EDINBURGH UNIVERSITY
FORMULA STUDENT

SLAM

Simultaneous Localisation and Mapping

6th November 2024

Outline

1. Formula Student AI
2. Reference Problem
3. Localisation
4. Mapping
5. SLAM
6. SLAM Algorithms
7. Applications

Edinburgh University Formula Student

FS-AI - Problem

- There are 4 dynamic events.

- We focus on **Trackdrive**:

FS-AI - Problem

- There are 4 dynamic events.

- We focus on **Trackdrive**:

1. 10 laps of an unknown track

FS-AI - Problem

- There are 4 dynamic events.

- We focus on **Trackdrive**:

1. 10 laps of an unknown track
2. Blue cones on the left
3. Yellow cones on the right
4. Orange cones mark start/finish line

FS-AI - Problem

- There are 4 dynamic events.

- We focus on **Trackdrive**:

1. 10 laps of an unknown track
2. Blue cones on the left
3. Yellow cones on the right
4. Orange cones mark start/finish line
5. **Fastest time wins**

FS AI – The Assumptions

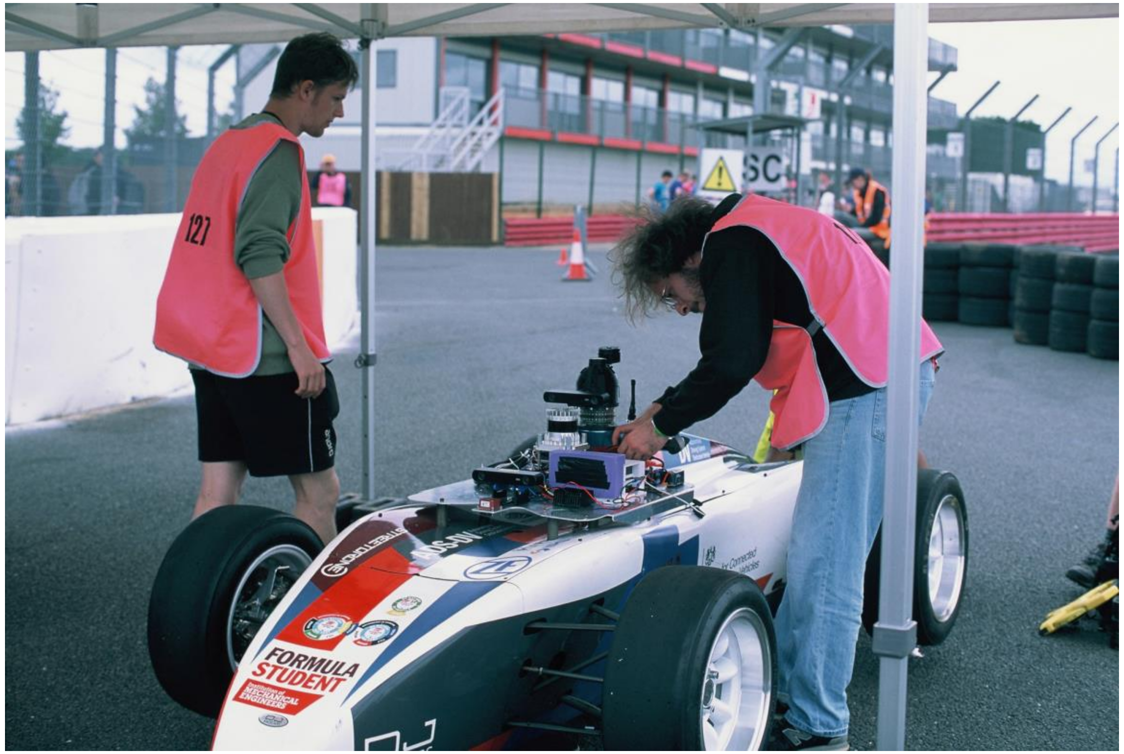
Given to us at each time step:

1. Cone positions near car

2. Estimate of the velocity of the car

FS-AI – Difficulties

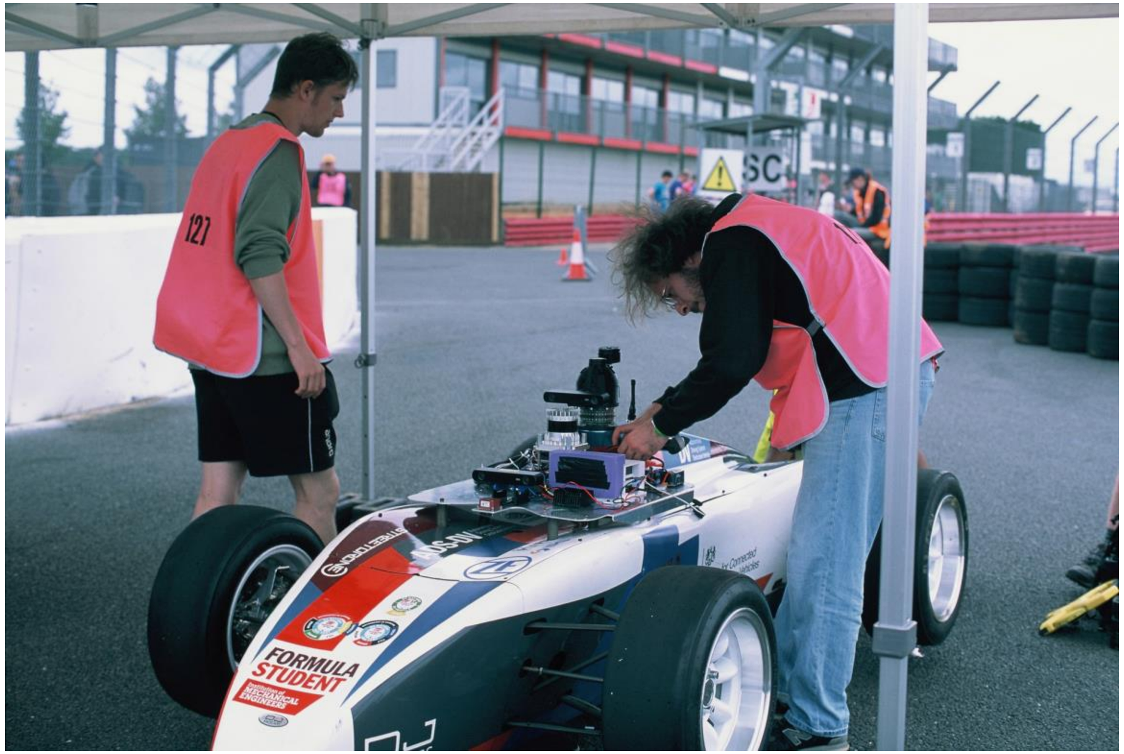
What prevents us from going fast?



