AUSROS 2024

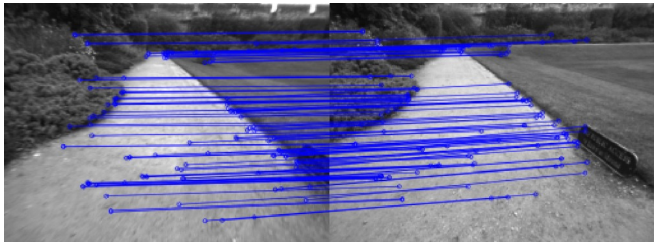
Session A2: Localisation and Mapping

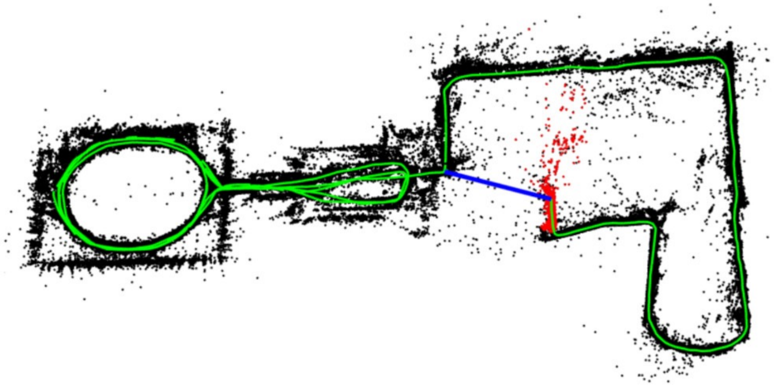
Dr. Mitch Bryson

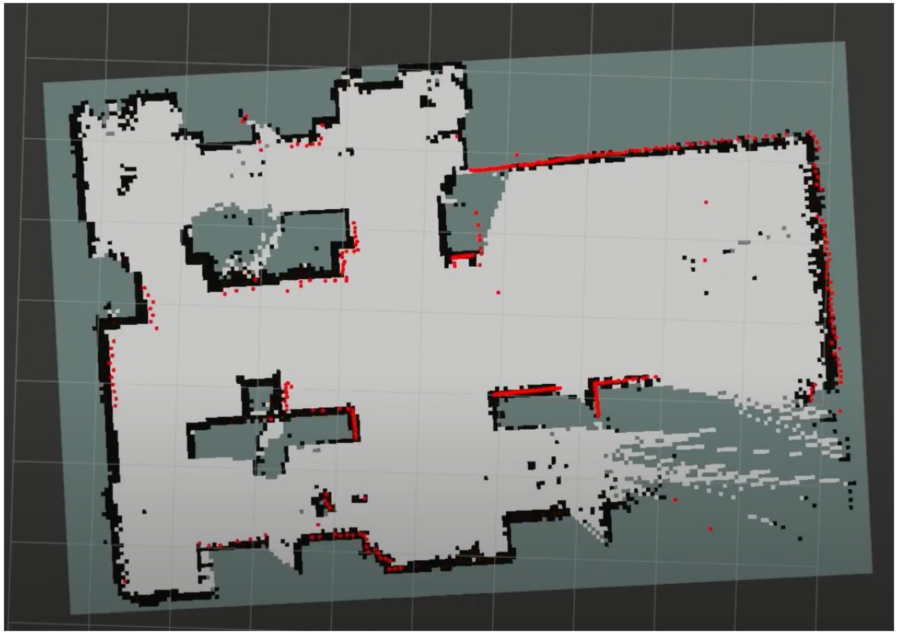
Australian Centre for Robotics
University of Sydney

Introduction

- **Localisation:** the art/science of determining one's position and orientation (pose) with respect to a known frame of reference
- **Mapping:** the art/science of determining the spatial configuration of the surrounding environment with respect to a known frame of reference

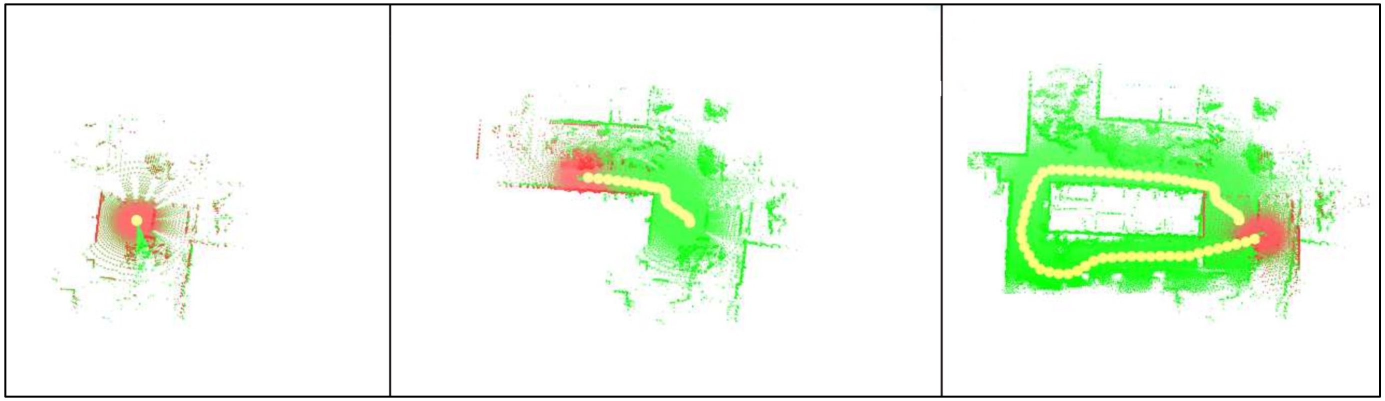




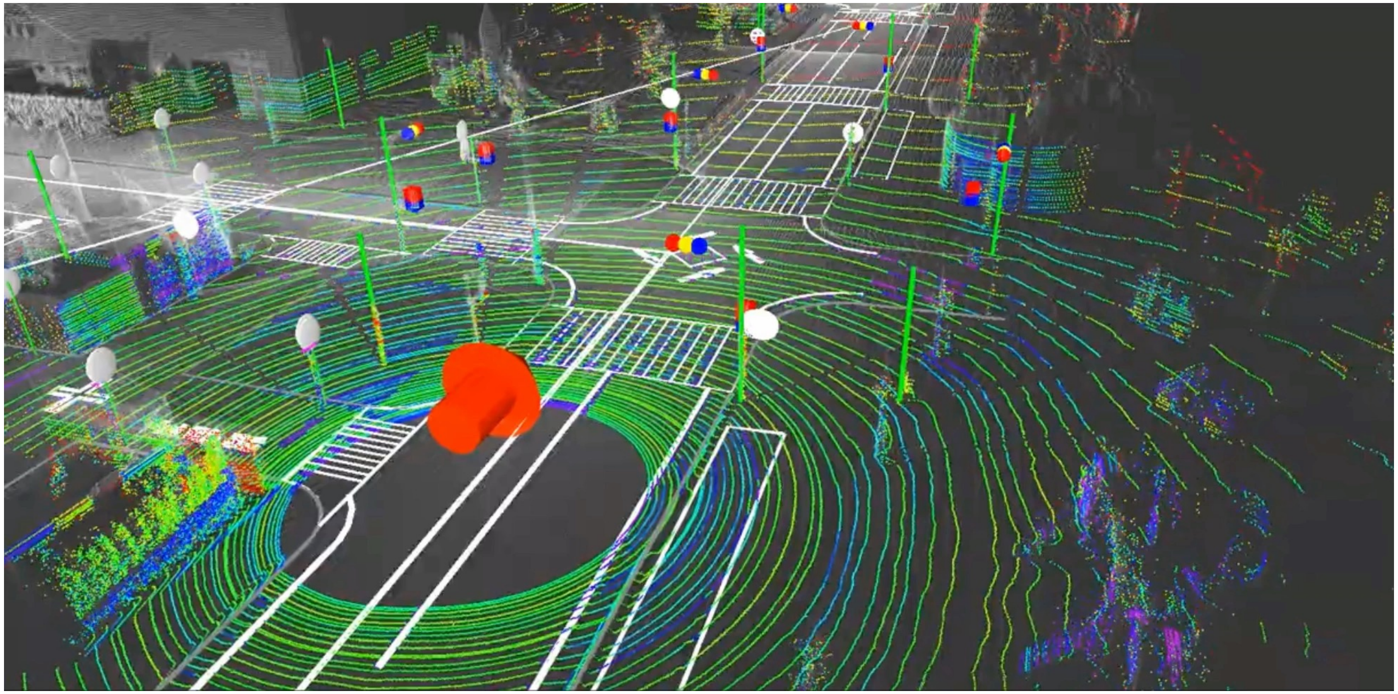


- Why localise and map?
  - Robot control, stability
  - Obstacle avoidance
  - Navigation from one location to another
  - Spatial information itself as an end goal for robotic exploration

Examples: LiDAR-based localization and mapping



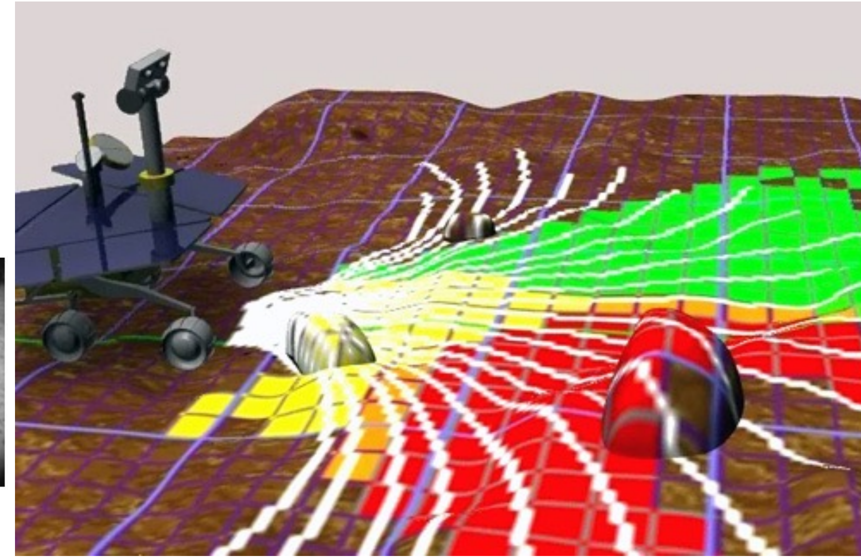
Bormann et. al., Globally Consistent 3D Mapping with Scan Matching, 2007



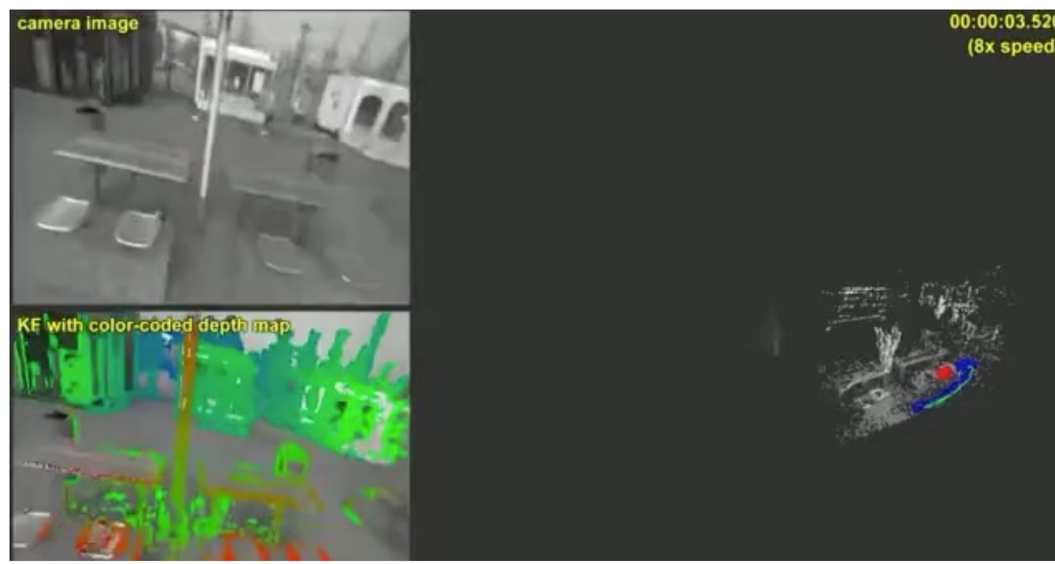
Autoware.AI LiDAR localization using NDT scan matching

Examples: Vision-based Localisation and Navigation

S. Goldberg, M. Maimone and L. Matthies, “Stereo Vision and Rover Navigation Software for Planetary Exploration”, IEEE Aerospace Conference, 2012



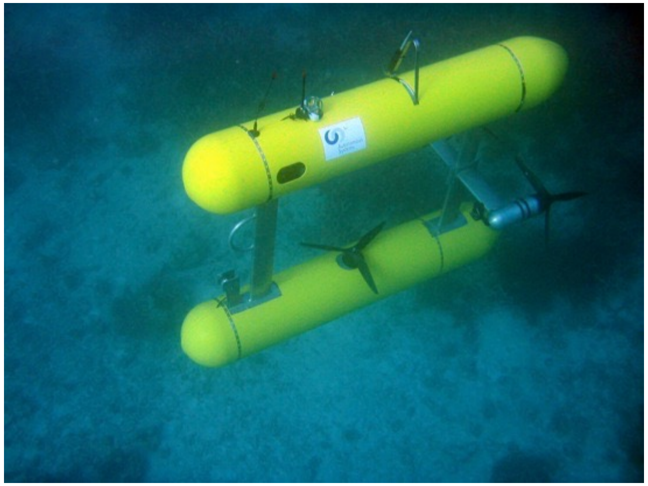
G. Conte and P. Doherty, "An Integrated UAV Navigation System Based on Aerial Image Matching", IEEE Aerospace Conference 2007.

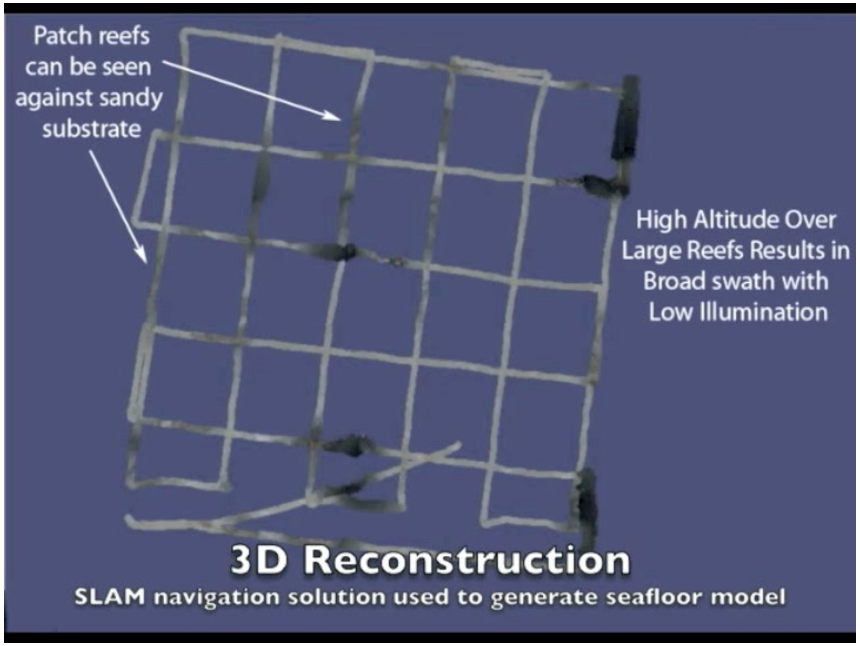


Jakob Engel and Thomas Schops and Daniel Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM", ECCV 2014

Examples: 3D mapping using vision/structure-from-motion

Furukawa et. al., "Towards Internet-scale Multi-view Stereo", CVPR, 2010



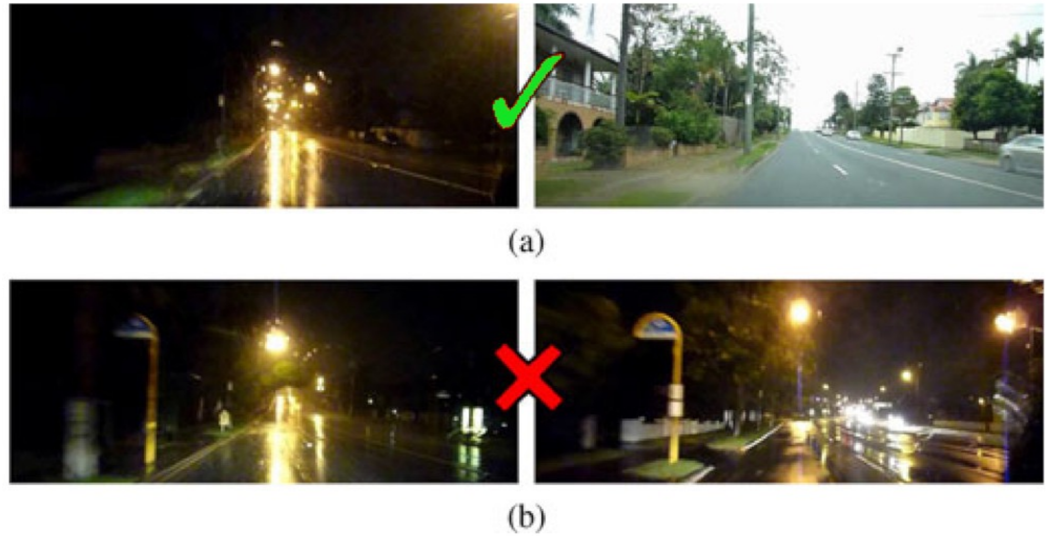


Johnson-Roberson et. al., "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys", JFR, 2010

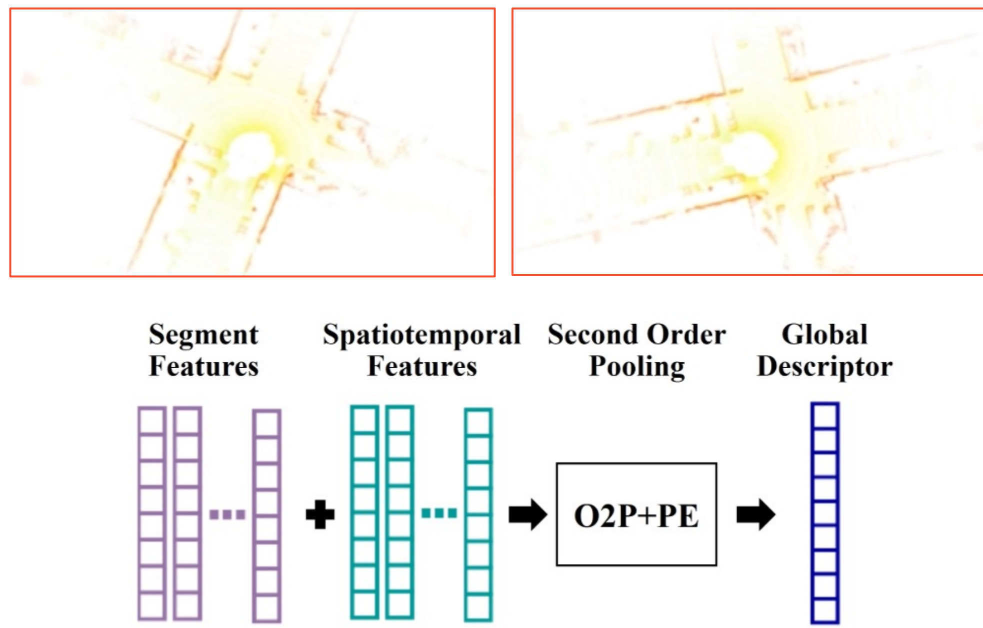
Place Recognition

- Recognising a place the robot has previously visited using sensor data
  - under potentially large domain shifts (perspective shift, day vs. night, seasonal etc.)
  - also reject false positives containing repetitious scene elements
- Typical approaches segment the environment and assign descriptors (handcrafted or learnt), perform matching in feature and/or location space

Lowry et. al., Visual Place Recognition: A Survey”, IEEE TRO, 2016



(a) (b)

