G]$ and the coordinate of vector $\mathbf{a}$ as $[a_x^G, a_y^G, a_z^G]$.

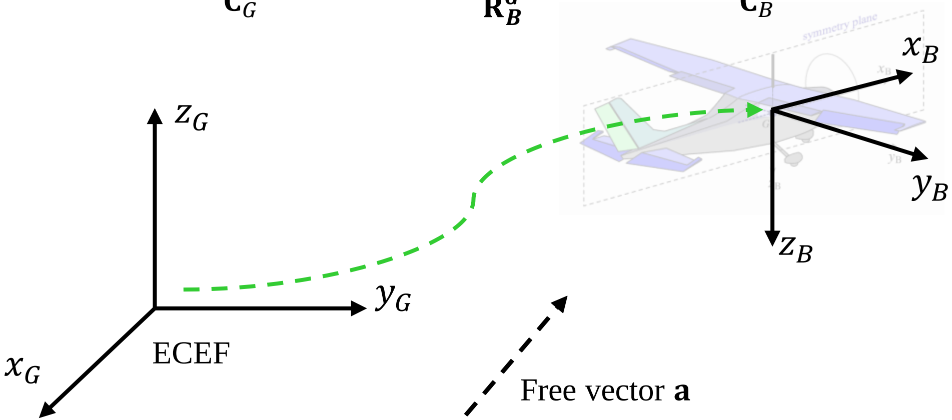
Define the unit orthogonal basis of $\mathbf{C}_B$ as $[\mathbf{e}_x^B, \mathbf{e}_y^B, \mathbf{e}_z^B]$ and the coordinate of vector $\mathbf{a}$ as $[a_x^B, a_y^B, a_z^B]$.

Since the vector $\mathbf{a}$ itself is **constant despite of the representation** in different coordinate systems. We have

$$[\mathbf{e}_x^G, \mathbf{e}_y^G, \mathbf{e}_z^G]\begin{bmatrix} a_x^G \\ a_y^G \\ a_z^G \end{bmatrix} = [\mathbf{e}_x^B, \mathbf{e}_y^B, \mathbf{e}_z^B]\begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix}$$

Multiply the $[\mathbf{e}_x^G, \mathbf{e}_y^G, \mathbf{e}_z^G]^T$ to both sides, we get

$$\underbrace{\begin{bmatrix} a_x^G \\ a_y^G \\ a_z^G \end{bmatrix}}_{\text{Position in } \mathbf{C}_G} = \underbrace{\begin{bmatrix} \mathbf{e}_x^G\mathbf{e}_x^B & \mathbf{e}_x^G\mathbf{e}_y^B & \mathbf{e}_x^G\mathbf{e}_z^B \\ \mathbf{e}_y^G\mathbf{e}_x^B & \mathbf{e}_y^G\mathbf{e}_y^B & \mathbf{e}_y^G\mathbf{e}_z^B \\ \mathbf{e}_z^G\mathbf{e}_x^B & \mathbf{e}_z^G\mathbf{e}_y^B & \mathbf{e}_z^G\mathbf{e}_z^B \end{bmatrix}}_{\text{Rotation Matrix } \mathbf{R}_B^G} \underbrace{\begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix}}_{\text{Position in } \mathbf{C}_B}$$



The rotation between two coordinate systems can be represented by the rotation matrix $\mathbf{R}_B^G$!

Rotation and Position Representation

Given

- The position of a particle **a** in the body-fixed coordinate as (a_x^B, a_y^B, a_z^B)
- The transformation between between $\mathbf{C}_B$ and $\mathbf{C}_G$ as rotation matrix $\mathbf{R}_B^G$ and translation vector $\mathbf{t}_B^G(x_B^G, y_B^G, z_B^G)$