FS-AI – Difficulties

What prevents us from going fast?

What if we see a turn too late?



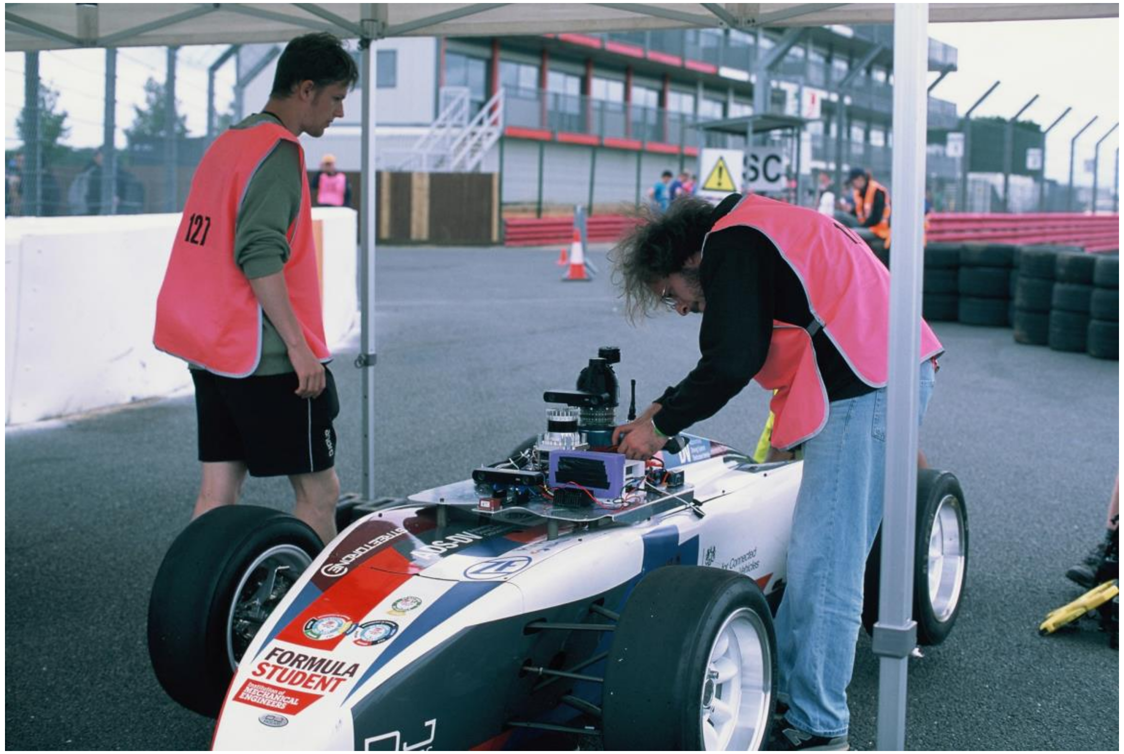
FS-AI – Difficulties

What prevents us from going fast?

What if we see a turn too late?

What do we need?