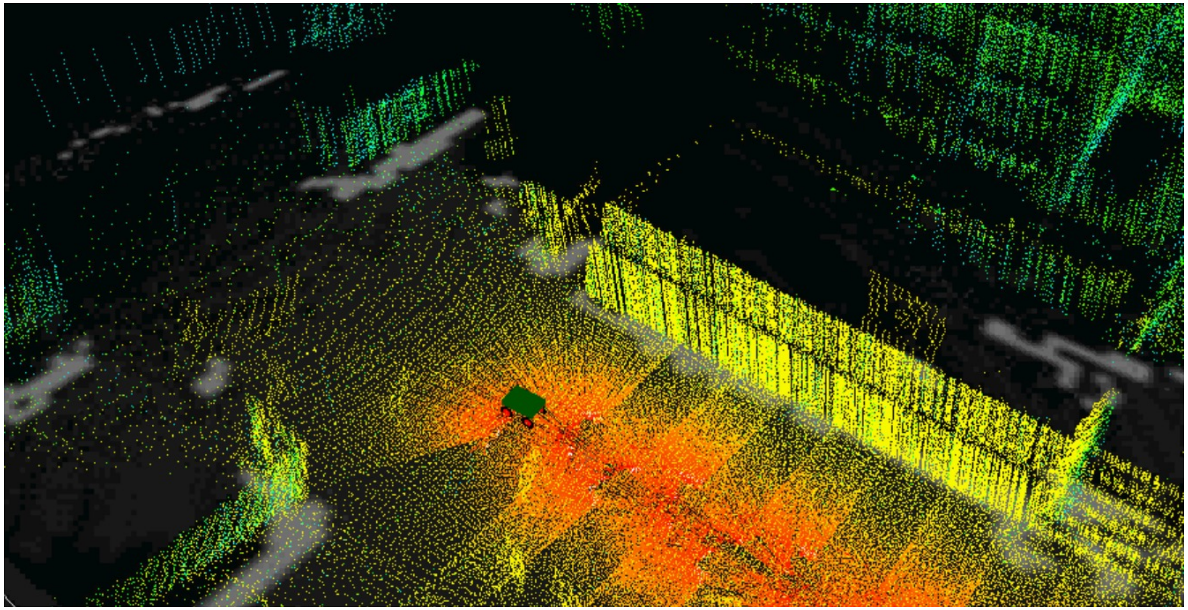


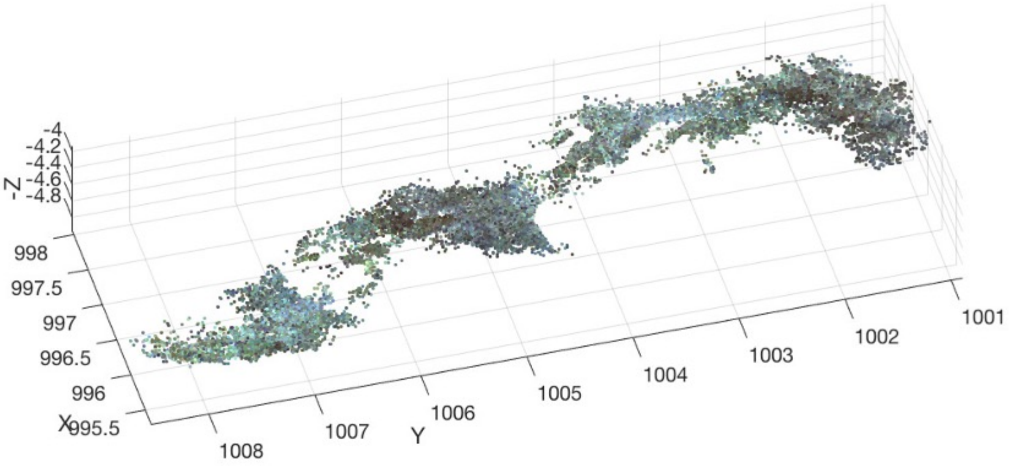
Vidanapathirana et.al., “Locus: LiDAR-based Place Recognition using Spatiotemporal Higher-Order Pooling.” ICRA 2021

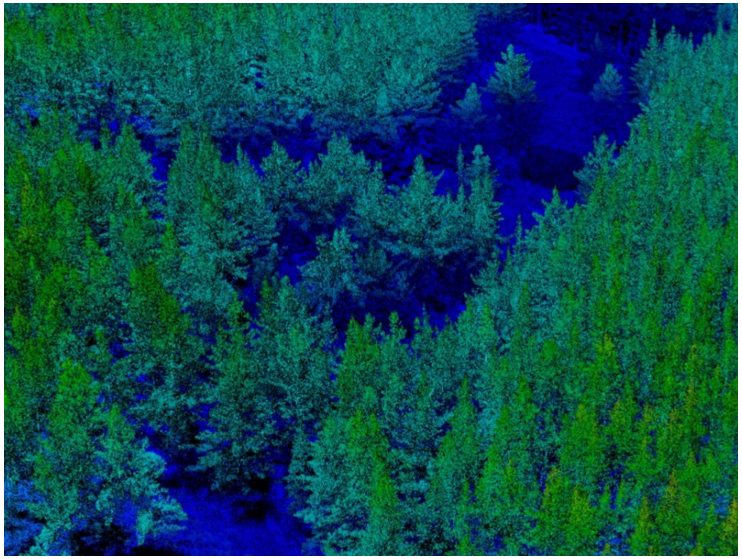
Map Representations

Whitty et. al., Autonomous Navigation using a Real-Time 3D Point Cloud, ACRA 2010

- Different types of map data to suit different purposes
  - Pointclouds
  - Occupancy maps
  - Surfaces: meshes, signed distance fields etc.

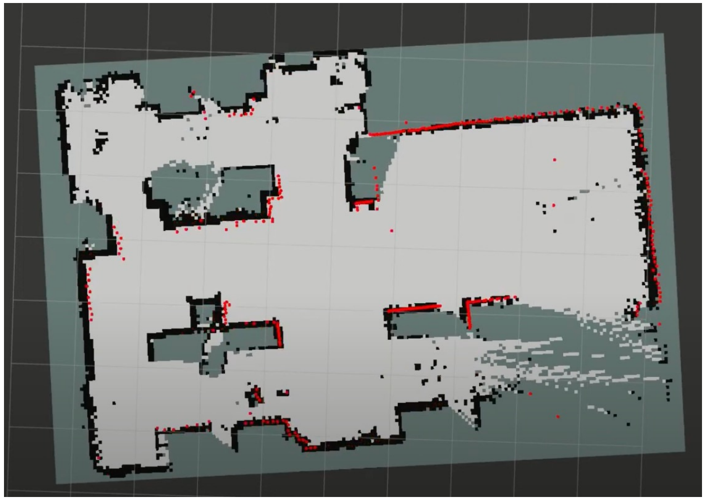




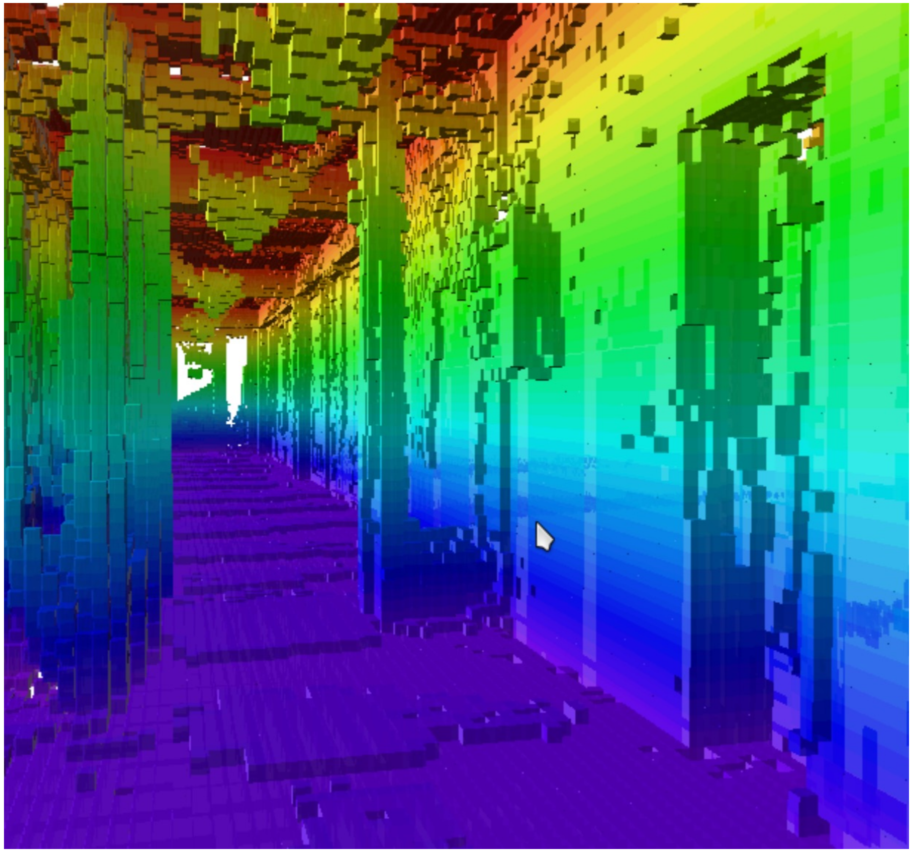


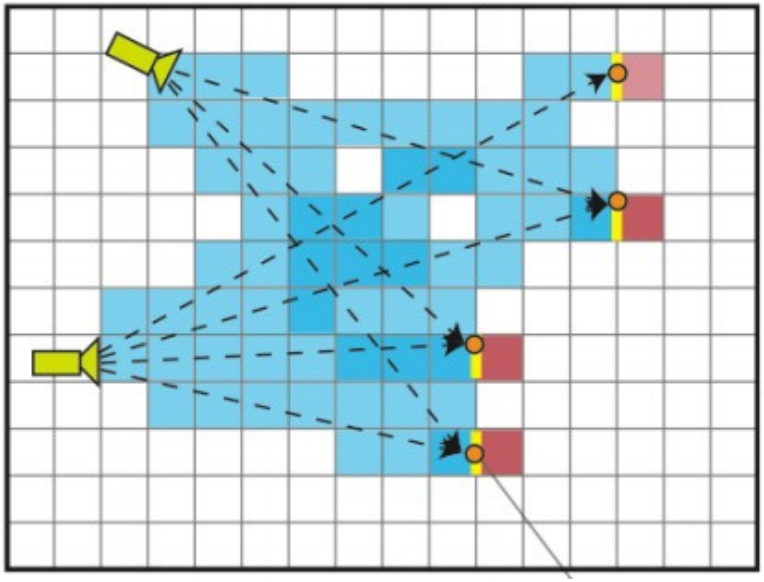
Map Representations

- Different types of map data to suit different purposes
  - Pointclouds
  - Occupancy maps
  - Surfaces: meshes, signed distance fields etc.



Hornug et. al., "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees", 2013



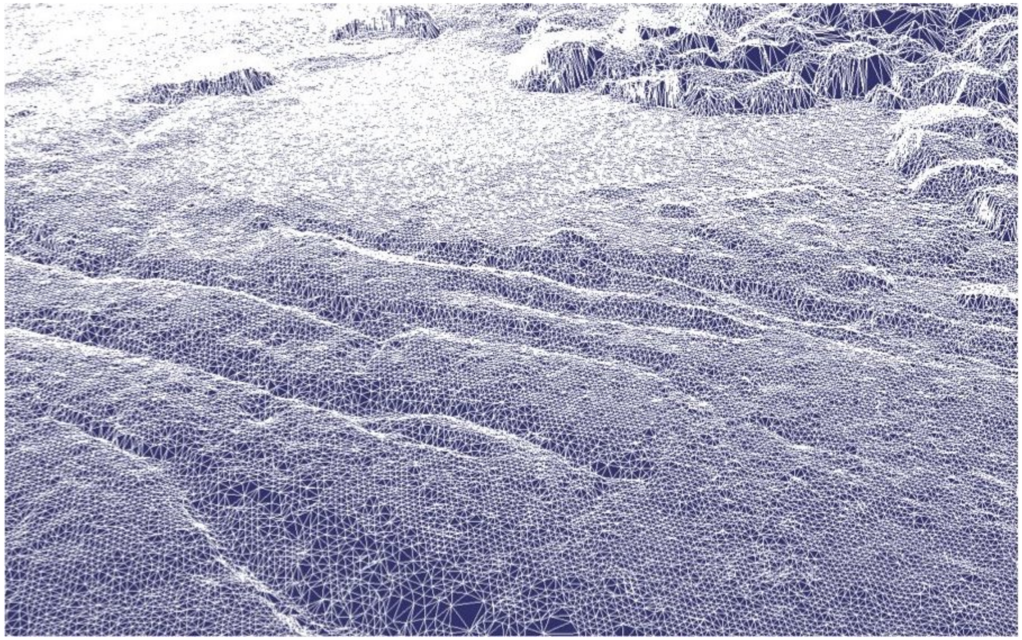


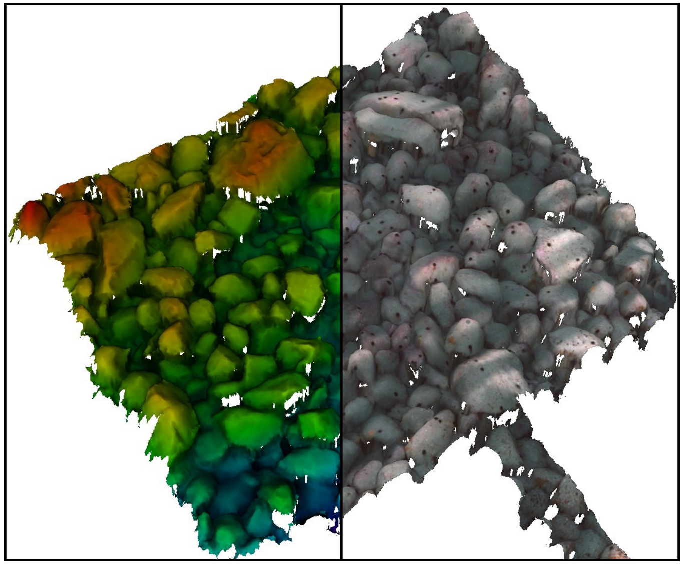
Vogiatzis et. al., “Multi-view stereo via volumetric graph-cuts”,2005.

Map Representations

- Different types of map data to suit different purposes
  - Pointclouds
  - Occupancy maps
  - **Surfaces: meshes, signed distance fields etc.**

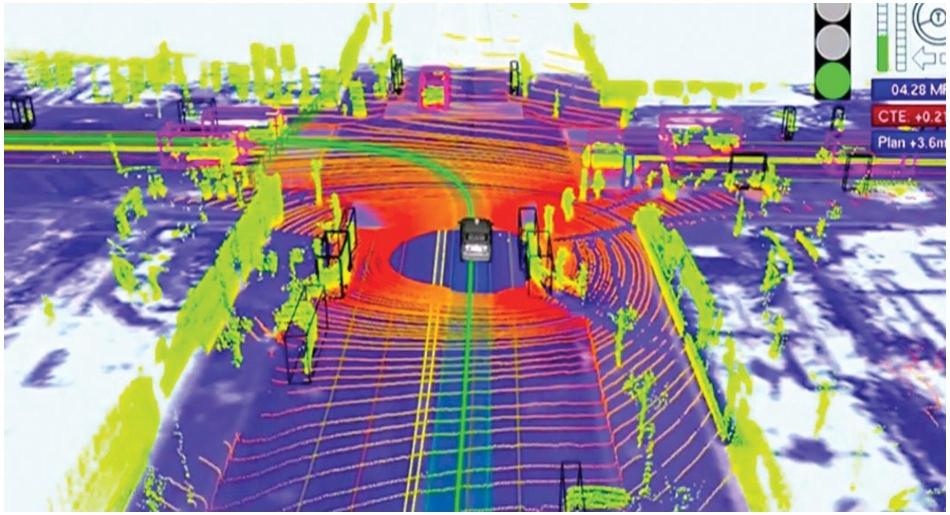
Triangulated surface mesh from point cloud



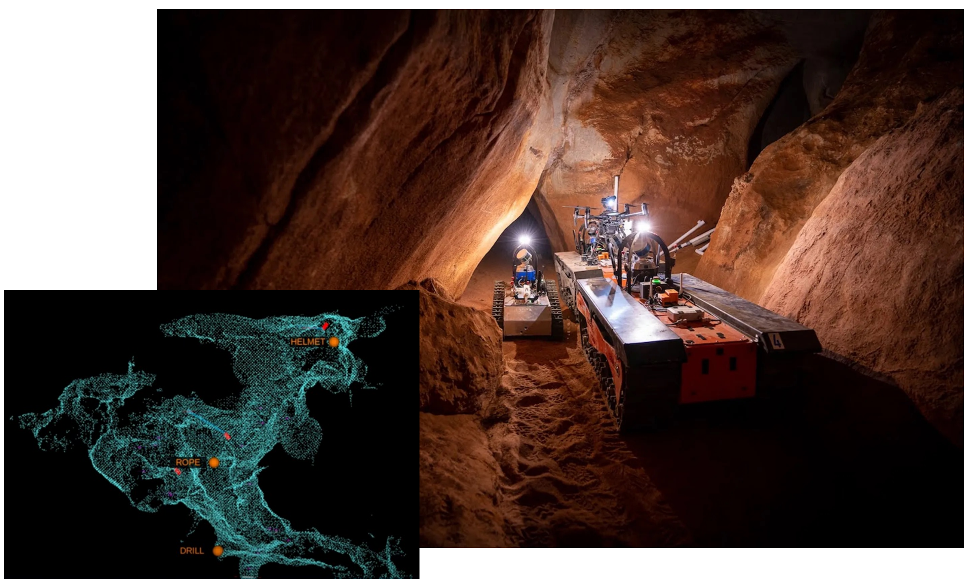


Examples of photo-textured triangulated surface mesh models

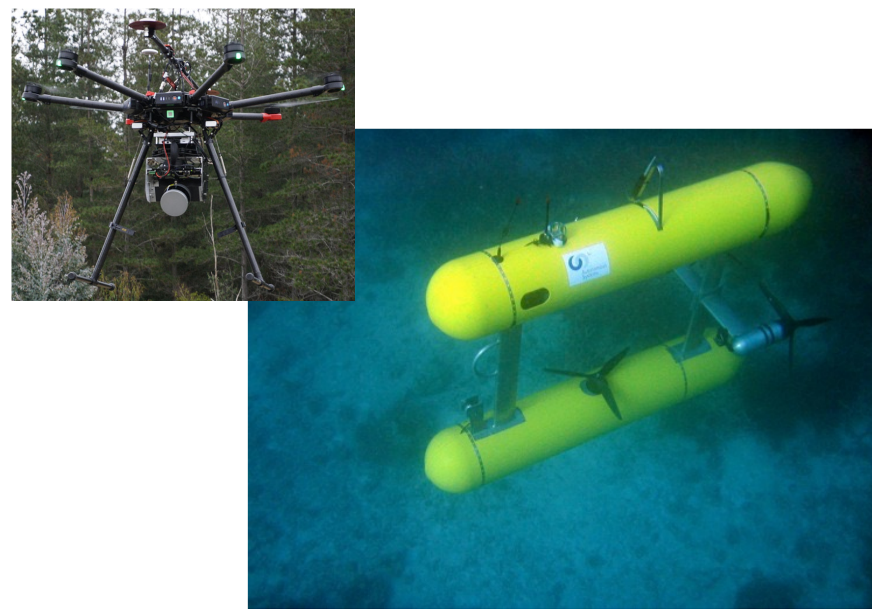
Considerations in different environments/domains



- Availability of external information/signals
- Sensing through medium (e.g. underwater and acoustics)
- Dynamic nature of the environment (urban etc.)
- Platform dynamics, maneuverability, 2D vs. 3D



DARPA SubT Challenge Cave Circuit event, CSIRO Data61 team



Terraluma LiDAR drone and AUV Sirius, ACFR

Overview of the lecture:

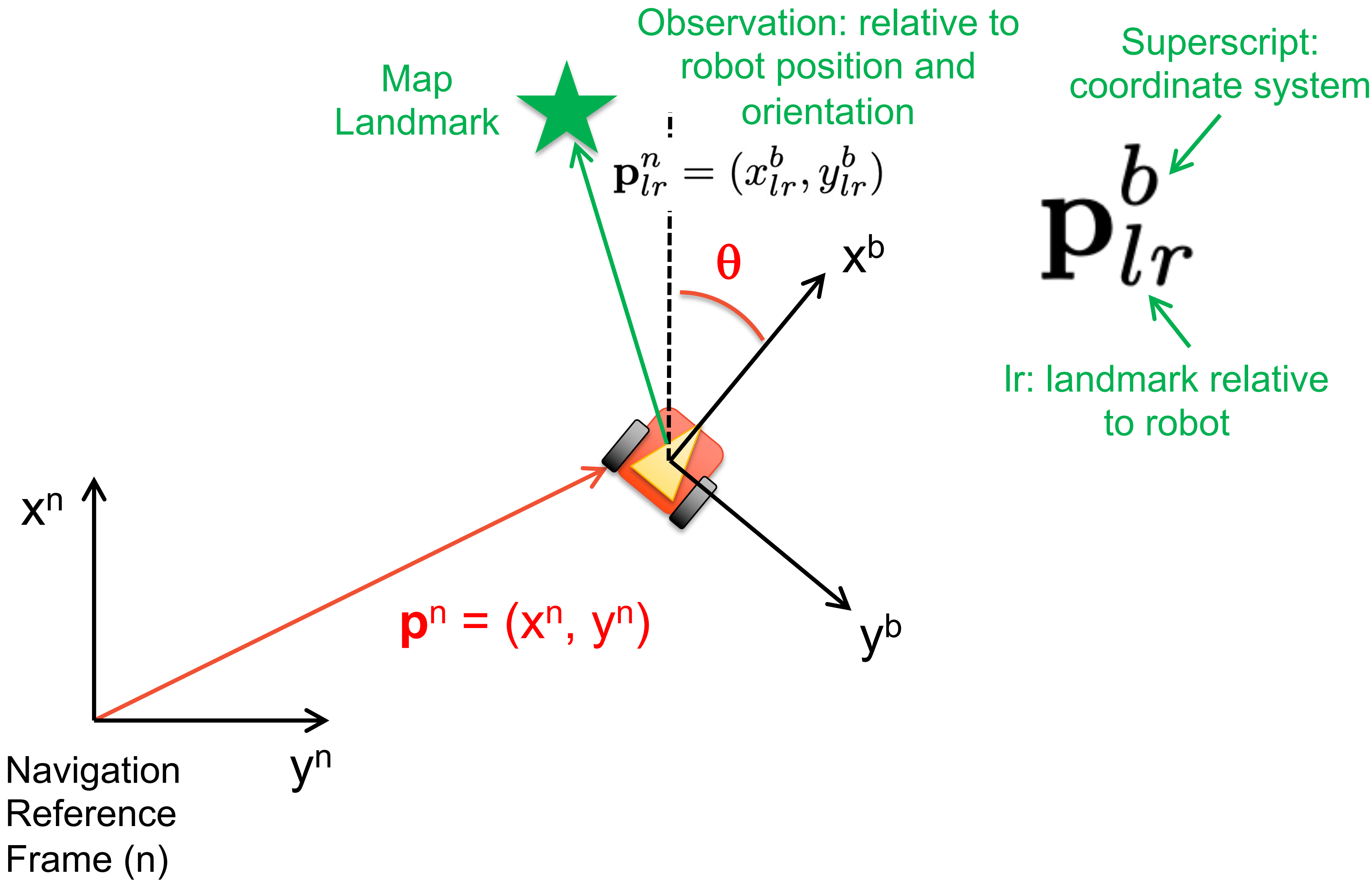
- Introduction to Robot Localisation and Mapping