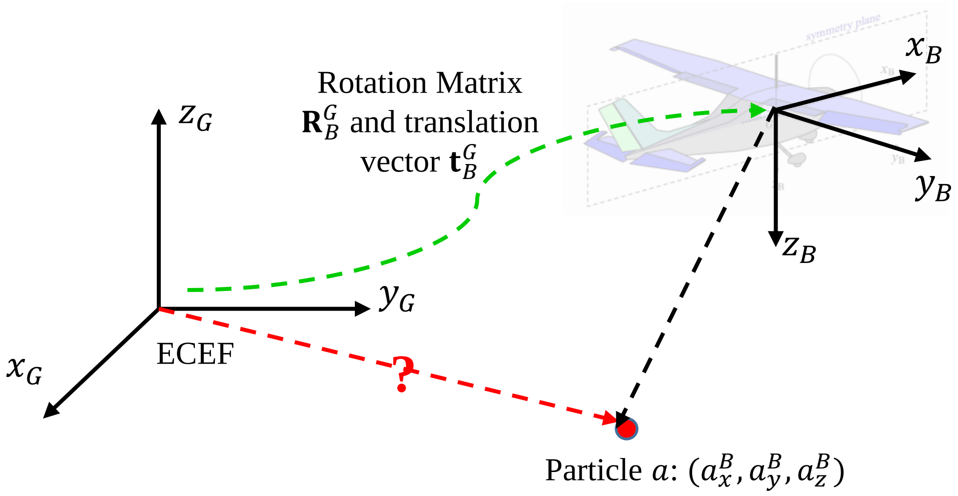
Question:

- Calculate the coordinate of particle **a** in the coordinate $\mathbf{C}_G$.

Considering both the rotation and position, we get

$$\begin{bmatrix} a_x^G \\ a_y^G \\ a_z^G \end{bmatrix} = \underbrace{\begin{bmatrix} e_x^G e_x^B & e_x^G e_y^B & e_x^G e_z^B \\ e_y^G e_x^B & e_y^G e_y^B & e_y^G e_z^B \\ e_z^G e_x^B & e_z^G e_y^B & e_z^G e_z^B \end{bmatrix}} \begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix} + \begin{bmatrix} x_B^G \\ y_B^G \\ z_B^G \end{bmatrix}$$

Position in $\mathbf{C}_G$ **Rotation Matrix $\mathbf{R}_B^G$** **Position in $\mathbf{C}_B$** **$\mathbf{t}_B^G$,translation between $\mathbf{C}_G$ and $\mathbf{C}_B$**



The $\mathbf{R}_B^G$ represent the orientation and the $\mathbf{t}_B^G$ represents the position of the flight mechanic in the ECEF coordinate system!

Question

Given

- The position of a particle **a** in the ECEF coordinate as (a_x^G, a_y^G, a_z^G)
- The transformation between between $\mathbf{C}_B$ and $\mathbf{C}_G$ as rotation matrix $\mathbf{R}_B^G$ and translation vector $\mathbf{t}_B^G$

Question:

- Calculate the coordinate of particle **a** in the coordinate $\mathbf{C}_B$.

Solution:

Since we have $\begin{bmatrix} a_x^G \\ a_y^G \\ a_z^G \end{bmatrix} = \mathbf{R}_B^G \begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix} + \mathbf{t}_B^G,$

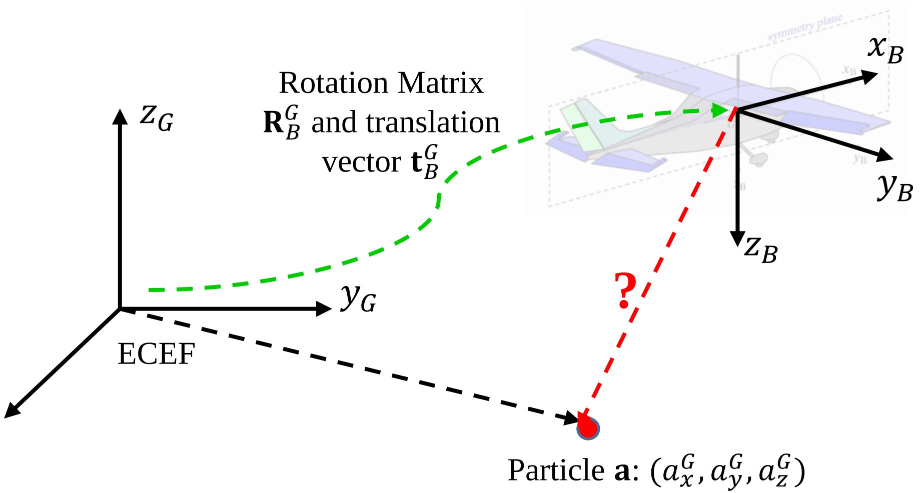
Therefore, we have

$$\begin{bmatrix} a_x^G \\ a_y^G \\ a_z^G \end{bmatrix} - \mathbf{t}_B^G = \mathbf{R}_B^G \begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix},$$

Solution:

Multiple ${\mathbf{R}_B^G}^{-1}$ on both sides, we get

$$\begin{bmatrix} a_x^B \\ a_y^B \\ a_z^B \end{bmatrix} = {\mathbf{R}_B^G}^{-1} \left(\begin{bmatrix} a_x^G \\ a_y^G \\ a_z^G \end{bmatrix} - \mathbf{t}_B^G \right),$$