- Map of the track
- Position in the map

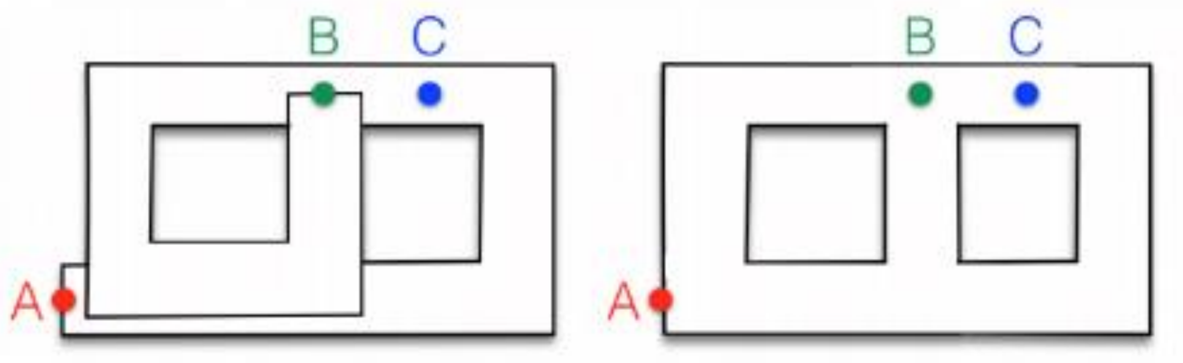


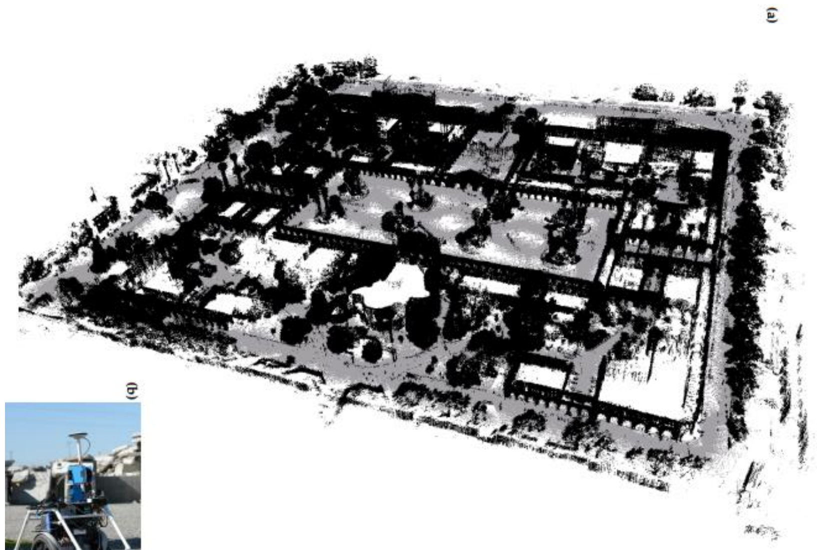
Mapping

Mapping – What is a map?