- Robot Localisation and Mapping:
  - Basics: Coordinate Systems and Frames of Reference
  - Localisation and Mapping as an estimation problem

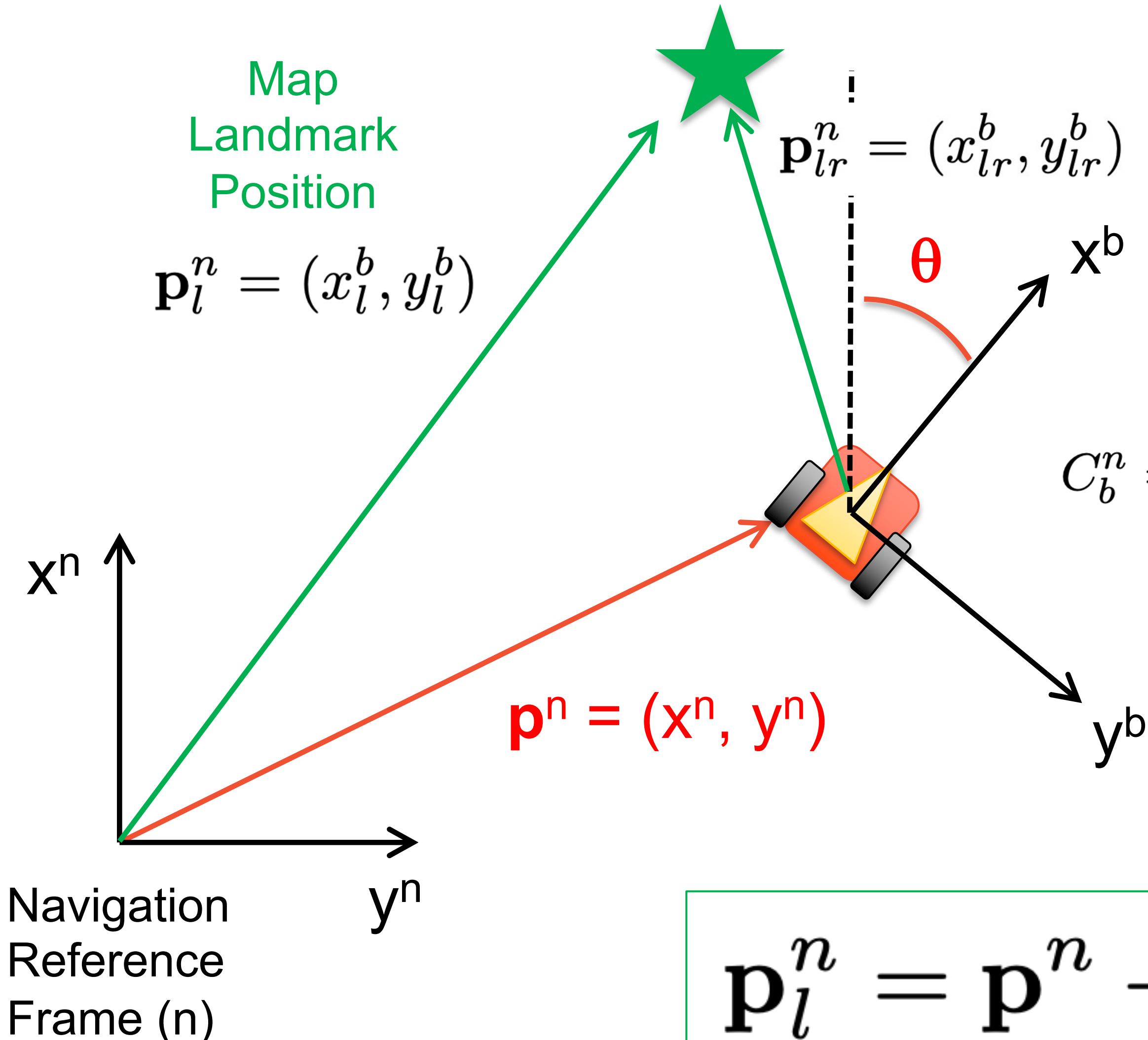
- Examples:
  - Beacon-based localisation
  - Aided-inertial navigation
  - Simultaneous Localisation and Mapping (SLAM)

- Conclusions and Future outlook

Basics: Coordinate Systems and Frames of Reference



Basics: Coordinate Systems and Frames of Reference

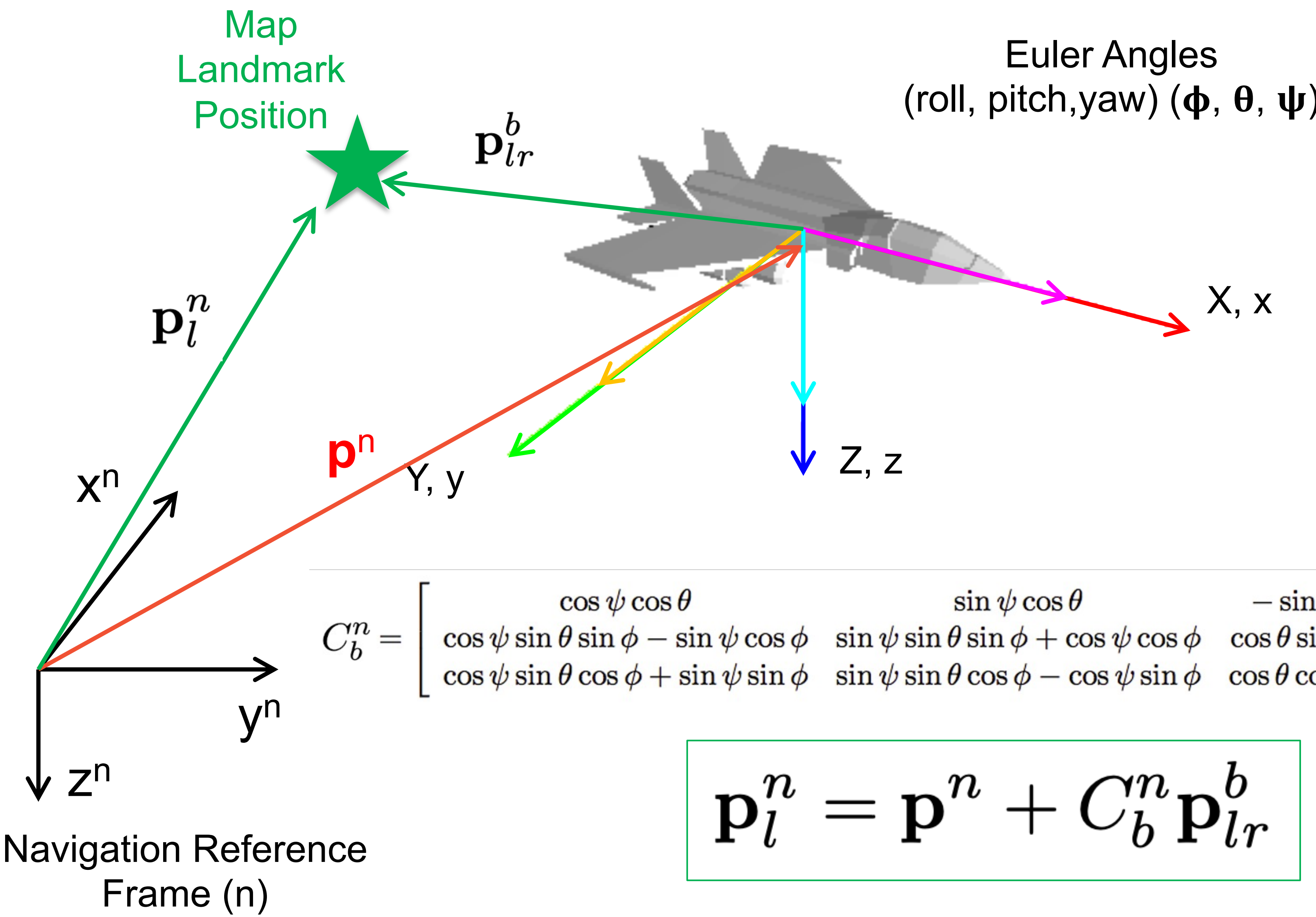


Direction Cosine
(Rotation) Matrix
(2D case)

$$C_b^n = R_b^n = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$\boxed{\mathbf{p}_l^n = \mathbf{p}^n + C_b^n \mathbf{p}_{lr}^b}$$

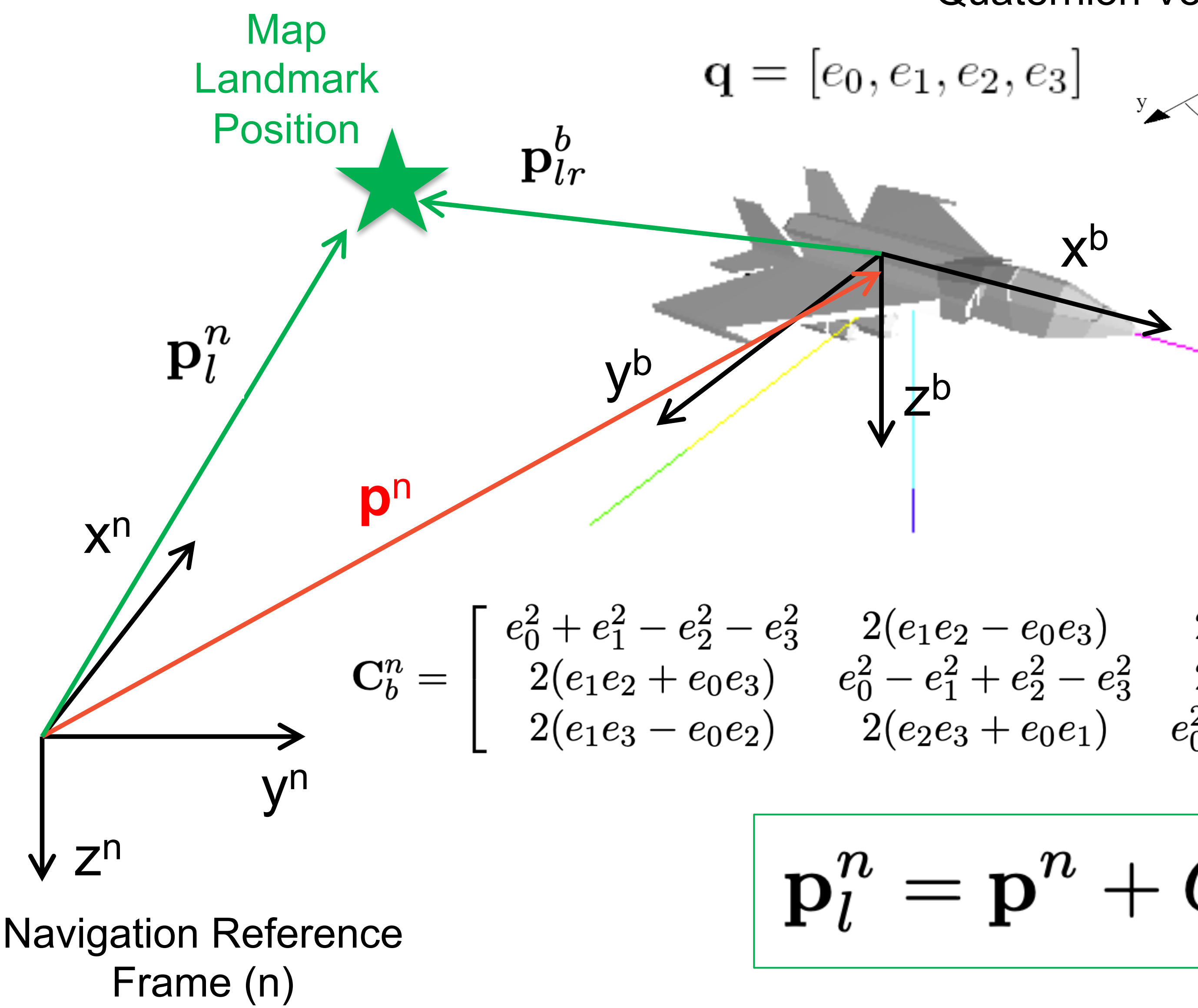
Basics: Coordinate Systems and Frames of Reference



$$C_b^n = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

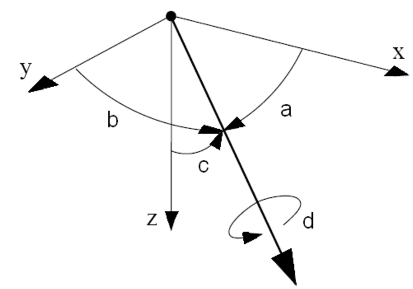
$$\boxed{\mathbf{p}_l^n = \mathbf{p}^n + C_b^n \mathbf{p}_{lr}^b}$$

Basics: Coordinate Systems and Frames of Reference



Quaternion Vector (q)

$$\mathbf{q} = [e_0, e_1, e_2, e_3]$$



$$\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix} \begin{matrix} = \cos\left(\frac{\mathrm{d}}{2}\right) \\ = \cos \mathrm{a} \sin\left(\frac{\mathrm{d}}{2}\right) \\ = \cos \mathrm{b} \sin\left(\frac{\mathrm{d}}{2}\right) \\ = \cos \mathrm{c} \sin\left(\frac{\mathrm{d}}{2}\right) \end{matrix}$$

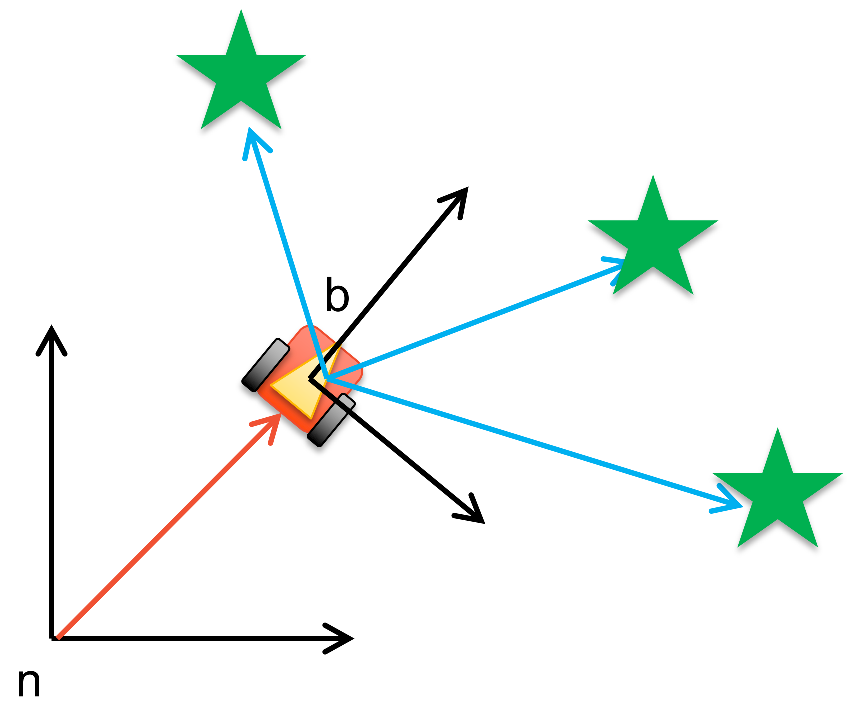
$$\mathbf{C}_b^n = \begin{bmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2(e_1e_2 - e_0e_3) & 2(e_1e_3 + e_0e_2) \\ 2(e_1e_2 + e_0e_3) & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2(e_2e_3 - e_0e_1) \\ 2(e_1e_3 - e_0e_2) & 2(e_2e_3 + e_0e_1) & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{bmatrix}$$

$$\boxed{\mathbf{p}_l^n = \mathbf{p}^n + C_b^n \mathbf{p}_{lr}^b}$$

Types of Localisation and Mapping

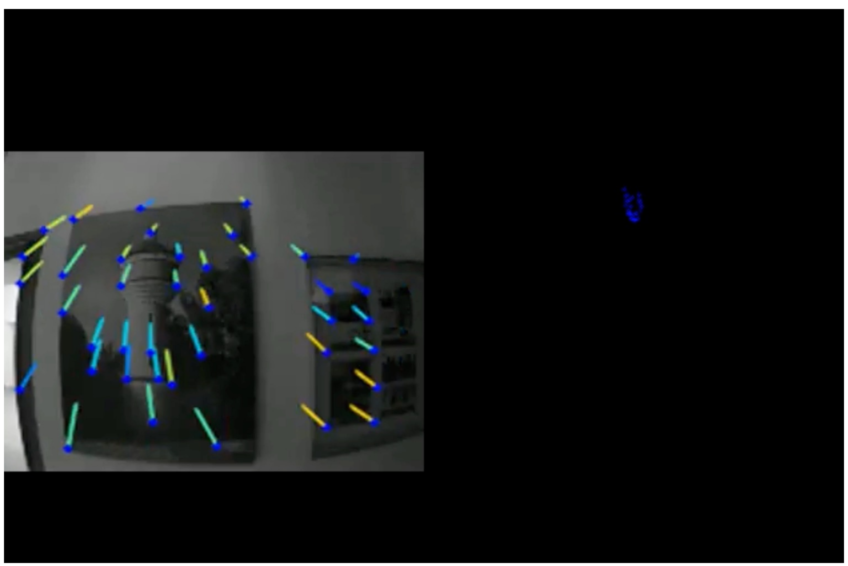
- Fundamentally localization and mapping is an estimation problem involving the fusion of sensor data (and other information sources)

- Robot localization and mapping problems can be categorized based on available information and required accuracy in the estimation of key states

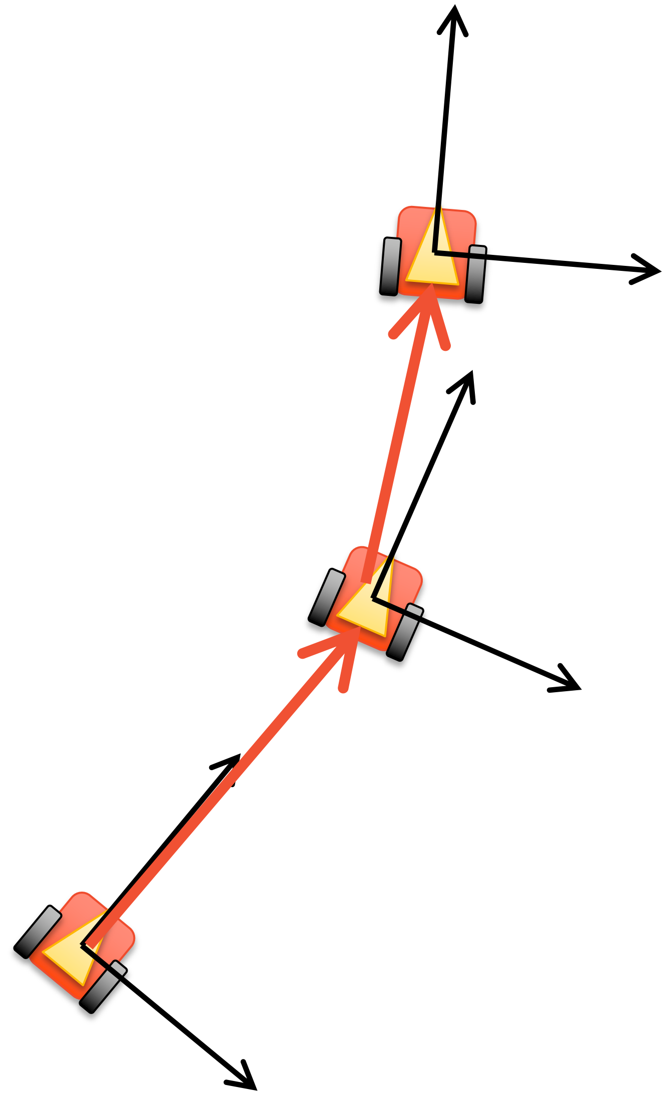


Dead-reckoning and Odometry

- Dead reckoning: determine my current pose from last pose and relative information
- Examples: measurements of velocity from wheel encoders, inertial navigation, LiDAR scan matching, visual odometry