Q&A

Thank you for your attention ☺

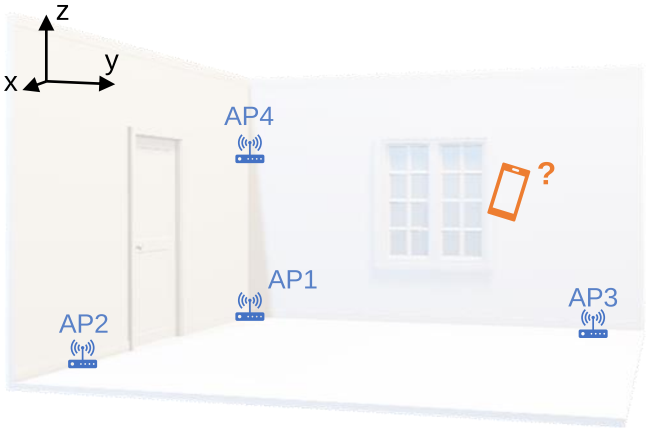
Q&A

Dr. Weisong Wen, Assistant Professor, PQ408

If you have any questions or inquiries, please feel free to contact me.

Email: welson.wen@polyu.edu.hk

Example: TDOA (Time Difference of Arrival) Model



Question:

You are given the coordinates of four access points (APs), and the measured distances from these APs to a user's device.

Considering an existence of error b_i caused by clock errors, it is assumed that the errors are constant.

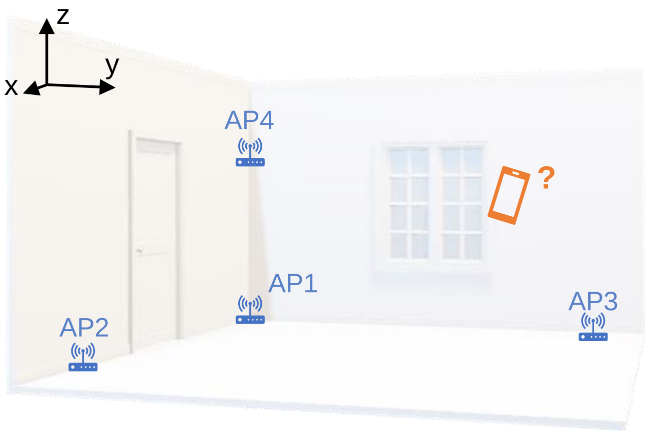
	AP Coordinate	The measured distances from these APs to a user's device
AP1	(0,0,0)	11
AP2	(6,0,0)	9
AP3	(0,8,0)	7
AP4	(0,0,7.5)	13.5

Challenges

This problem is inherently nonlinear due to the square distance calculation.

Estimating x, y, z, and b_i simultaneously requires solving a system of nonlinear equations.

Example: TDOA (Time Difference of Arrival) Model



Question:

You are given the coordinates of four access points (APs), and the measured distances from these APs to a user's device.

Considering an existence of error b_i caused by multipath effects and clock errors, it is assumed that the errors are constant.

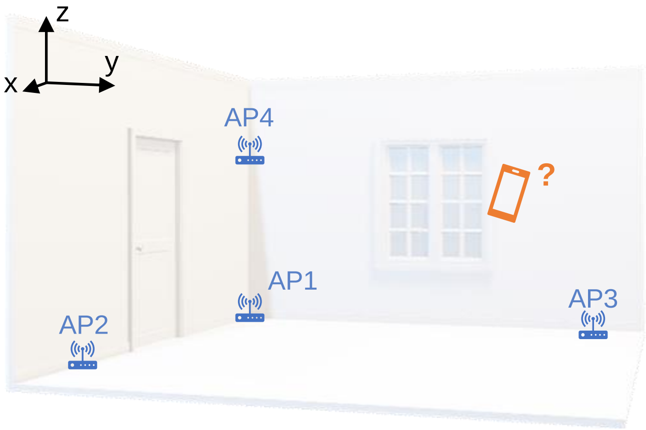
	AP Coordinate	The measured distances from these APs to a user's device
AP1	(0,0,0)	11
AP2	(6,0,0)	9
AP3	(0,8,0)	7
AP4	(0,0,7.5)	13.5

Challenges

This problem is inherently nonlinear due to the presence of the error term b_i affecting the square of the distances in a non-additive manner.

Estimating x, y, z, and b_i simultaneously requires solving a system of nonlinear equations.