Produce a map of the track

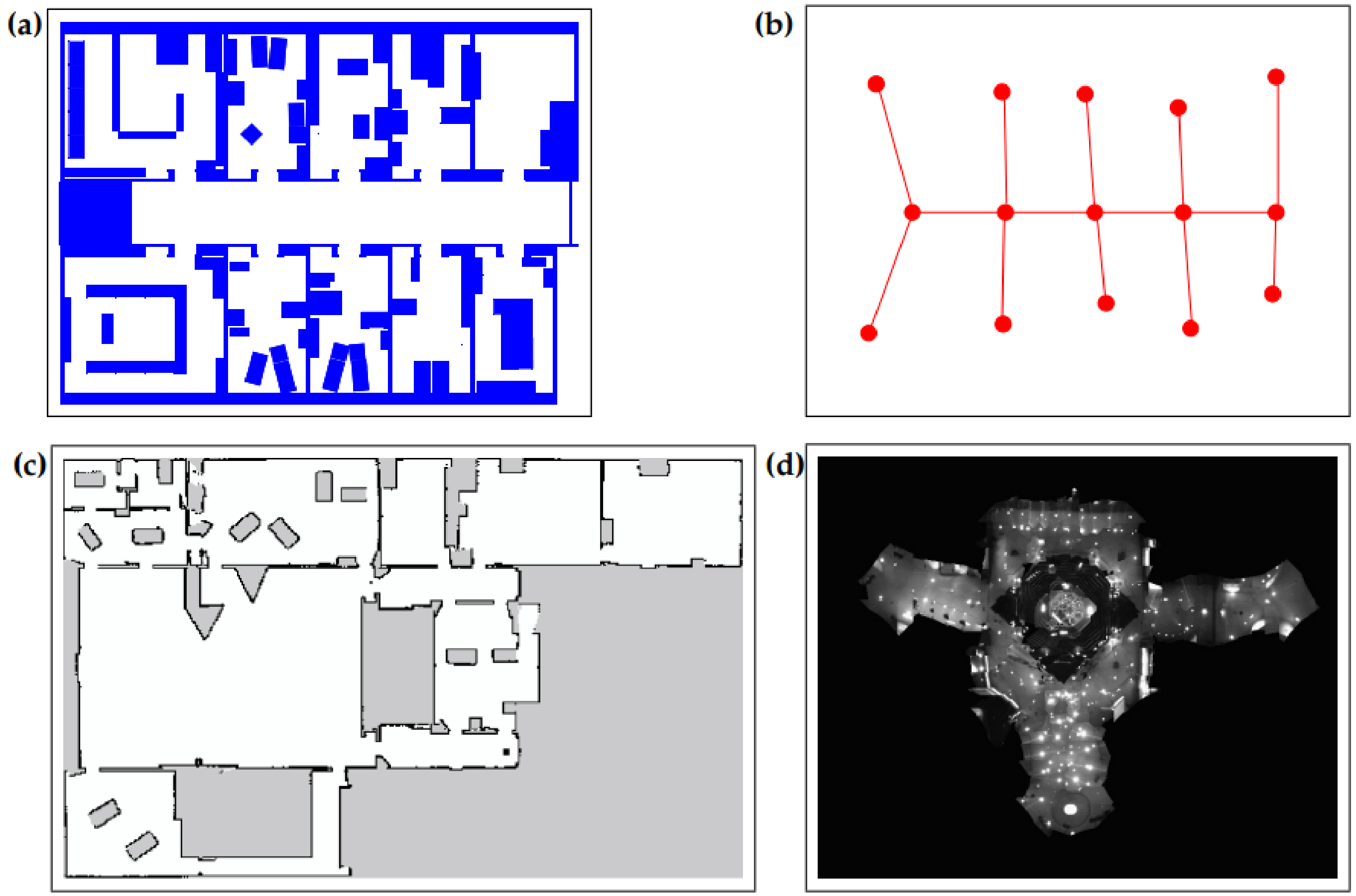
Representation of the environment





Mapping – Example types of map

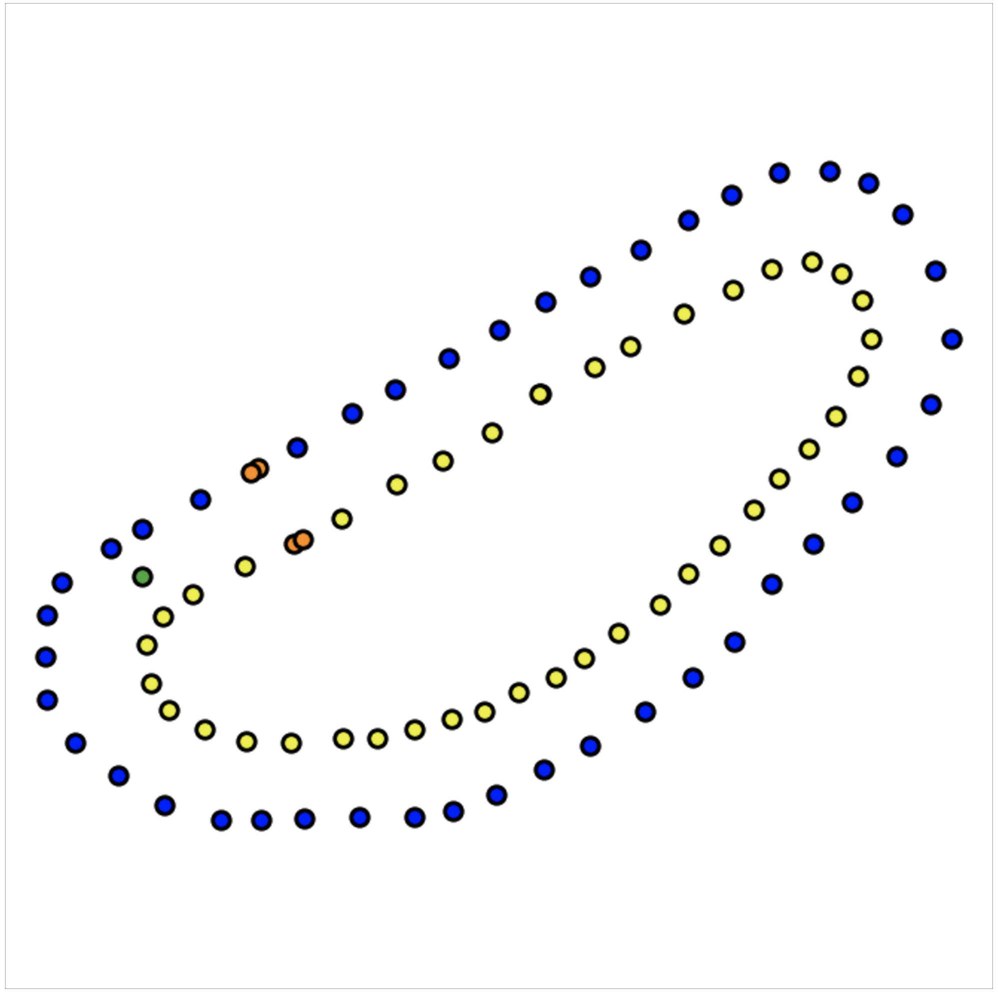
- Topological map (a)
- Landmark-based map (b)
- Occupancy grid map (c)
- Image map (d)