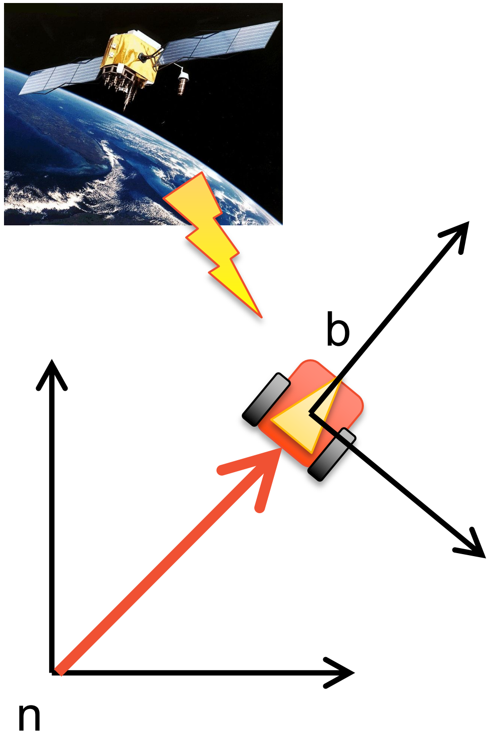


TUM Monocular Visual Odometry Dataset



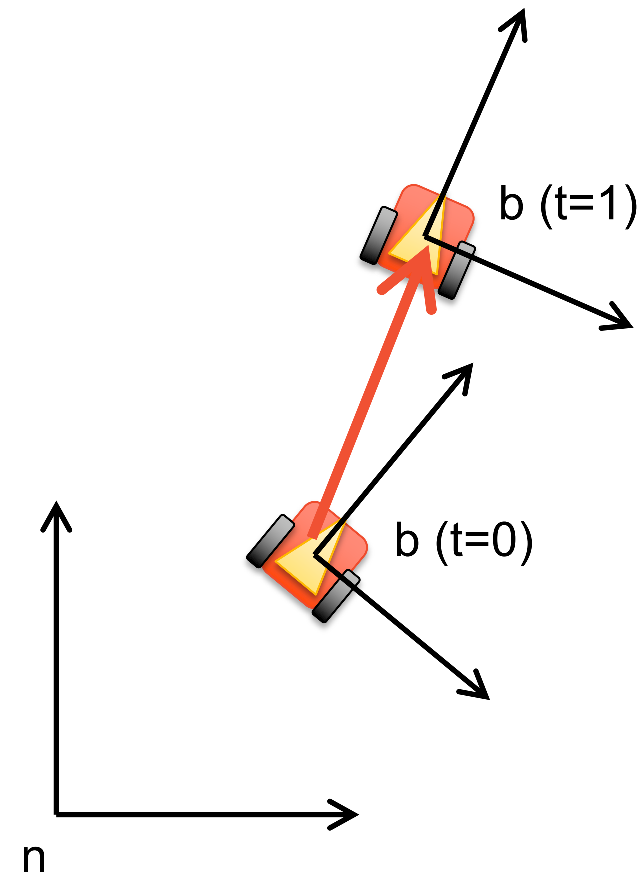
Types of Localisation and Mapping: Information Sources

- **External localization reference:**
  - Location information provided from external source
  - Examples: GNSS, Motion-capture, magnetic heading etc.



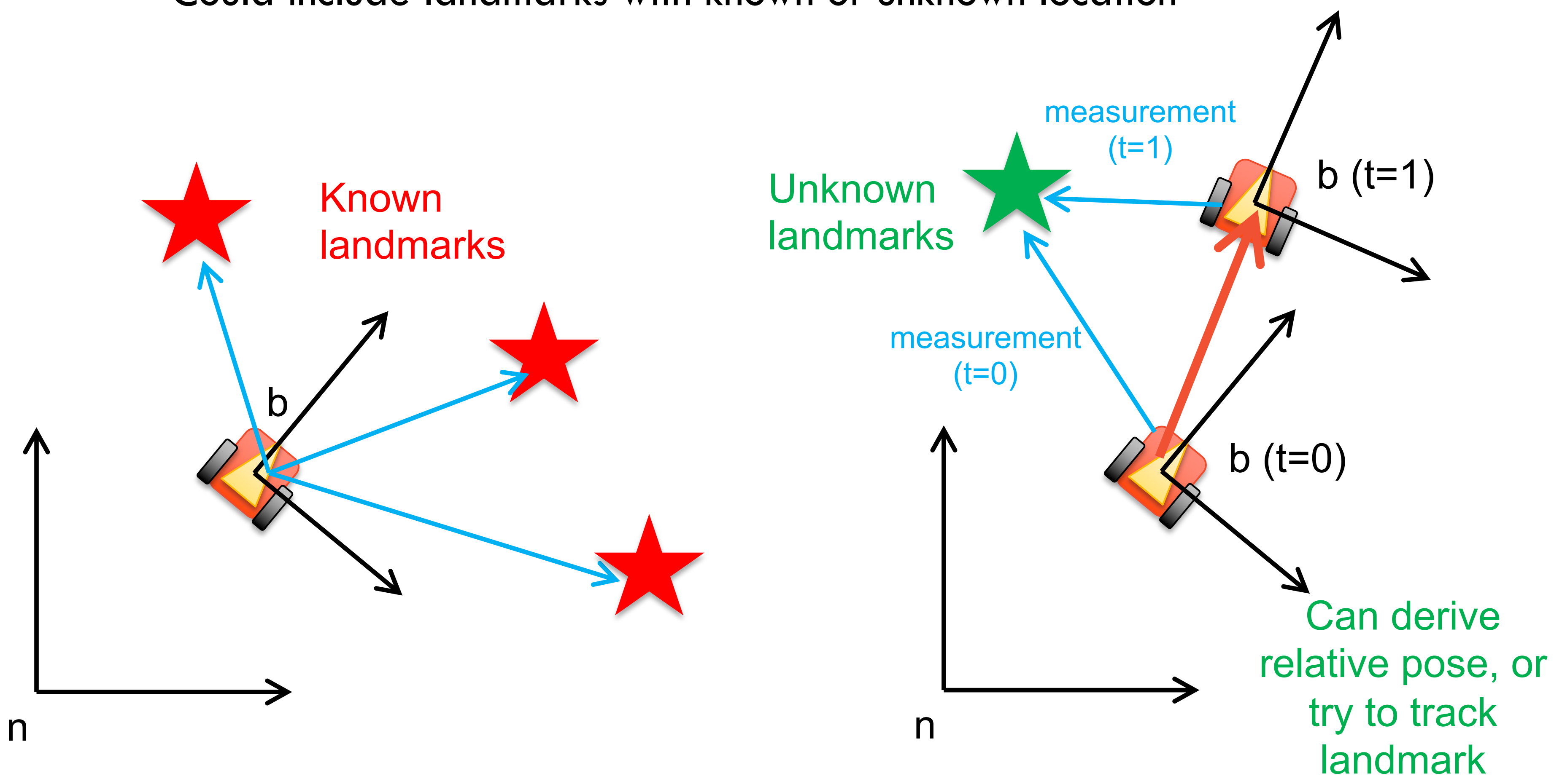
Types of Localisation and Mapping: Information Sources

- **Pose-Relative Measurements:**
  - Location information is provided relative to last pose, or as time-derivatives of pose
  - Examples: odometry, inertial navigation, scan/feature matching between subsequent sensor measurements of external, unknown landmarks



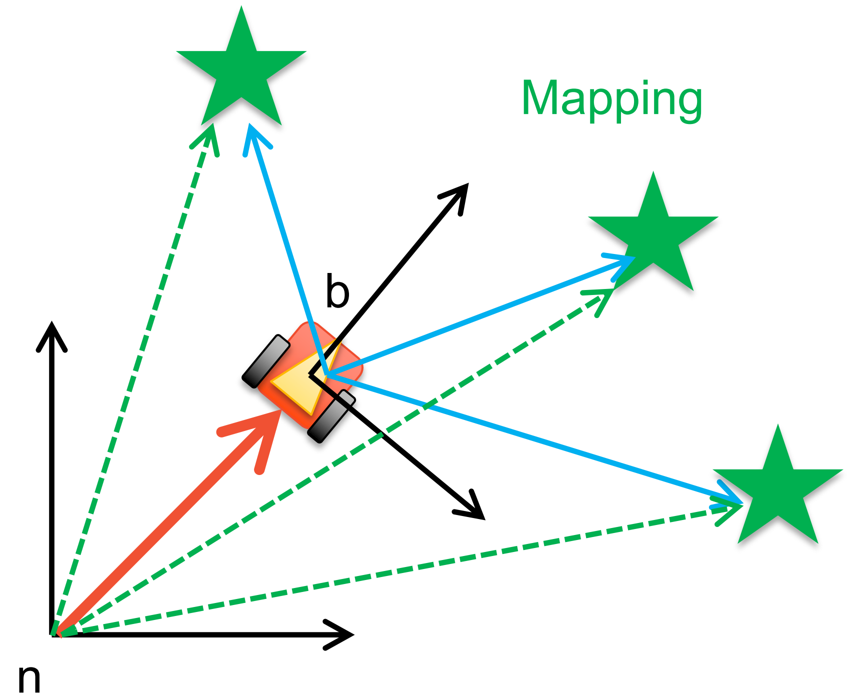
Types of Localisation and Mapping: Information Sources

- **Relative Map/Environment Measurements:**
  - Robot-relative measurements made to environment
  - Examples: scan/feature matching from LiDAR scans, vision features
  - Could include landmarks with known or unknown location



Types of Localisation and Mapping: Estimation Problem

- **Mapping using external localization reference:**
  - If we assume pose of the robot is known, we can compute the locations of map features directly robot-relative sensor measurements
  - Sometimes referred to a "Direct Georeferencing"

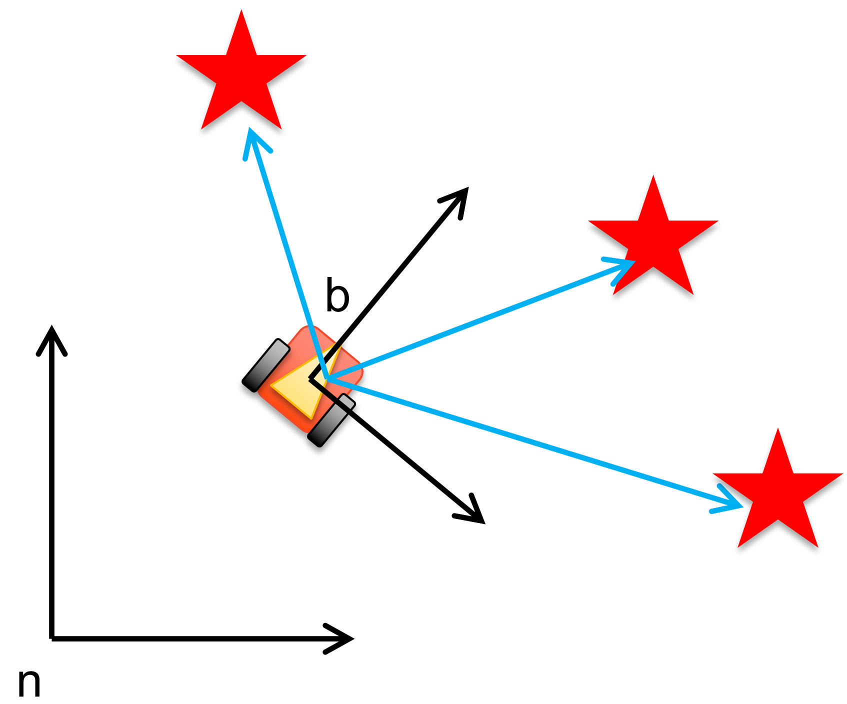


$$\mathbf{p}_l^n = \mathbf{p}^n + C_b^n \mathbf{p}_{lr}^b$$

unknown known measured

Types of Localisation and Mapping: Estimation Problem

- **Beacon-based Localisation:**
  - Robot-relative measurements made to known landmarks
  - Can compute unknown pose that best matches observations and known map
  - Examples: scan/feature matching to known map, multilateration

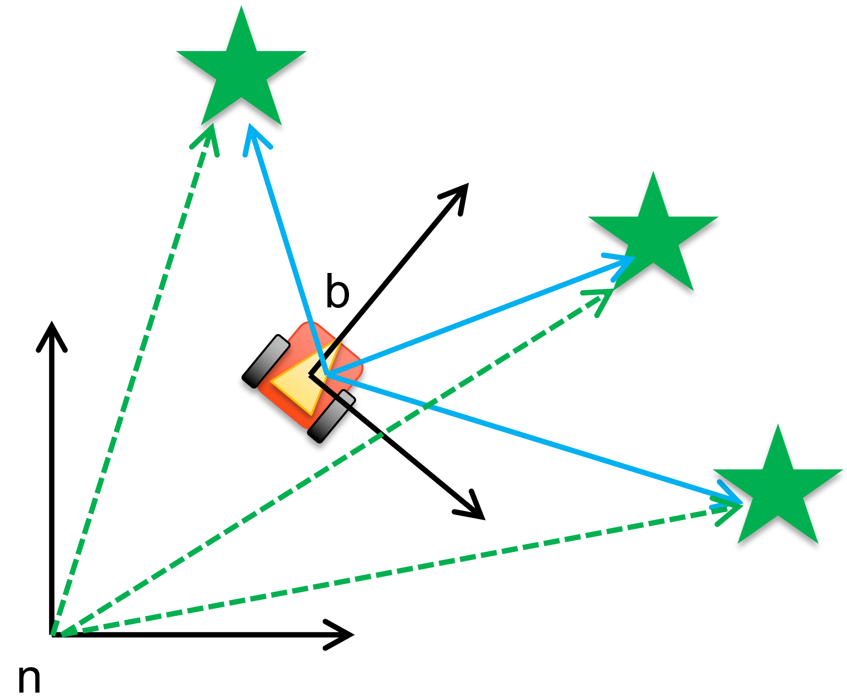


$$\mathbf{p}_l^n = \mathbf{p}^n + C_b^n \mathbf{p}_{lr}^b$$

known unknown measured

Types of Localisation and Mapping: Estimation Problem

- **Simultaneous Localisation and Mapping:**
  - Robot-relative measurements made to unknown landmarks
  - Must estimate both robot and landmark location, assuming landmarks are stationary



$$\mathbf{p}_l^n = \mathbf{p}^n + C_b^n \mathbf{p}_{lr}^b$$

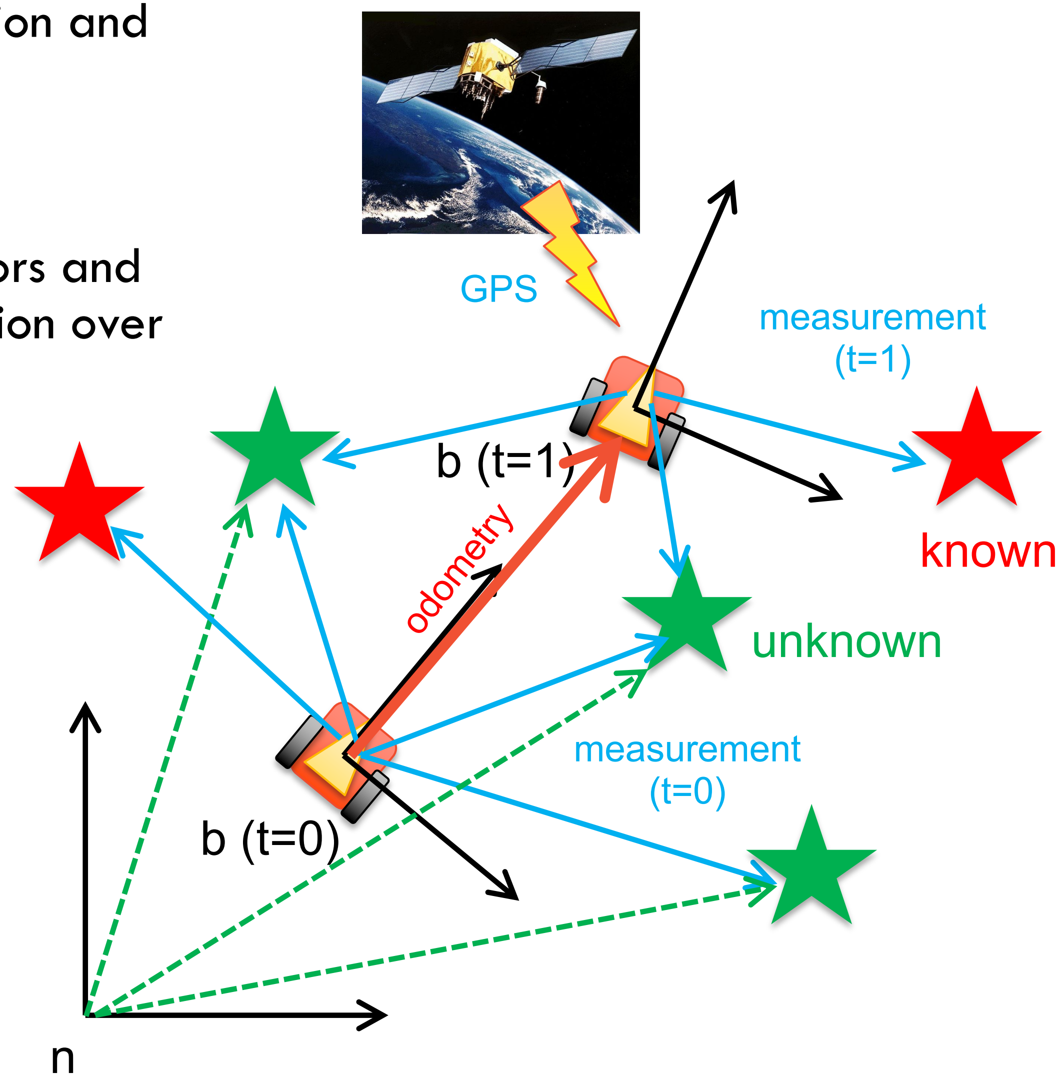
unknown unknown measured

Types of Localisation and Mapping

- Many practical robot navigation and mapping problems involve combinations of information sources/estimation tasks
  - Sensors contain noise/errors and may not provide information over all poses states

- Localisation and mapping is a multi-sensor estimation problem

- Even though some estimated states are directly measured, it still may improve results to estimate/optimize these (e.g. GPS-aided SLAM)



Localisation, Mapping and Estimation

- As a state-estimation problem:
  - Estimated states (x)
  - Sensor Measurements/External information (z)
  - Assume errors/noise on measurements

$$\mathbf{x} = [\mathbf{p}_1^n, \mathbf{\Psi}_2^n, \mathbf{p}_2^n, \mathbf{\Psi}_2^n, \ldots, \mathbf{p}_k^n, \mathbf{\Psi}_k^n, \mathbf{p}_{l1}, \mathbf{p}_{l2}, \mathbf{p}_{l3} \ldots]$$

$$\mathbf{z} = h(\mathbf{x}) + \delta z$$

$$\mathbf{x} = [\mathbf{p}_k^n, \mathbf{\Psi}_k^n]$$

Process model

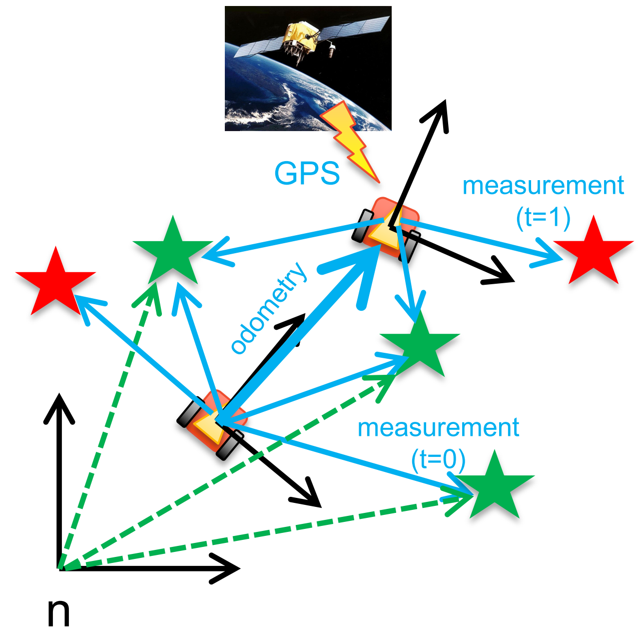
rearrange

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{q} \longrightarrow \mathbf{u}_k = h(\mathbf{x}_{k+1}, \mathbf{x}_k) + \mathbf{q}$$

Different frameworks, but typically achieving the same thing!