Example: TDOA (Time Difference of Arrival) Model



Question:

You are given the coordinates of four access points (APs), and the measured distances from these APs to a user's device.

Considering an existence of error b_i caused by multipath effects and clock errors, it is assumed that the errors are constant.

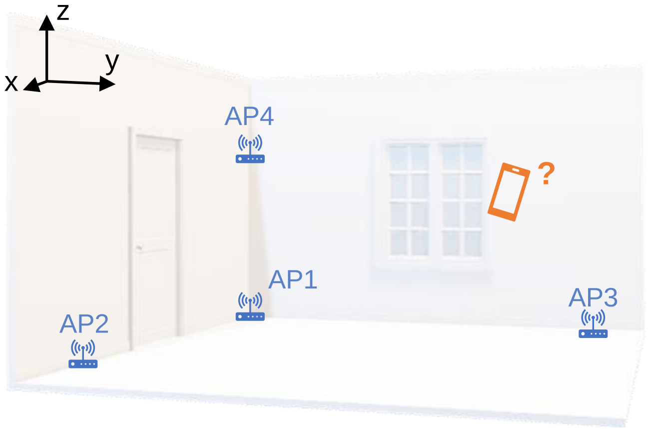
	AP Coordinate	The measured distances from these APs to a user's device
AP1	(0,0,0)	11
AP2	(6,0,0)	9
AP3	(0,8,0)	7
AP4	(0,0,7.5)	13.5

Solution Approach

To solve this problem, a nonlinear least squares optimization method can be applied. This involves:

- **Defining a Residual Function**: This function calculates the discrepancy between the predicted squared distances (based on the estimated position and error) and the squared adjusted measurements (considering the error).

Example: TDOA (Time Difference of Arrival) Model



Question:

You are given the coordinates of four access points (APs), and the measured distances from these APs to a user's device.

Considering an existence of error b_i caused by multipath effects and clock errors, it is assumed that the errors are constant.

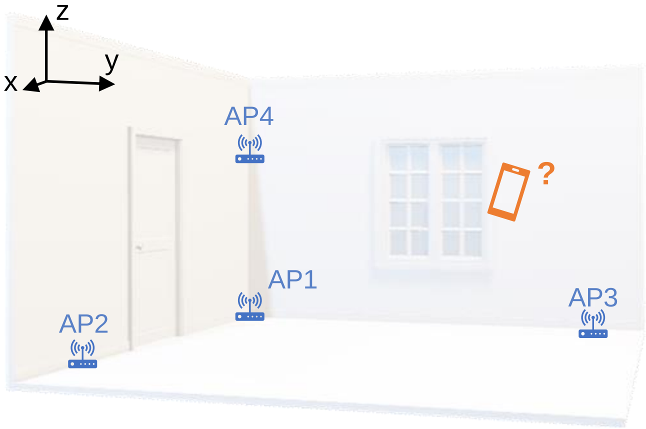
	AP Coordinate	The measured distances from these APs to a user's device
AP1	(0,0,0)	11
AP2	(6,0,0)	9
AP3	(0,8,0)	7
AP4	(0,0,7.5)	13.5

Solution Approach

To solve this problem, a nonlinear least squares optimization method can be applied. This involves:

- **Optimization Method**: Using a method like least_squares from the scipy.optimize module, which can handle the minimization of the sum of the squares of the residuals between the predicted and observed data.

Example: TDOA (Time Difference of Arrival) Model



Question:

You are given the coordinates of four access points (APs), and the measured distances from these APs to a user's device.

Considering an existence of error b_i caused by multipath effects and clock errors, it is assumed that the errors are constant.

	AP Coordinate	The measured distances from these APs to a user's device
AP1	(0,0,0)	11
AP2	(6,0,0)	9
AP3	(0,8,0)	7
AP4	(0,0,7.5)	13.5

The equation can be rewritten as:

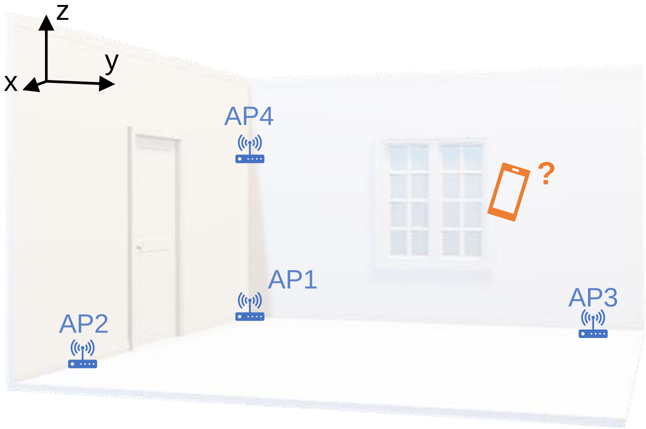
$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = (d_i - b_i)^2$$