Mapping – Map for SLAM

- Landmark-based map

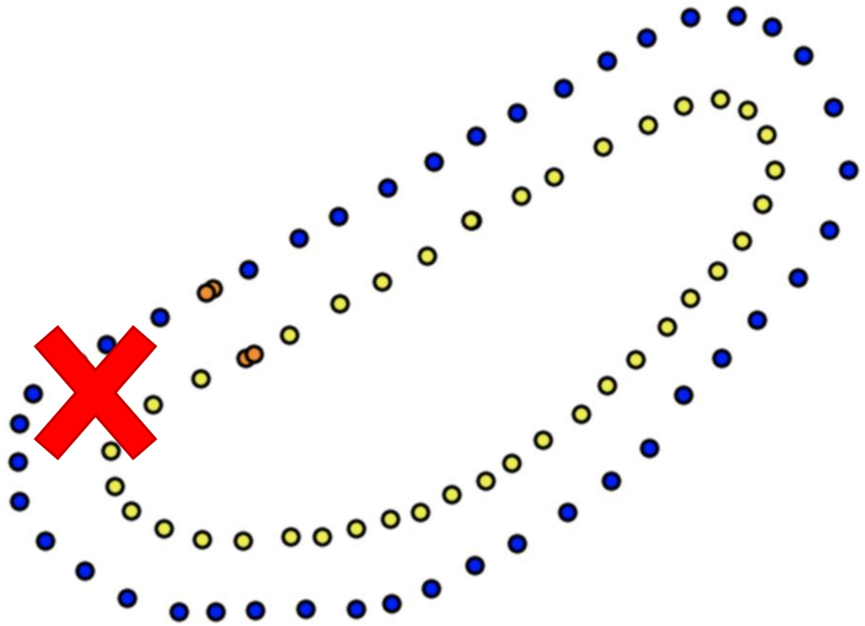
- Each cone is a landmark
- Static map

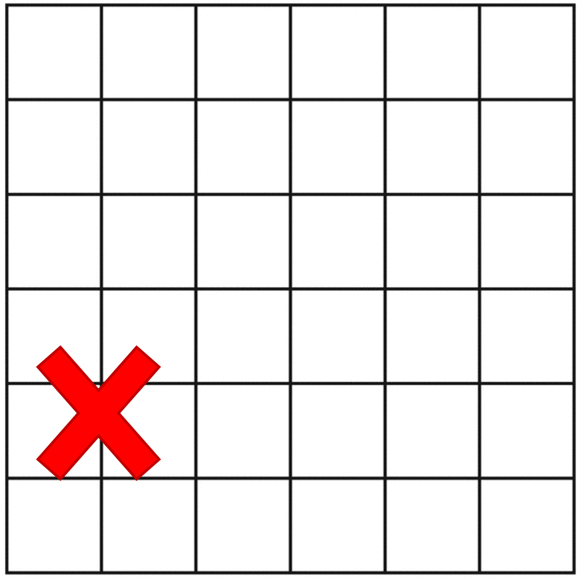


Localisation

Localisation

- Figure out where the robot is in the map, i.e. its position



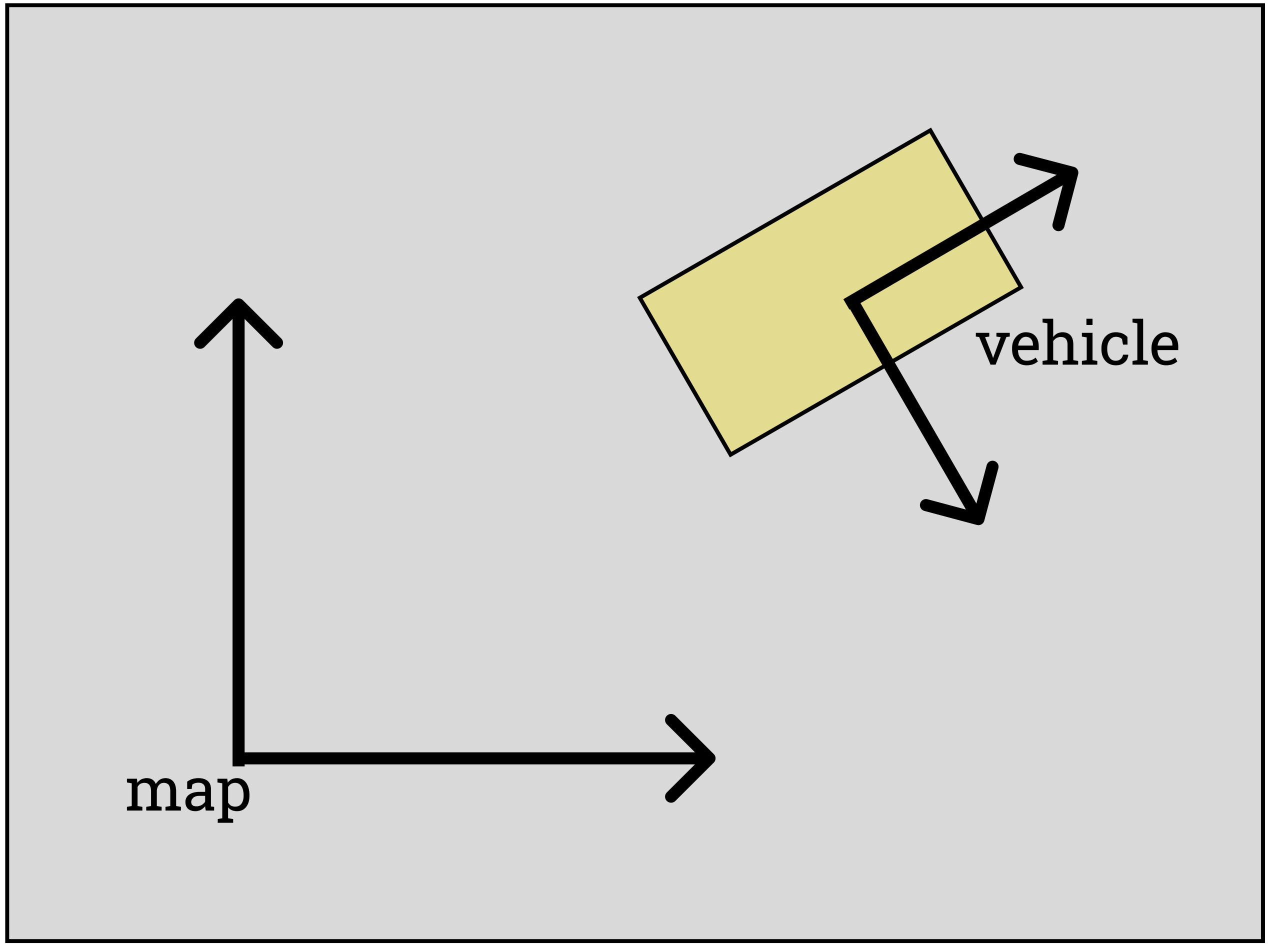


Localisation – Frames of Reference

There are two frames of importance:
1. Map frame
2. Vehicle frame

Map frame is the global coordinate system.