Localisation, Mapping and Estimation

- Typical estimation problems:
  - Maximum likelihood (e.g. weighted least-squares)
  - Filtering/sequential estimation (e.g Kalman Filter)



$$\mathbf{z} = \mathbf{h}(\mathbf{x}) \qquad \mathbf{x}^* = \underset{\hat{\mathbf{x}}}{\text{arg min}} \frac{1}{2}(\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}))^T \mathbf{Q}^{-1} (\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}))$$

$$\nabla \mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x} = \hat{\mathbf{x}}}$$

$$\mathbf{Y} = \nabla \mathbf{H}^T \mathbf{R}^{-1} \nabla \mathbf{H}$$

$$\mathbf{y} = \nabla \mathbf{H}^T \mathbf{R} (\mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}))$$

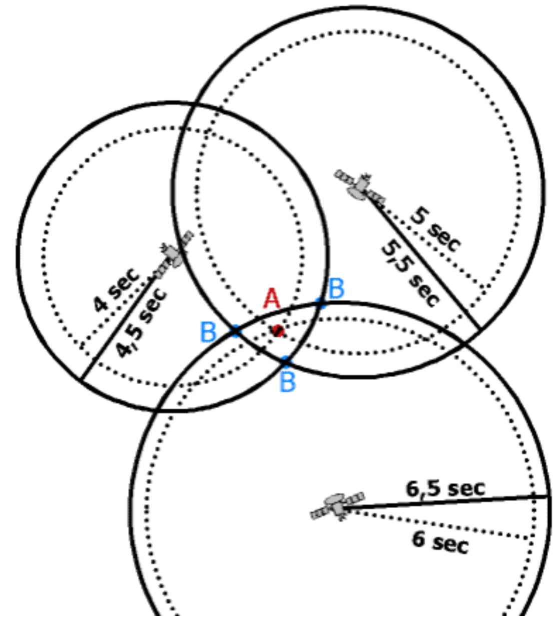
$$\boxed{\mathbf{P} = \mathbf{Y}^{-1}}$$

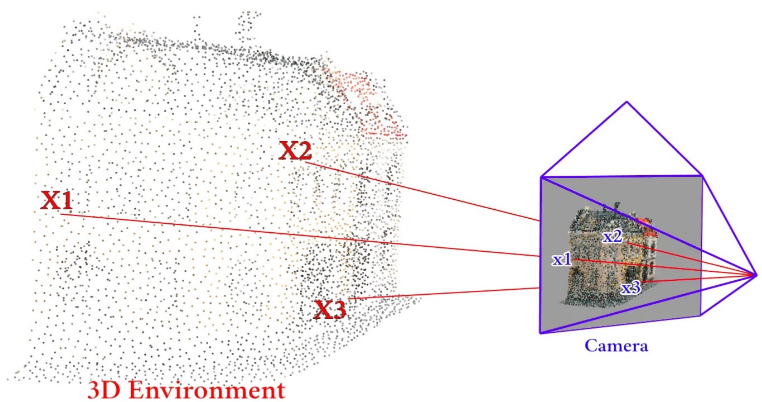
Covariance matrix Representing uncertainty in estimate

$$\hat{\mathbf{x}} = \hat{\mathbf{x}} + \mathbf{Y} \setminus \mathbf{y}$$

Instantaneous Localisation (in time)

- Many localization problems require us to track the robot over multiple time steps, but some can be solved using sensor data from a snap-shot in time
- Examples include:
  - **Beacon-based Localisation**: determine pose from robot-relative observations to known landmarks
  - **Multilateration**: where range measurements are made to known landmarks
  - **Perspective-n-Point**: determine pose from measured projection (i.e. image) of known landmarks
- Many of these problems are solved using maximum likelihood approaches



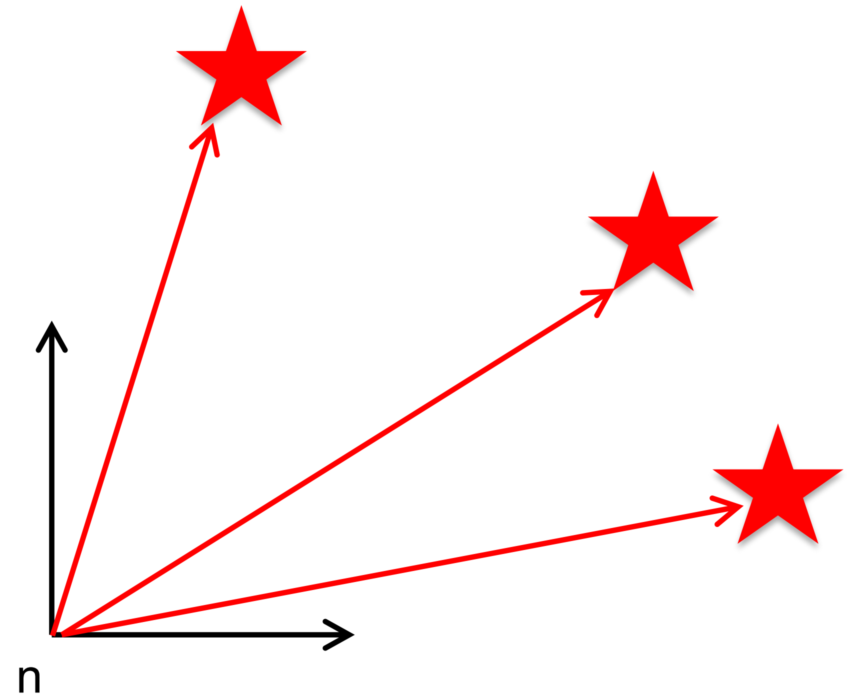


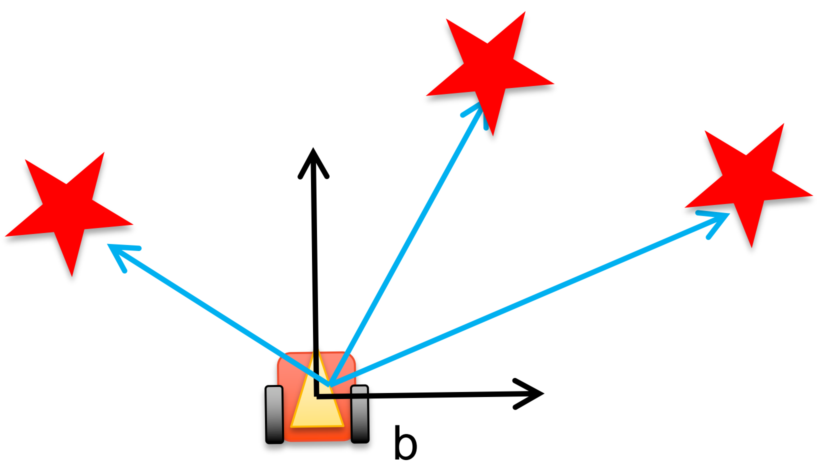
Courtesy: Rashik Shrestha

Beacon-based Localisation

- Robot-relative measurements made to known landmarks
- Solution: find a pose that minimizes the difference between “predicted” landmark locations and know landmark locations

Known Map of Landmarks



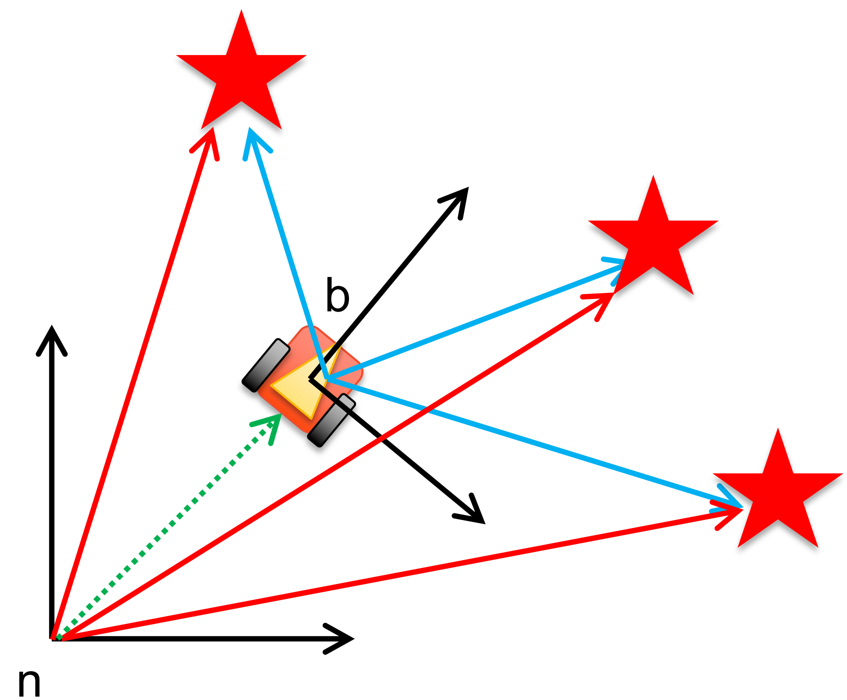


Robot-centric Measurements

Beacon-based Localisation

$$\mathbf{p}_l^n = \mathbf{p}^n + C_b^n \mathbf{p}_{lr}^b$$

- Robot-relative measurements made to known landmarks
- Solution: find a pose that minimizes the difference between "predicted" landmark locations and know landmark locations
- This can be posed as a least-squares estimation problem:

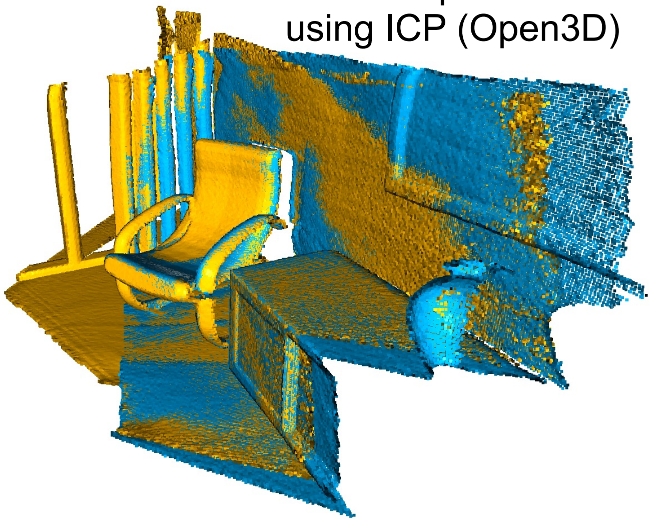


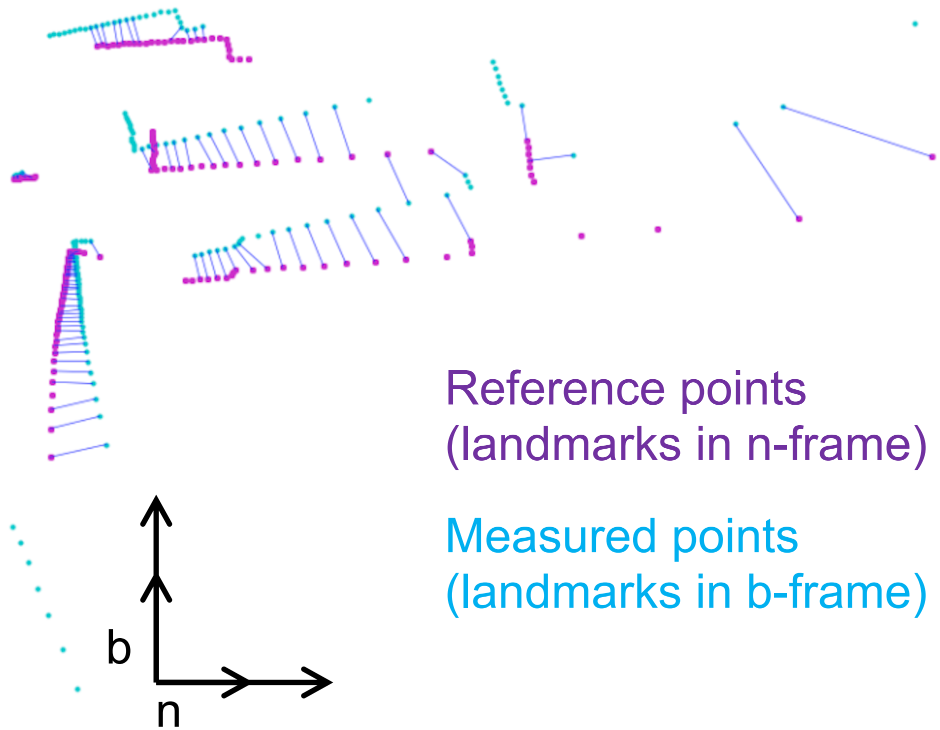
$$\mathbf{p}^{n*}, C_b^{n*} = \operatorname{argmin} \sum_{i=1}^{n} ||\mathbf{p}_{li}^n - (\mathbf{p}^n + C_b^n \mathbf{p}_{lri}^b)||^2$$

Iterated Closest Point (ICP)

- Typically associations between observed and known landmarks might not be known
- Iterated Closest Point (ICP) is an algorithm that allows sets of measured vs. reference points to be matched:
  1. Using current estimate of relative pose, associate points based on distance
  2. Perform point-to-point least square optimization
  3. Repeat from (1) until convergence

Matched points in 3D using ICP (Open3D)

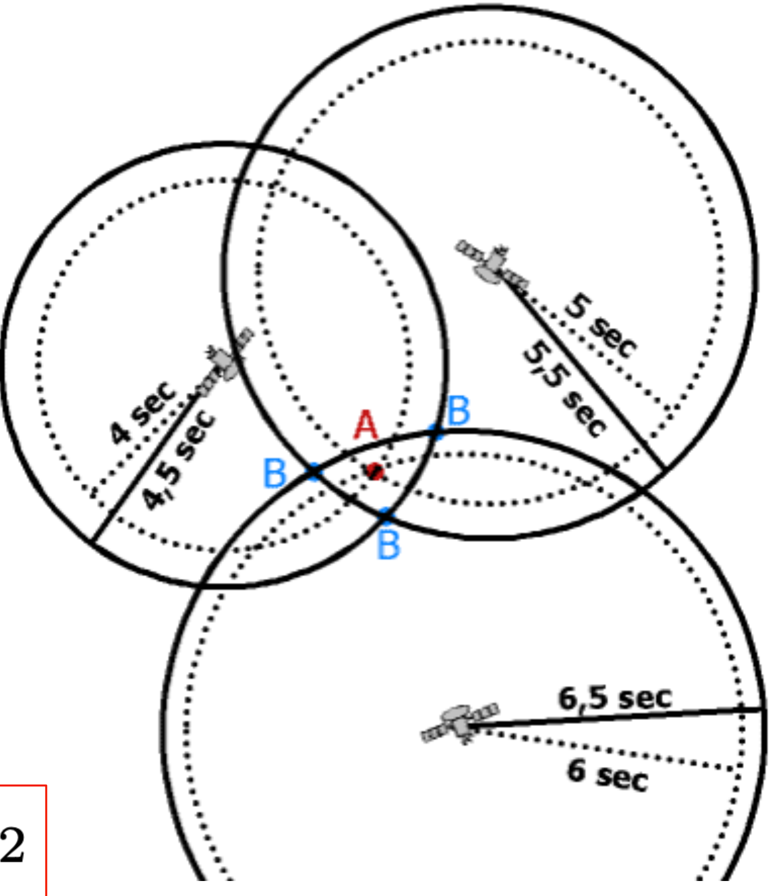




Courtesy: Andrew Kramer

Multilateration

- Multilateration: process of determining location when only the range (or signal time-of-flight) is known to beacons
- Processing used in Global Navigation Satellite Systems (GNSS) or mobile-phone positioning
- Typically position is solved as a maximum likelihood estimate



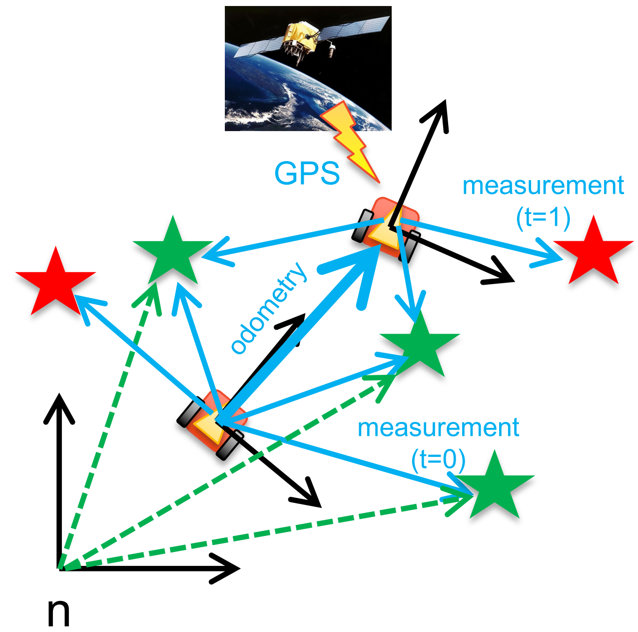
$$\rho_1 = \sqrt{(X_{SV1}-x)^2+(Y_{SV1}-y)^2+(Z_{SV1}-z)^2}+cb_U+\chi_1$$
$$\rho_2 = \sqrt{(X_{SV2}-x)^2+(Y_{SV2}-y)^2+(Z_{SV2}-z)^2}+cb_U+\chi_2$$
$$\rho_3 = \sqrt{(X_{SV3}-x)^2+(Y_{SV3}-y)^2+(Z_{SV3}-z)^2}+cb_U+\chi_3$$
$$\rho_4 = \sqrt{(X_{SV4}-x)^2+(Y_{SV4}-y)^2+(Z_{SV4}-z)^2}+cb_U+\chi_4$$

$$\mathbf{z} = [\rho_1, \rho_2, \ldots, \rho_n]$$

$$\boxed{\mathbf{p}^{n*} = \operatorname{argmin} \sum (\rho - f(\mathbf{p}^n))^2}$$

Localisation, Mapping and Estimation

- Typical estimation problems:
  - Maximum likelihood (e.g. weighted least-squares)
  - Filtering/sequential estimation (e.g Kalman Filter)



$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \qquad \mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1})$$

update

$$\begin{aligned} \mathbf{x}^-_{k+1} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{P}^-_{k+1} &= \nabla\mathbf{F}\mathbf{P}_k\nabla\mathbf{F}^T + \nabla\mathbf{G}\mathbf{Q}_k\nabla\mathbf{G}^T \end{aligned}$$

prediction

$$\begin{aligned} \mathbf{x}^+_{k+1} &= \mathbf{x}^-_{k+1} + \mathbf{W}_{k+1}\nu_{k+1} \\ \mathbf{P}^+_{k+1} &= \mathbf{P}^-_{k+1} - \mathbf{W}_{k+1}\mathbf{S}_{k+1}\mathbf{W}^T_{k+1} \end{aligned}$$

$$\nu_{k+1} = \mathbf{z}_{k+1} - \mathbf{h}(\mathbf{x}^-_{k+1}) \qquad \text{"Innovation"}$$

$$\mathbf{S}_{k+1} = \nabla\mathbf{H}\mathbf{P}^-_{k+1}\nabla\mathbf{H}^T + \mathbf{R}_{k+1} \qquad \text{"Innovation Covariance"}$$

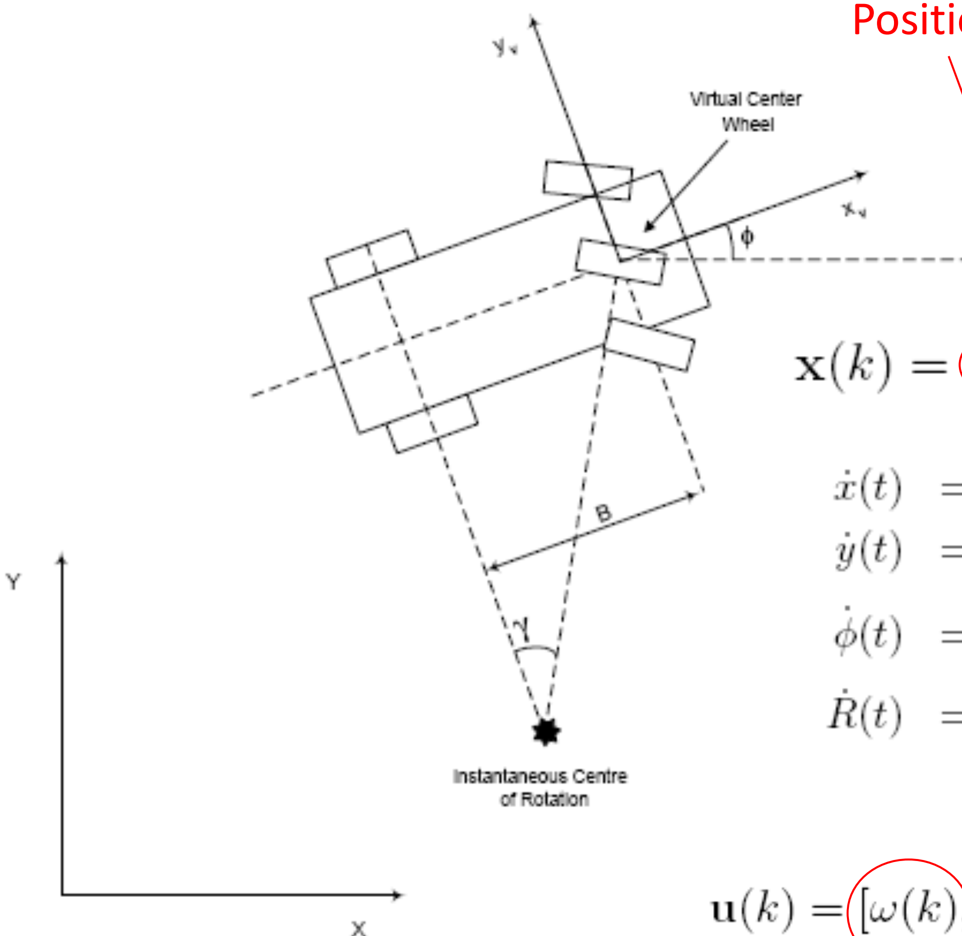
$$\mathbf{W}_{k+1} = \mathbf{P}^-_{k+1}\nabla\mathbf{H}^T\mathbf{S}^{-1}_{k+1} \qquad \text{"Kalman Weight"}$$