This equation holds for every access point (AP).
We can expand it as:

$$x^2 - 2xx_i + x_i^2 + y^2 - 2yy_i + y_i^2 + z^2 - 2zz_i + z_i^2 = d_i^2 - 2d_ib_i + {b_i}^2$$

$$\mathbf{Ax} = \mathbf{b}$$

Example: TDOA (Time Difference of Arrival) Model



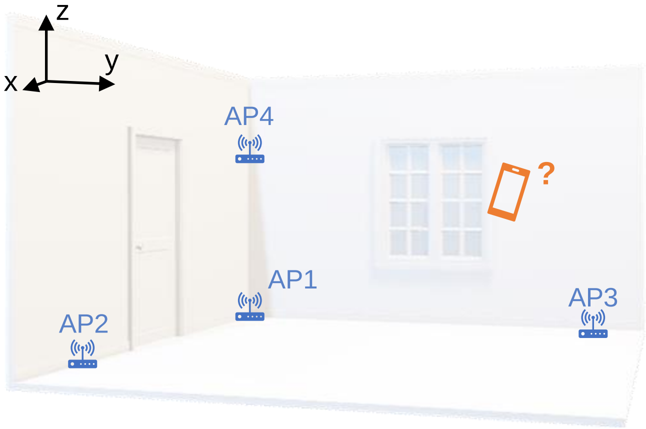
https://jupyter.org/try-jupyter/lab/

Python Solution:

```
import numpy as np from scipy.optimize
import least_squares
# Define the coordinates of the Access Points
(APs).
ap_coords = np.array([
[0, 0, 0], # AP1 coordinates
[6, 0, 0], # AP2 coordinates
[0, 8, 0], # AP3 coordinates
[0, 0, -7.5] # AP4 coordinates, possibly
located below the level of the others ])

# The actual distances from the user to each
AP.
dd = np.array([11, 9, 7, 13.5])
```

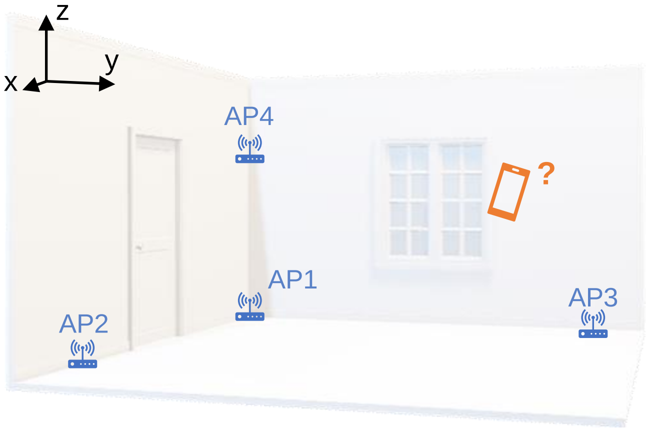
Example: TDOA (Time Difference of Arrival) Model



Python Solution:

```
# Define the residual function that calculates the
difference between
# the predicted distances (from the guessed coordinates)
and the measured distances.
def residuals(params): x, y, z, err = params
# x, y, z are the user's coordinates and err is an
additional error term
 res = [] # List to hold the residuals for each AP
 for i in range(len(ap_coords)):
  xi, yi, zi = ap_coords[i] # Coordinates of the i-th AP
  # Calculate the squared predicted distance from the
guessed user position to the i-th AP
  predicted_d2 = (x - xi)**2 + (y - yi)**2 + (z - zi)**2
  # Adjust the measured distance by the error term and
then square it
  measured_d2 = (dd[i] - err)**2
  # Calculate the residual (difference) and append to the
list
  res.append(predicted_d2 - measured_d2)
 return res
```

Example: TDOA (Time Difference of Arrival) Model



Python Solution:

```
# Initial guess for the user's position and the
error term.
initial_guess = [0, 0, 0, 1]

# Solving the least squares problem using the
residuals function and the initial guess.
result = least_squares(residuals, initial_guess)

# Extract the optimized values from the result.
x, y, z, err = result.x

print(f"User coordinates: (x: {x:.2f}, y: {y:.2f},
z: {z:.2f})")
print(f"Error: {err:.2f}")
```