Vehicle frame is a coordinate system attached to the vehicle.

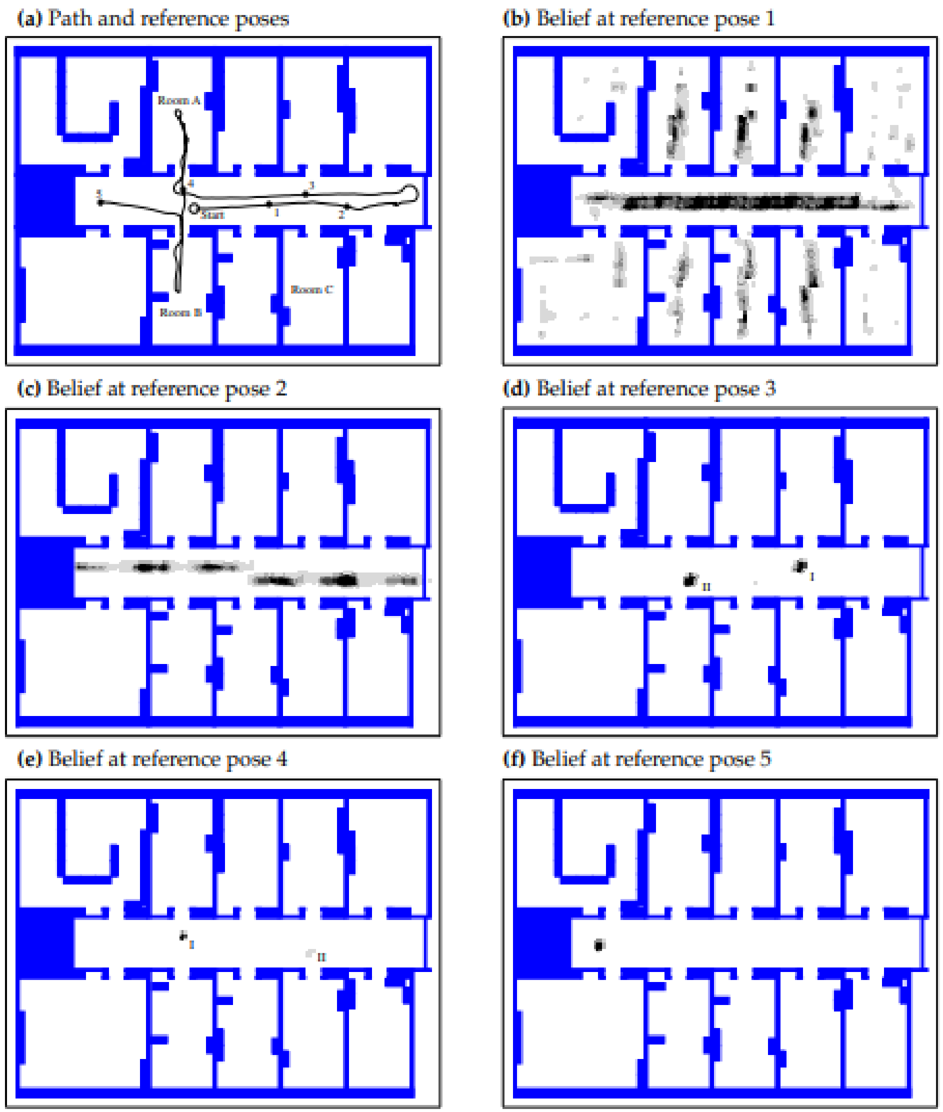


Localisation – Taxonomy

- **Tracking** – initial position known
- **Global localisation** – initial position unknown
- **Kidnapped robot problem**

What type is EUFS? *Tracking*

Static vs dynamic environments

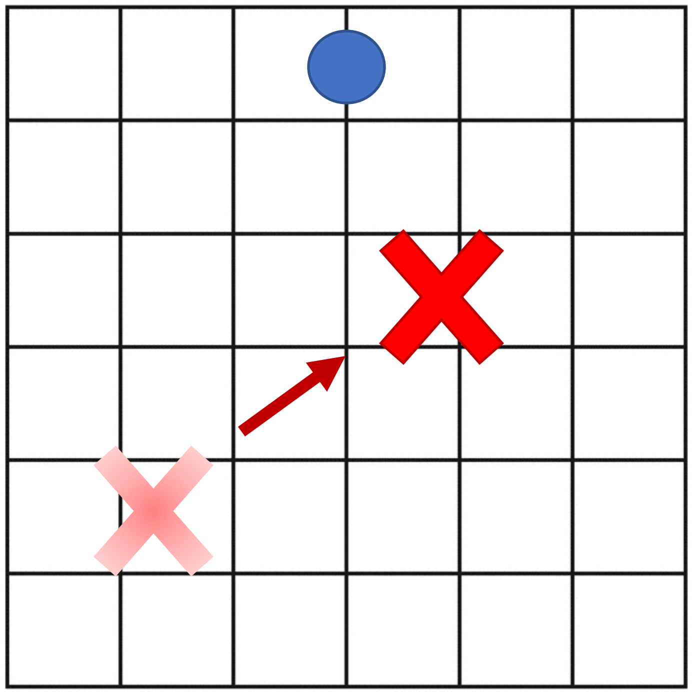


Localisation

"Where are the landmarks I saw in the previous timestep and how did I move to get here?"

"Where am I in relation to that landmark?"