Example: Beacon-based localization for straddle carrier

- Autonomous straddle carrier navigation system
- Navigation occurs in a structure environment: reflective beacons placed into the operating environment
- Localisation system uses:
  - Proprioceptive sensors: wheel speed encoders and steering angle measurment
  - Radar and lidar observations to beacons with known positions

H. Durrant-White, “An autonomous guided vehicle for cargo handling applications”, IJRR, 1996

Example: Beacon-based localization for straddle carrier



Position

Wheel Radius

State Vector

$$\mathbf{x}(k) = [x(k), y(k), \phi(k), R(k)]^T,$$

$$\begin{aligned}
\dot{x}(t) &= R(t)\omega(t)\cos(\phi(t) + \gamma(t)) \\
\dot{y}(t) &= R(t)\omega(t)\sin(\phi(t) + \gamma(t)) \\
\dot{\phi}(t) &= \frac{R(t)\omega(t)}{B}(\sin\gamma(t)) \\
\dot{R}(t) &= 0,
\end{aligned}$$

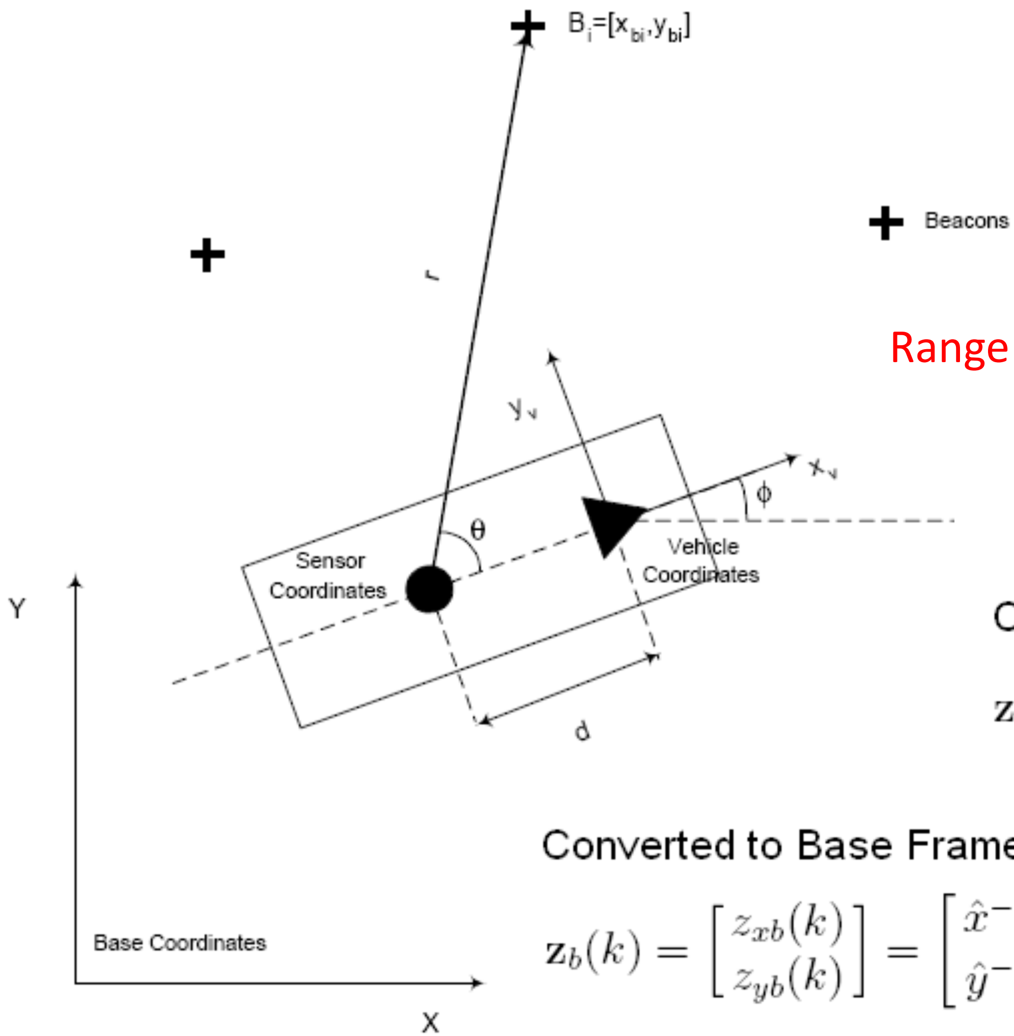
Heading Angle

Steering Angle

$$\mathbf{u}(k) = [\omega(k), \gamma(k)]^T$$

Wheel Speed

Example: Beacon-based localization for straddle carrier



Bearing to Beacon

Range to Beacon

$$\mathbf{z}(k) = \begin{bmatrix} r(k) & \theta(k) \end{bmatrix}^T$$

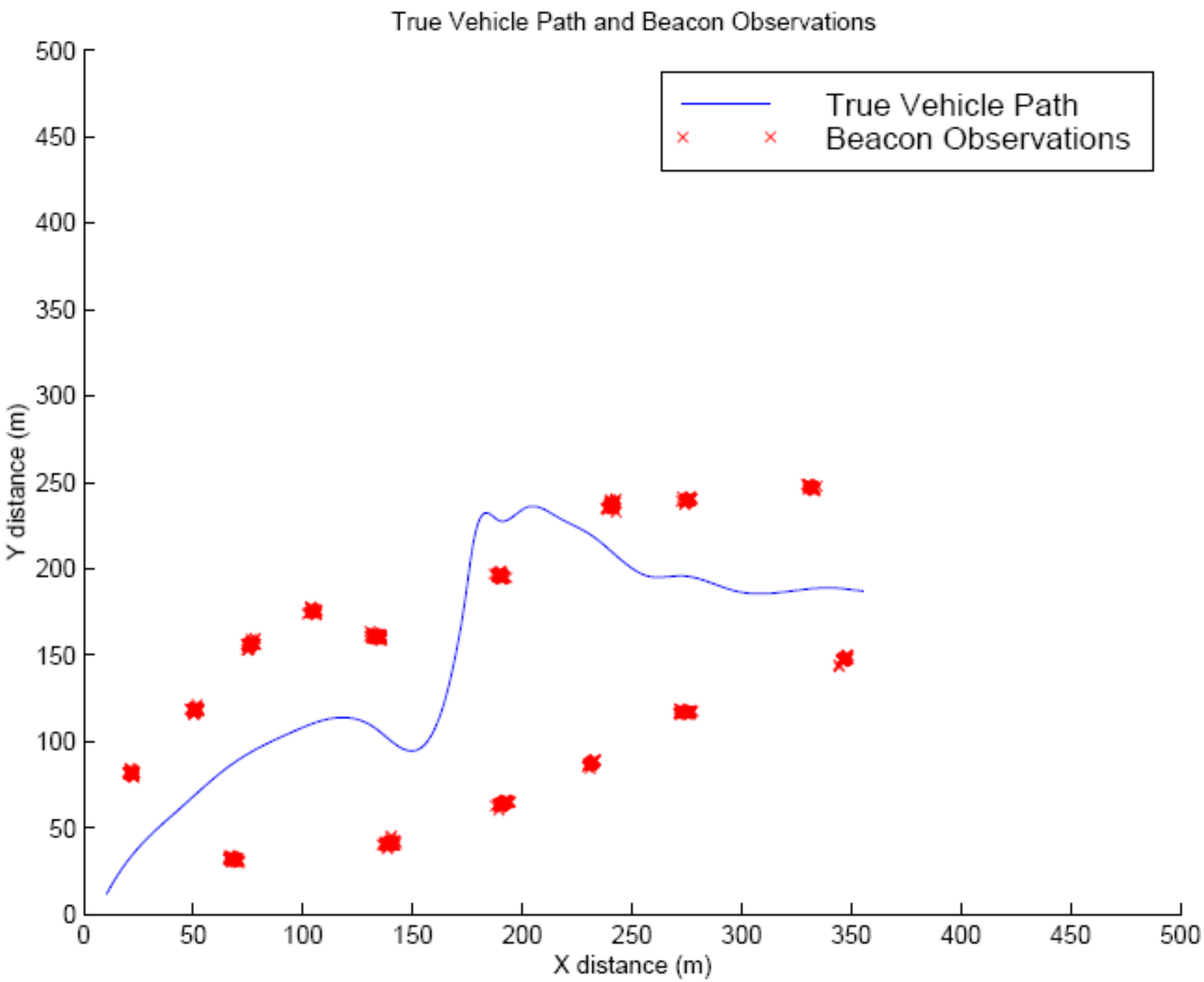
Converted to Cartesian Coordinates:

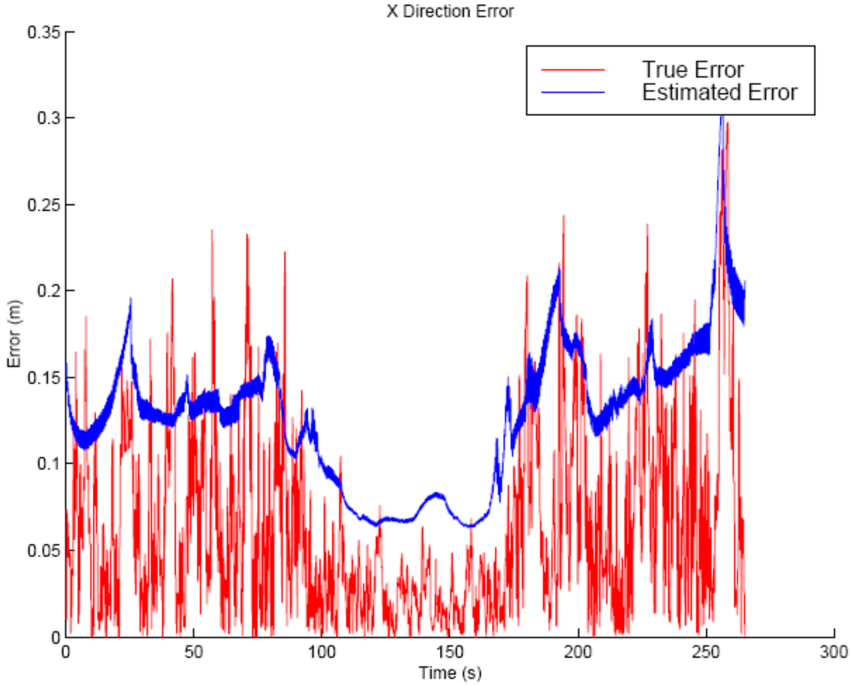
$$\mathbf{z}_v(k) = \begin{bmatrix} z_{xv}(k) \\ z_{yv}(k) \end{bmatrix} = \begin{bmatrix} d + r(k)\cos\theta(k) \\ r(k)\sin\theta(k) \end{bmatrix}$$

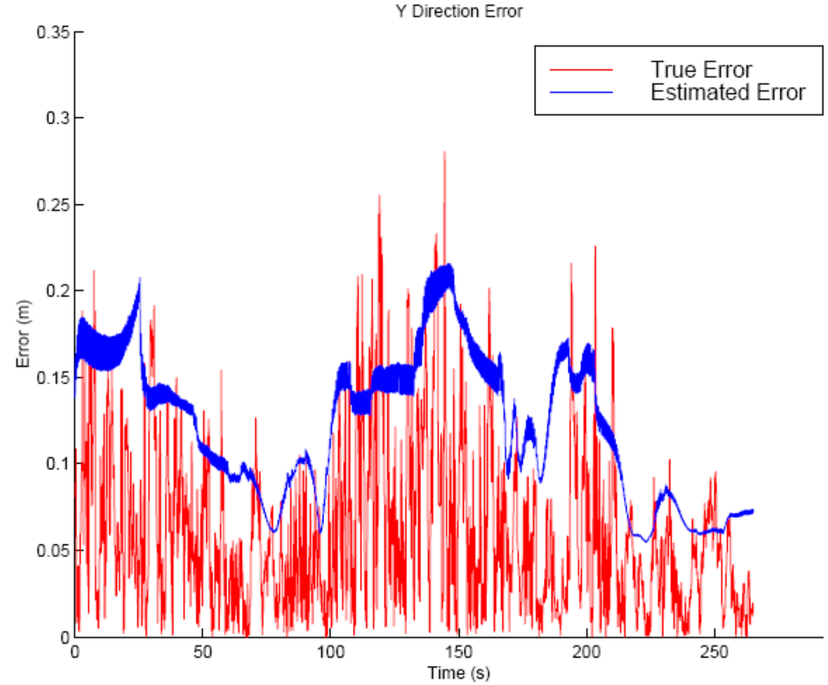
Converted to Base Frame:

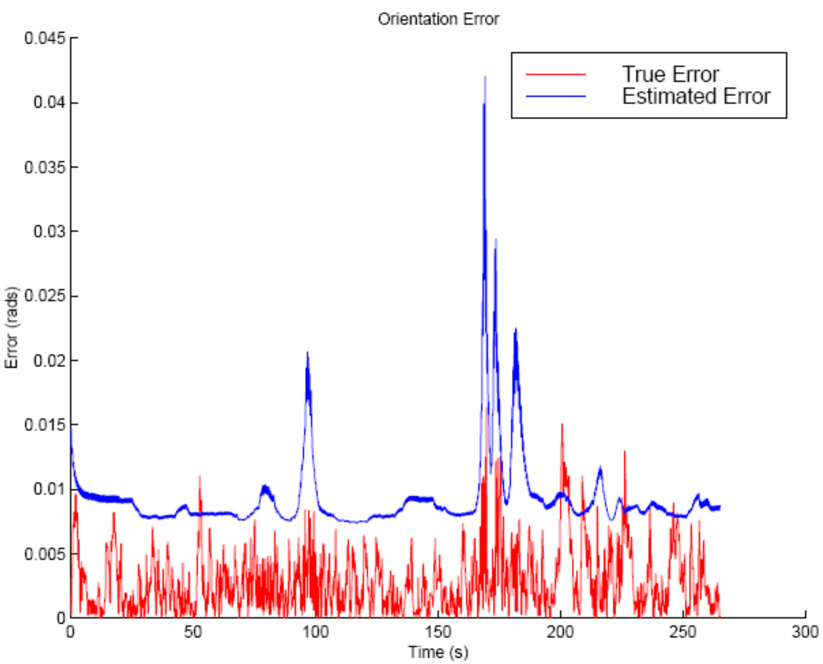
$$\mathbf{z}_b(k) = \begin{bmatrix} z_{xb}(k) \\ z_{yb}(k) \end{bmatrix} = \begin{bmatrix} \hat{x}^-(k) + z_{xv}(k)\cos\hat{\phi}^-(k) - z_{yv}(k)\sin\hat{\phi}^-(k) \\ \hat{y}^-(k) + z_{xv}(k)\sin\hat{\phi}^-(k) + z_{yv}(k)\cos\hat{\phi}^-(k) \end{bmatrix}$$

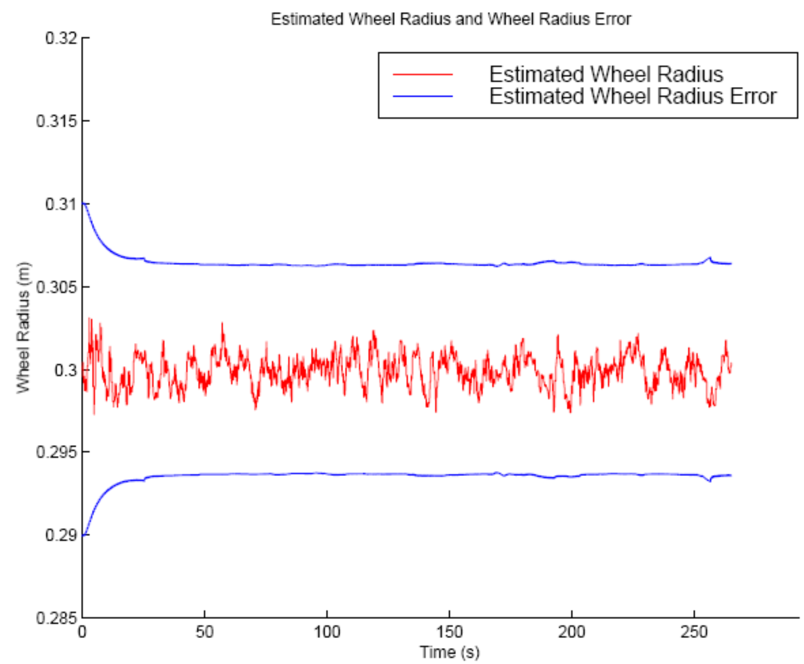
Example: Beacon-based localization for straddle carrier




X Direction Error
True Error
Estimated Error
Error (m)
Time (s)
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
50
100
150
200
250
300

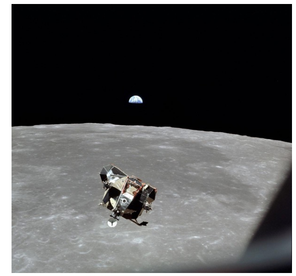

Y Direction Error
True Error
Estimated Error
Error (m)
Time (s)
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
50
100
150
200
250


Orientation Error
True Error
Estimated Error
Error (rads)
Time (s)
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250
300


Estimated Wheel Radius and Wheel Radius Error
Estimated Wheel Radius
Estimated Wheel Radius Error
Wheel Radius (m)
Time (s)
0.32
0.315
0.31
0.305
0.3
0.295
0.29
0.285
0
50
100
150
200
250

Example: Localisation using aided inertial navigation

- Aided-inertial navigation is used in a wide range of robotic localization systems including land-based, aerial and marine platforms
- An Inertial Measuring Unit (IMU) consists of a set of accelerometers and rate-measuring gyros (usually in a tri-axial configuration)
- By integrating acceleration and rotation rate over time, we can dead-reckon pose
  - But errors accumulate
- Other sensor information (e.g. GPS position observations) can be used to constrain drift

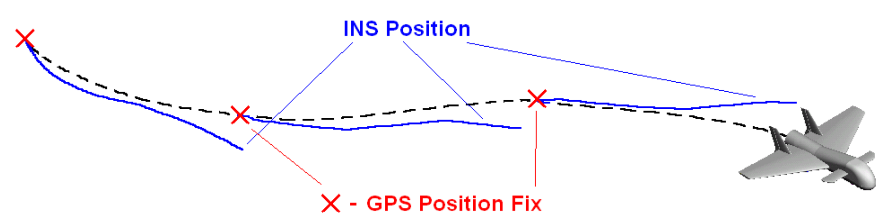


IMU

$\hat{\mathbf{f}}_k^b$ Specific force

(force applied to case per proof mass)

$\hat{\omega}_{ib,k}^b$ Angular velocity



Inertial Navigation: Principles

- Important note about Accelerometers: they measure "specific force", that is inertial forces, not acceleration!
- If the platform is affected by gravitational acceleration (almost always), then it needs to be accounted for

Measured
Specific Force

$$a = \mathbf{f} + \mathbf{g}$$

Total Acceleration

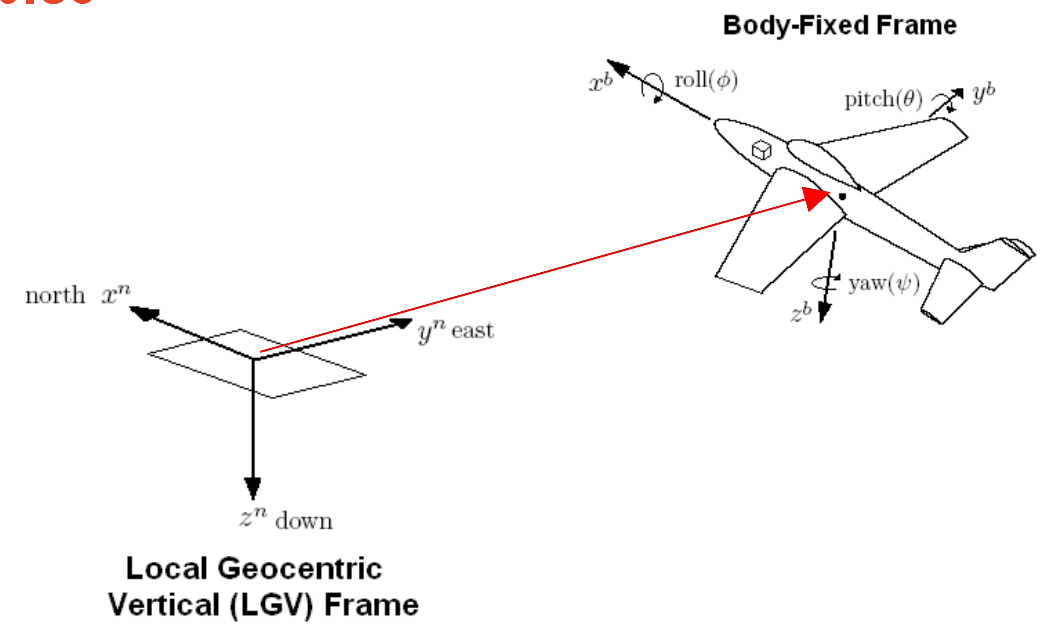
Acceleration due
to gravity

IMU

$$\mathbf{v}_t = \mathbf{v}_{t0} + \int_{t0}^{t} [\mathbf{f}(t) + \mathbf{g}]$$

$$\mathbf{p}_t = \mathbf{p}_{t0} + \int_{t0}^{t} \mathbf{v}(t)$$

Inertial Navigation: Principles



- As the platform moves, it also rotates: specific force measurements are made relative to a body-fixe frame
- The orientation of the body frame w.r.t a navigation frame n must be known in order to resolve the specific force vector into navigation-frame coordinates before integration
- The differential equations used to describe the motion (position and velocity) of the platforms are then:

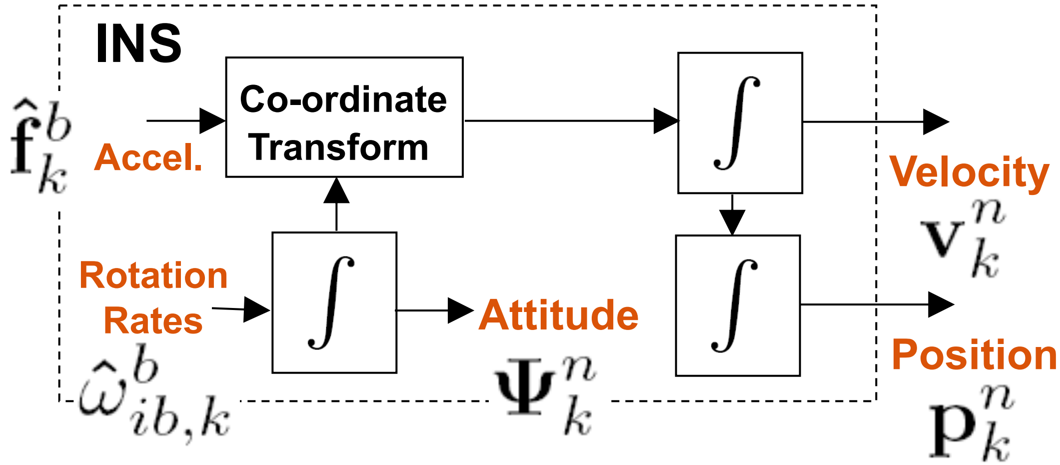
$$\dot{\mathbf{p}}^n = \mathbf{v}^n$$

$$\dot{\mathbf{v}}^n = \mathbf{C}_b^n \mathbf{f}^b + \mathbf{g}^n$$

Background – Inertial Navigation Systems (INS)

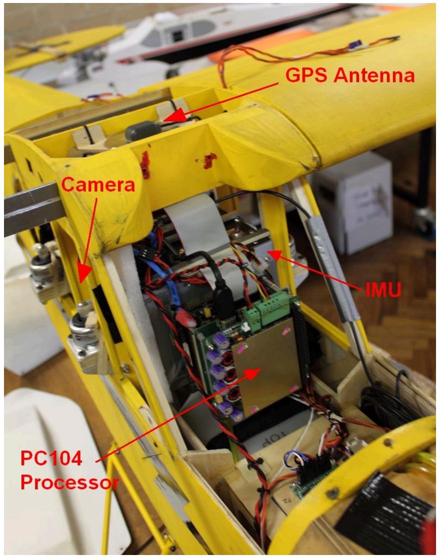
- Typically we start with an initial estimate of the position, velocity and attitude (PVA) of the vehicle
- IMU acceleration and rotation rate measurements are integrated numerically to predict forward the PVA state:

IMU



$$
\begin{aligned}
\mathbf{p}^n_k &= \mathbf{p}^n_{k-1} + \mathbf{v}^n_k \Delta t \\
\mathbf{v}^n_k &= \mathbf{v}^n_{k-1} + [\mathbf{C}^n_b(\hat{\mathbf{f}}^b - \delta \mathbf{f}^b) + \mathbf{g}^n]\Delta t \\
\mathbf{\Psi}^n_k &= \mathbf{\Psi}^n_{k-1} + [\mathbf{E}^n_b(\hat{\omega}^b_{ib} - \delta \omega^b_{ib})]\Delta t
\end{aligned}
$$

GNSS-inertial navigation using the EKF

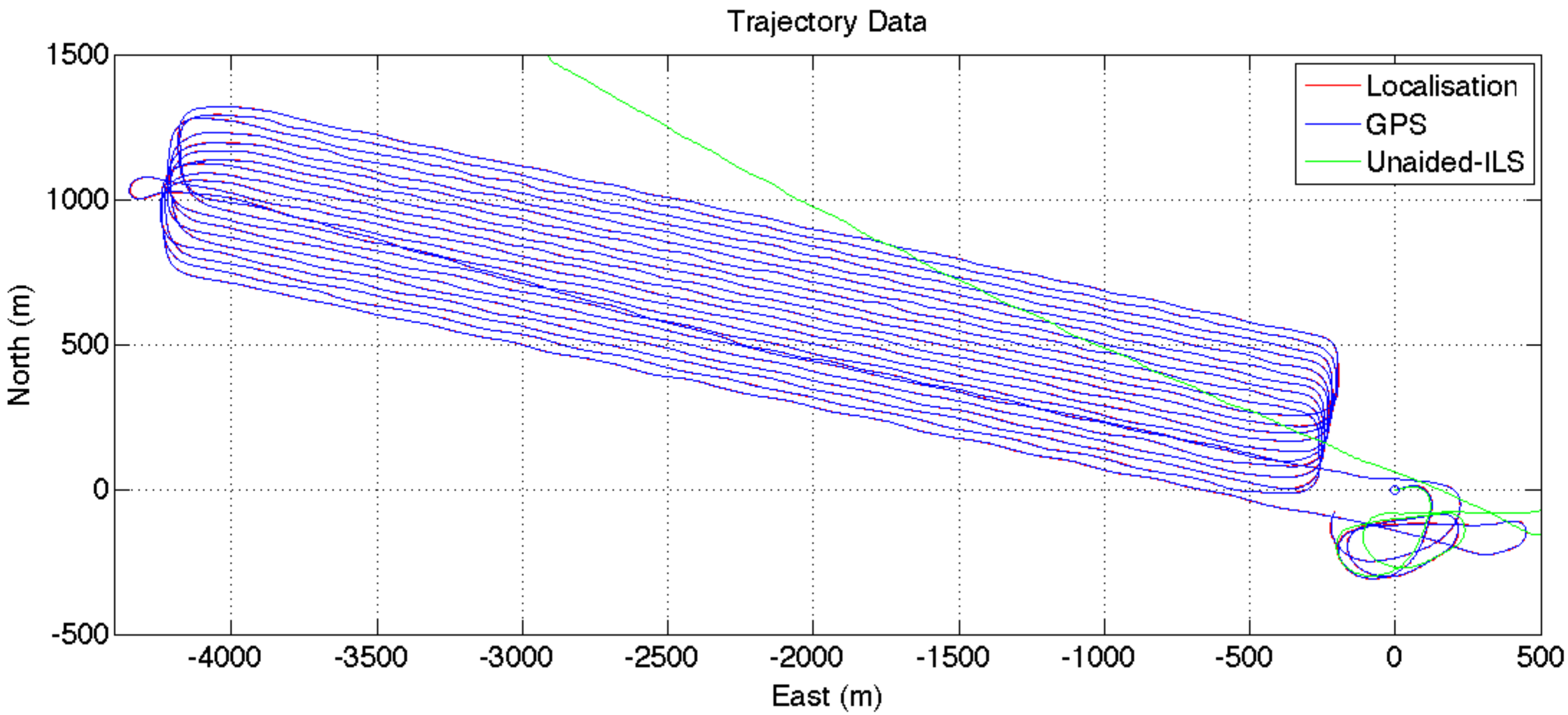


- Common form of aiding sensor is to use position measurements from GNSS (e.g. GPS)

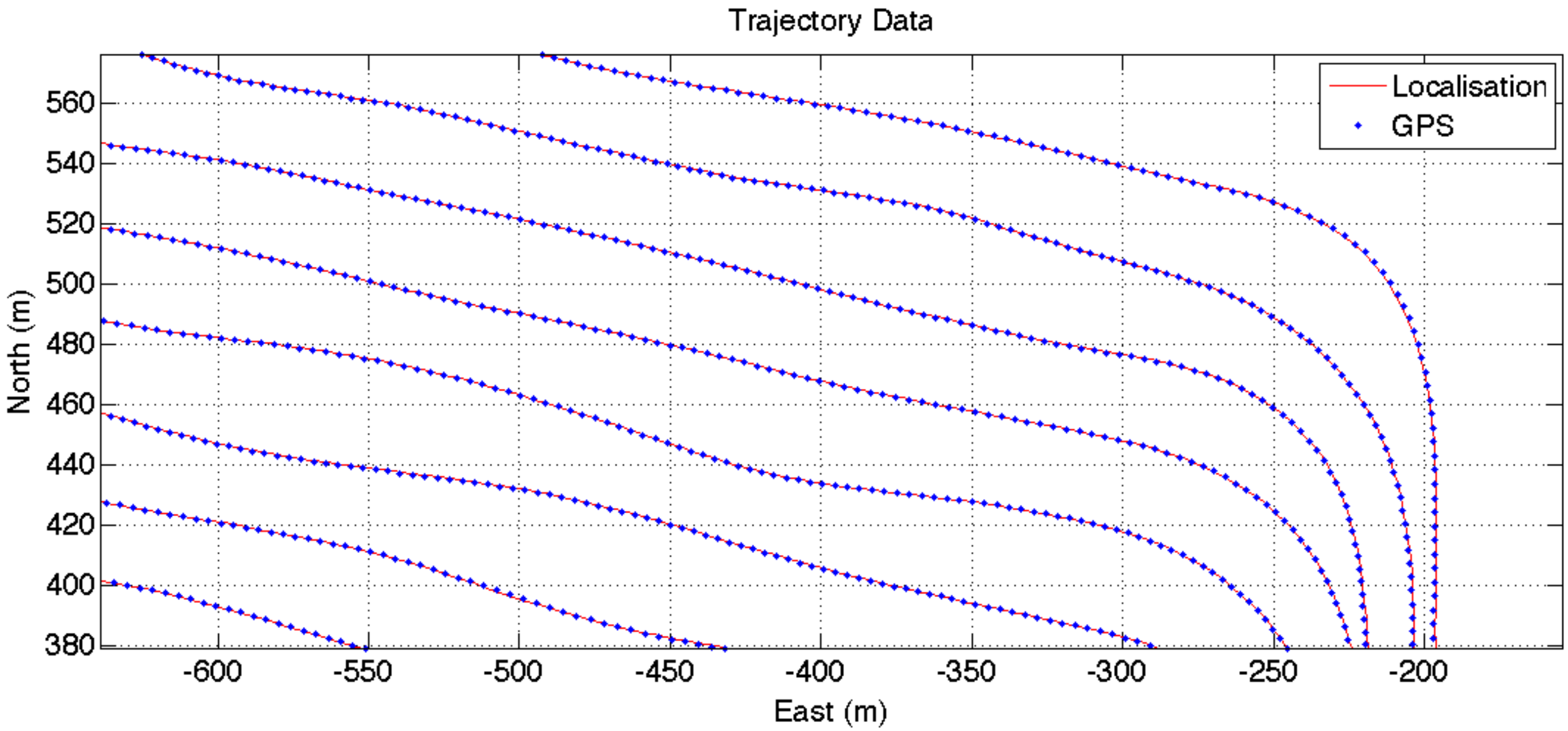
- Multi-sensor estimation can now be implemented using:
  - Process model f(x,u): inertial navigation equations with IMU inputs
  - Observation model h(x): GNSS position measurements

- EKF Prediction step (IMU): approximately 100Hz
- EKF Update step (GPS): approximately 1Hz

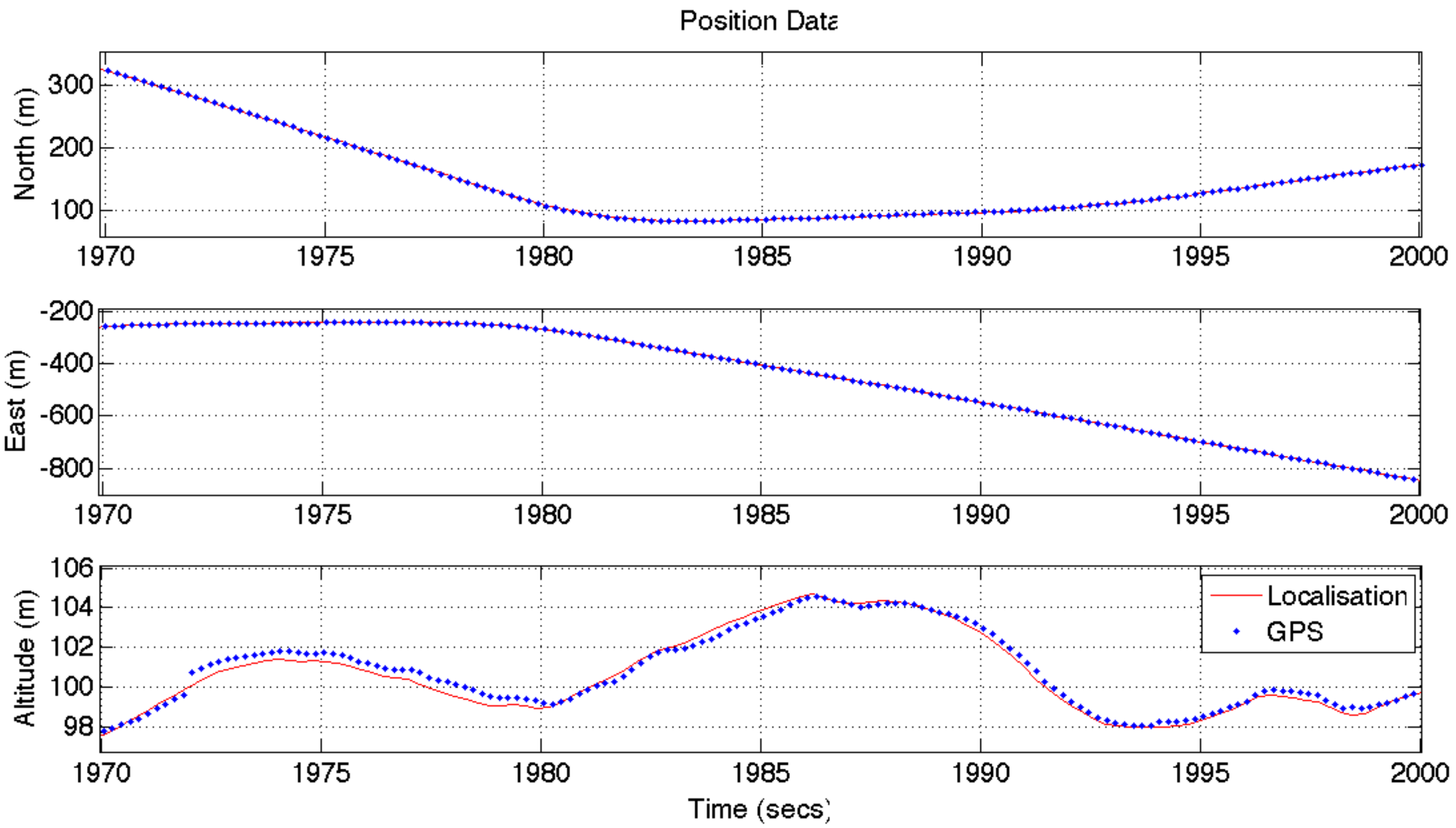
GNSS-inertial navigation using the EKF



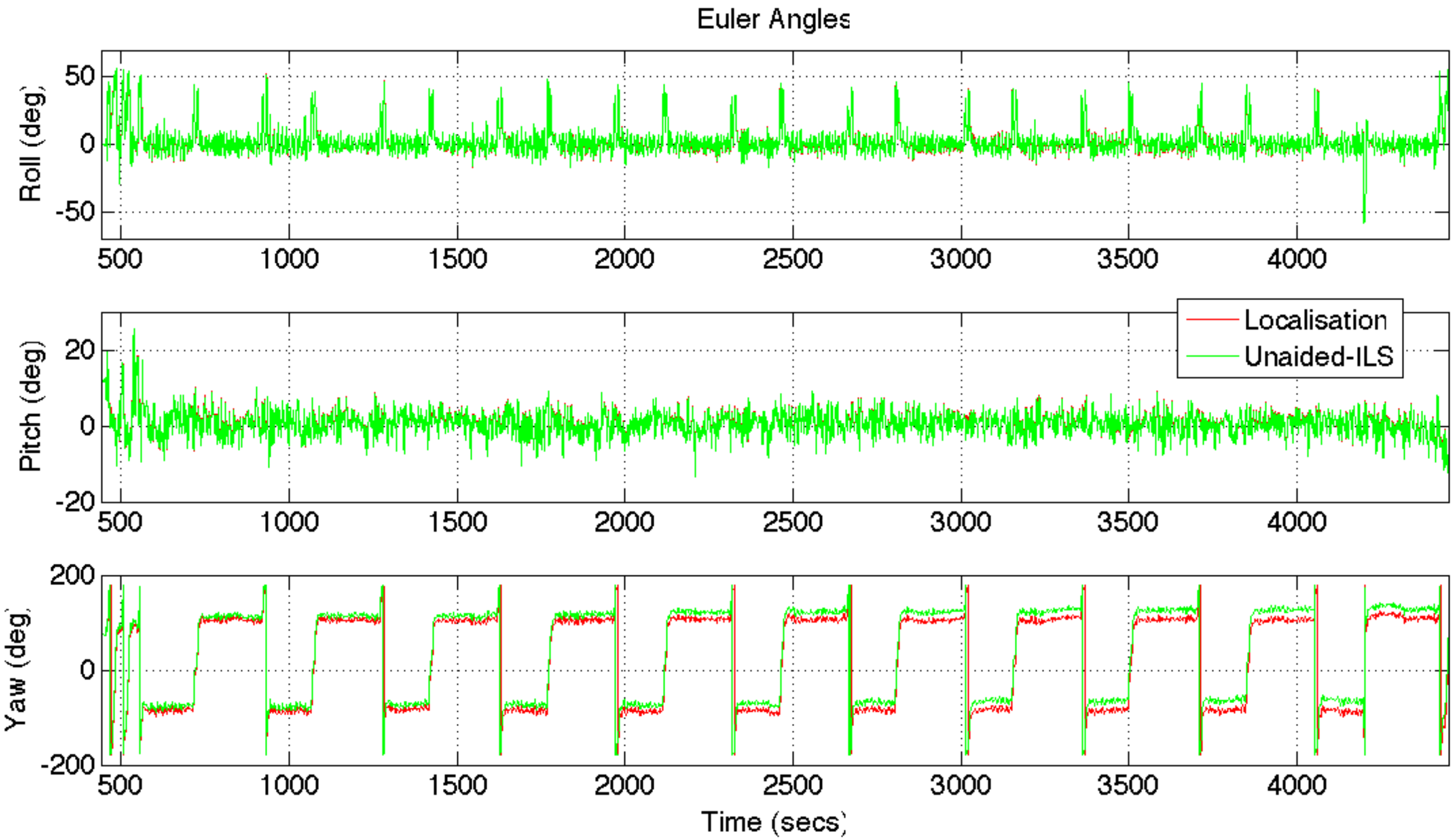
GNSS-inertial navigation using the EKF



GNSS-inertial navigation using the EKF



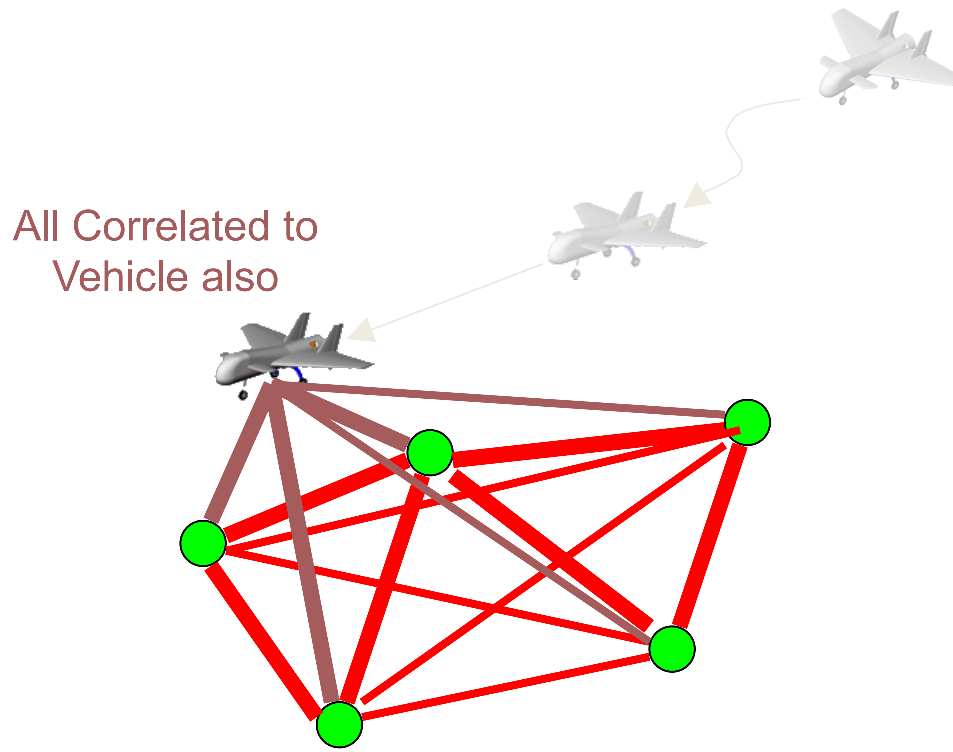
GNSS-inertial navigation using the EKF



- Orientation (Euler angles) can be estimated accurately and without drift via multi-sensor fusion in the EKF, even though no sensor observes these states directly