Mobile robot localisation



Localisation – Algorithms

1. Extended Kalman Filter
2. Particle Filters

Algorithm Bayes_filter($bel(x_{t-1}), u_t, z_t$):

for all x_t do

$\overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx$

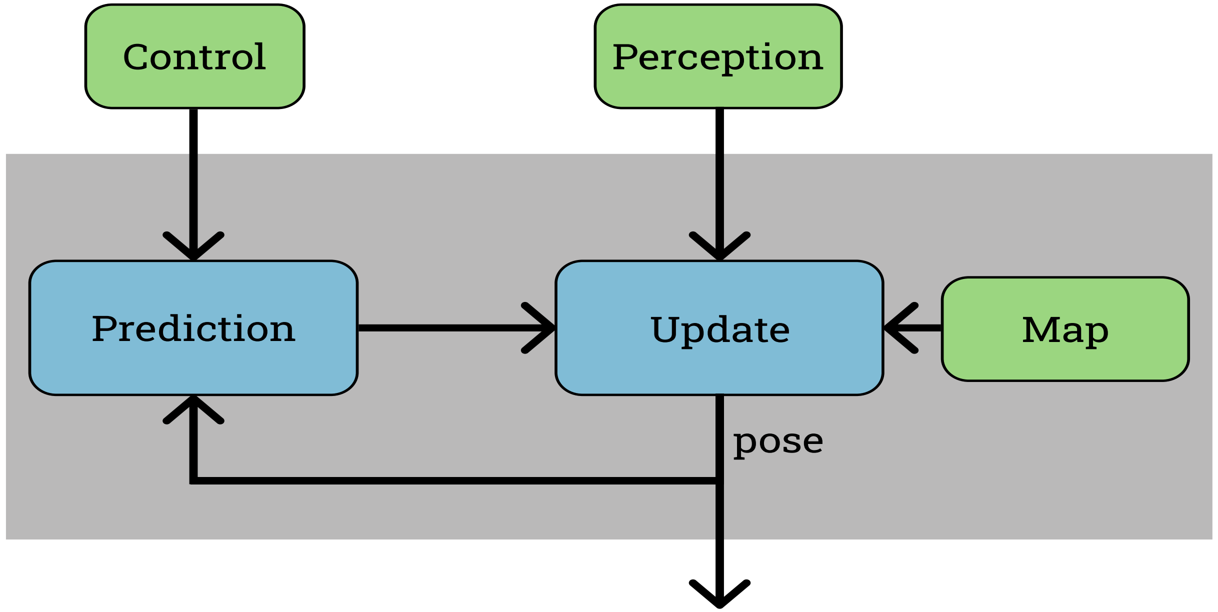
$bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$

endfor

return $bel(x_t)$

Require:

- Motion model
- Measurement model



Localisation – Particle Filter

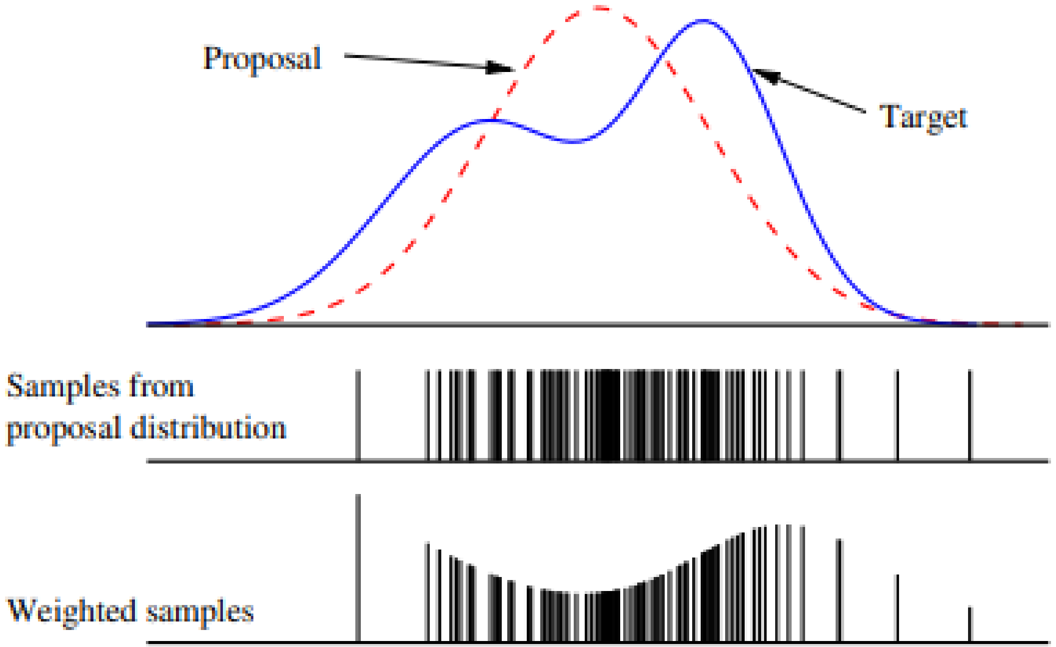
EKF represents pose using a Multivariate Gaussian distribution.