Simultaneous Localisation and Mapping (SLAM)

- In SLAM, we include locations of unknown landmarks into the estimation problem
- Typically it is assumed landmarks are static objects, relative to our navigation frame of reference
- SLAM is typically formulated as a recursive state estimation problem
  - However might also be solved at one time using a history of gathered sensor measurements



$\mathbf{x}_k$	Current estimate
$\mathbf{P}_k$	Estimate Covariance

$$\mathbf{x} = [\mathbf{p}_k^n, \mathbf{\Psi}_k^n, \mathbf{p}_{l1}, \mathbf{p}_{l2}, \mathbf{p}_{l3} \cdots]$$

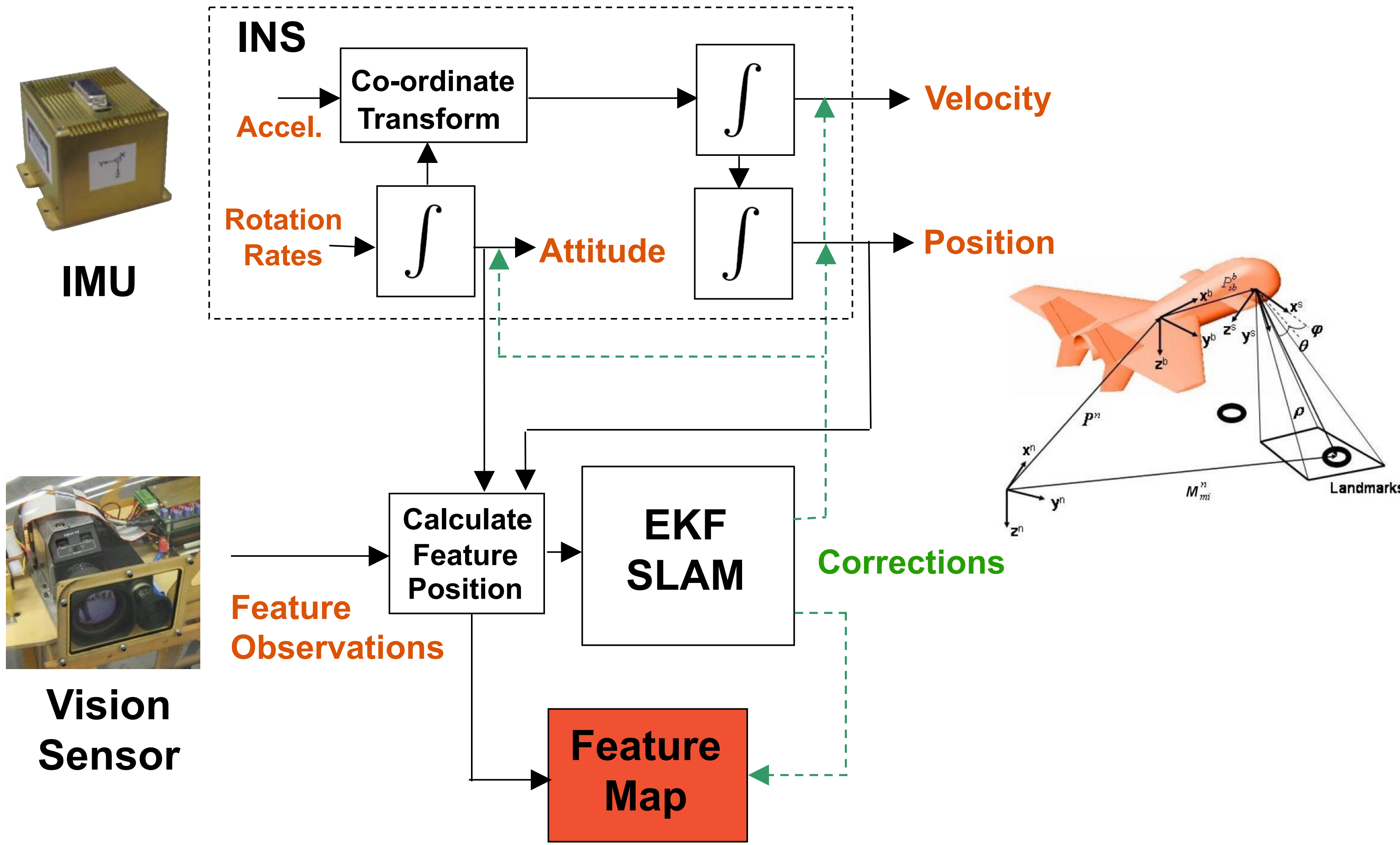
Simultaneous Localisation and Mapping (SLAM): Complications

- Loop Closure: How do we recognize and associate measurements of previously seen landmarks?
  - Relies on methods in place recognition, feature descriptors and matching, often also use current location estimate (and uncertainty!) to aid matching process

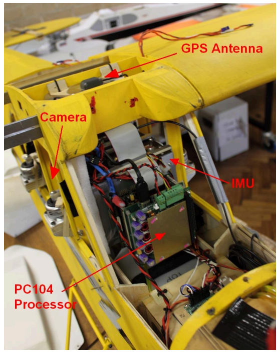
- Variant of SLAM problem: unknown initial robot pose:
  - Without using external sources of information (e.g. known landmarks, GNSS) can only determine map relative to starting pose

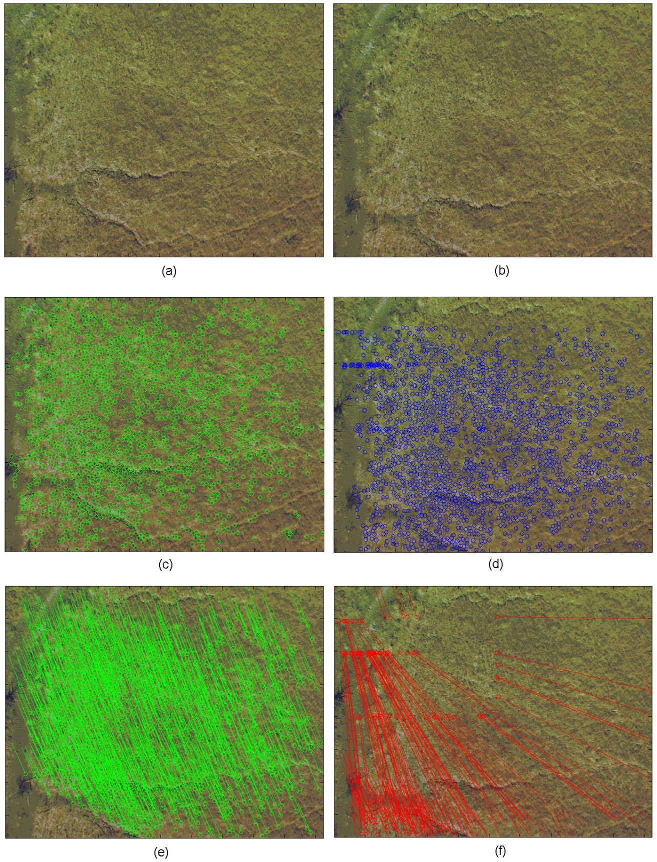
- Practical estimation issues:
  - Errors can build-up over large loop closures, violations of linearization assumptions and Gaussian error distributions
  - Computation: large number of states and measurements, scalability depends on methods used

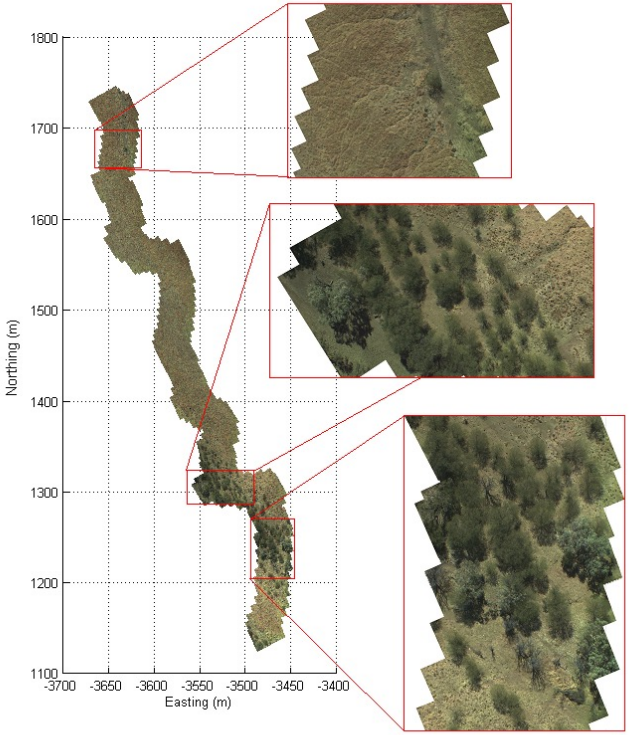
Inertial-Vision SLAM



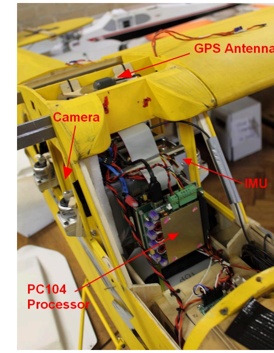
Inertial-Vision SLAM

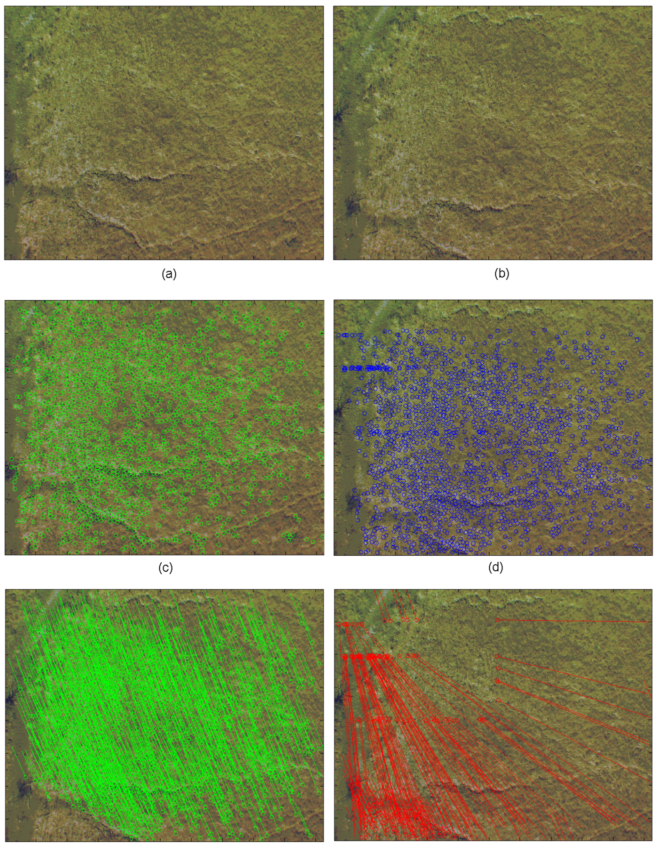


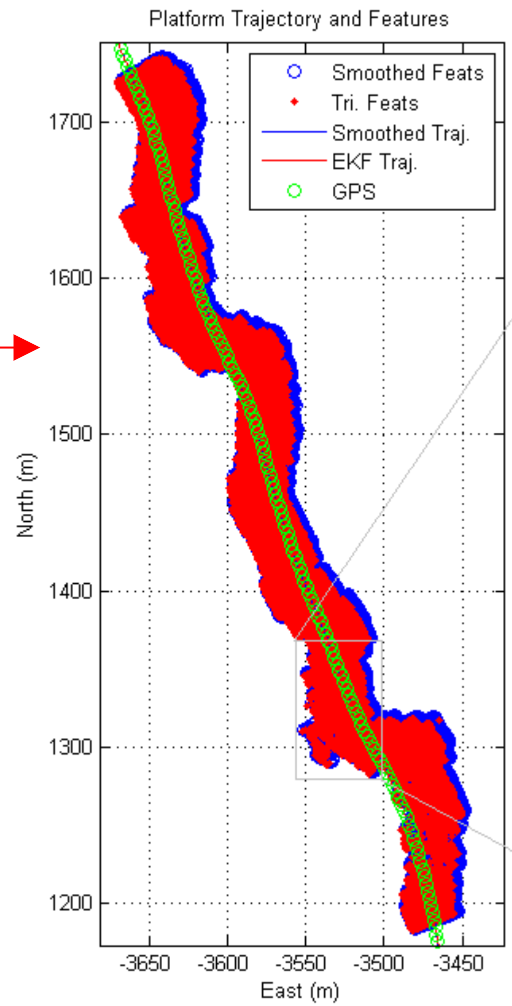


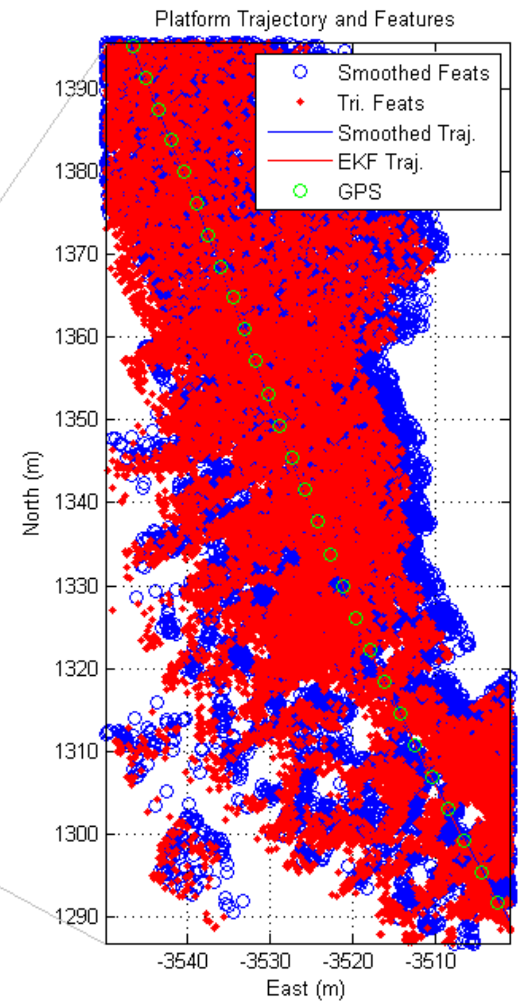


Inertial-Vision SLAM

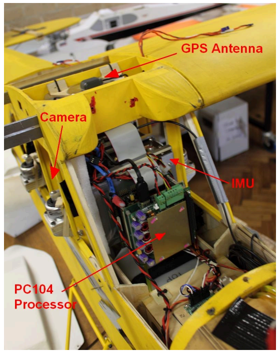


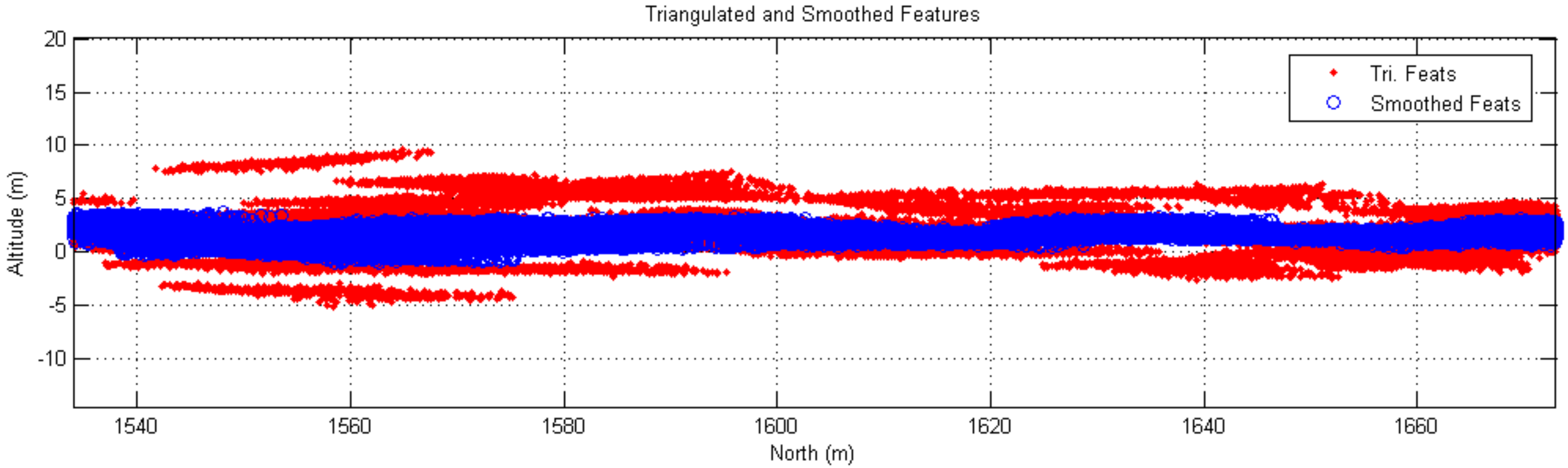




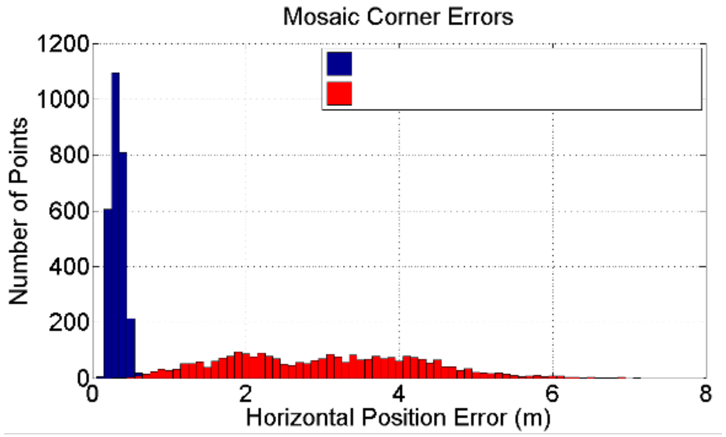


Inertial-Vision SLAM





– Direct geo-referencing (red) vs. IMU-GPS-SLAM fusion: improved quality of map via joint estimation/optimisation of poses and map features



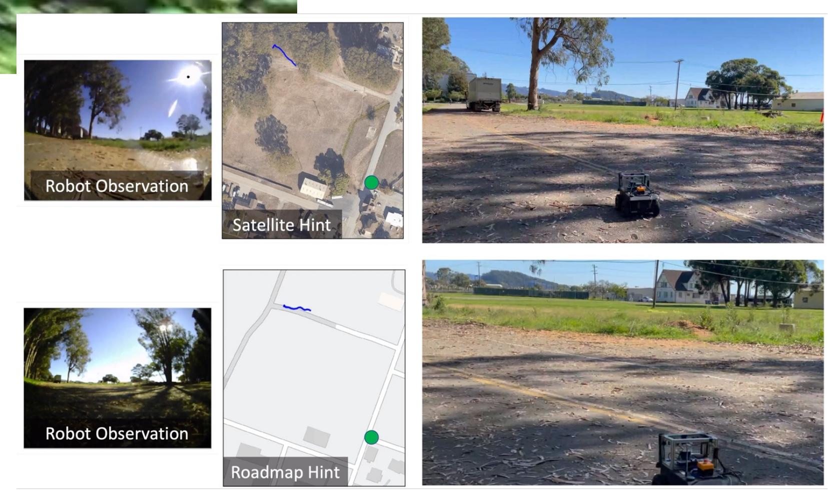
Overview of the lecture:

- Introduction to Robot Localisation and Mapping
- Robot Localisation and Mapping:
  - Basics: Coordinate Systems and Frames of Reference
  - Localisation and Mapping as an estimation problem
- Examples:
  - Beacon-based localisation
  - Aided-inertial navigation
  - Simultaneous Localisation and Mapping (SLAM)
- Conclusions and Future outlook

Autonomous Navigation and end-to-end Deep Learning

- Do we really need to perform localization and mapping explicitly?

N. Smolyanskiy, A. Kamenev, J. Smith, S. Birchfield, "Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness", IROS, 2017



D. Shah and S. Levine, "ViKiNG: Vision-Based Kilometer-Scale Navigation with Geographic Hints" RSS, 2022

Conclusions and Summary

- Purpose of localisation and mapping for mobile robots: capabilities that underlie navigation, control, manipulation, higher-level objectives of robotic systems

- Different representations for location and maps

- Connections/relationship between good localisation and good mapping

- Role of sensor fusion in localisation and mapping:
  - Sensors on their own don't typically cover all the relevant information, but when fused they do e.g. IMU-GPS