Three steps involved in a particle filter:

1. Update

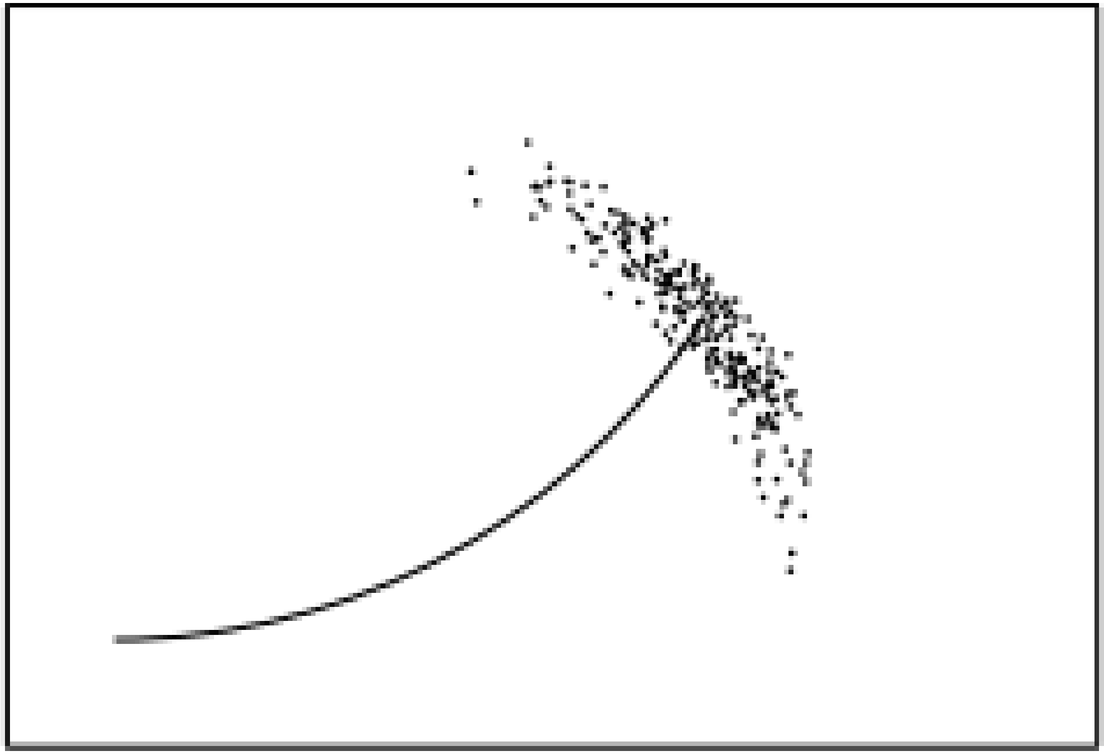
2. Weight

3. Resample



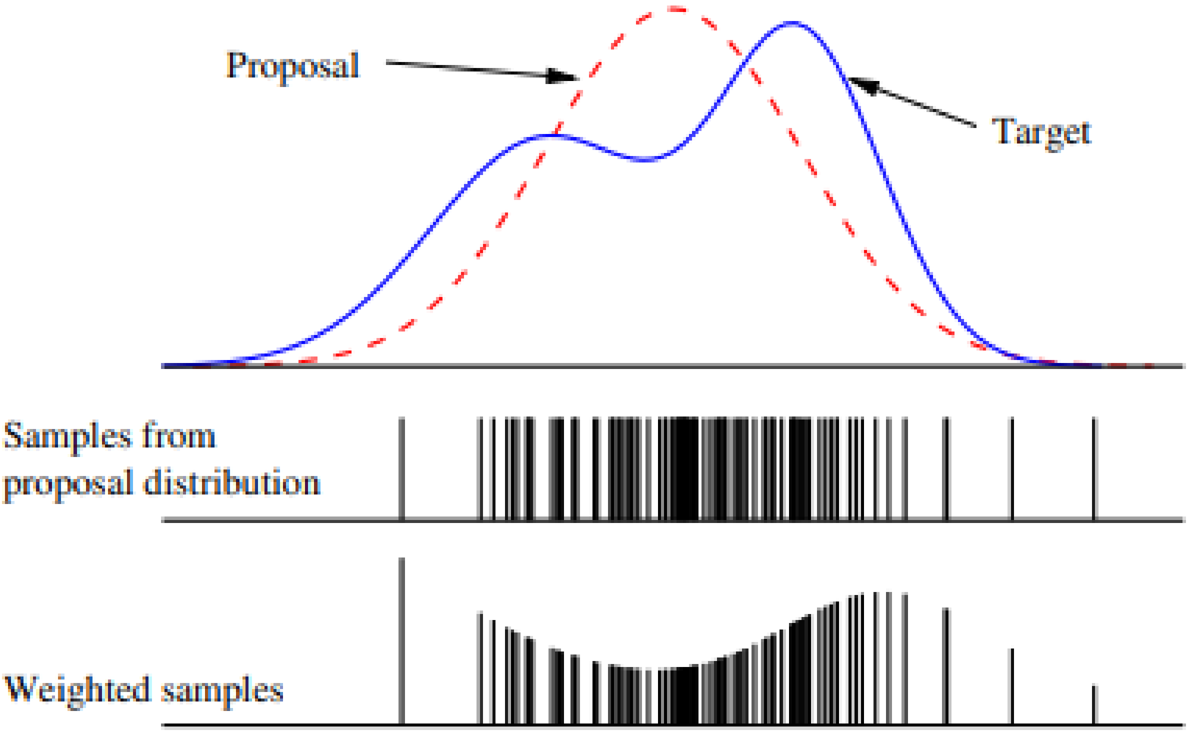
Localisation – Particle Filter

Update



Localisation – Particle Filter

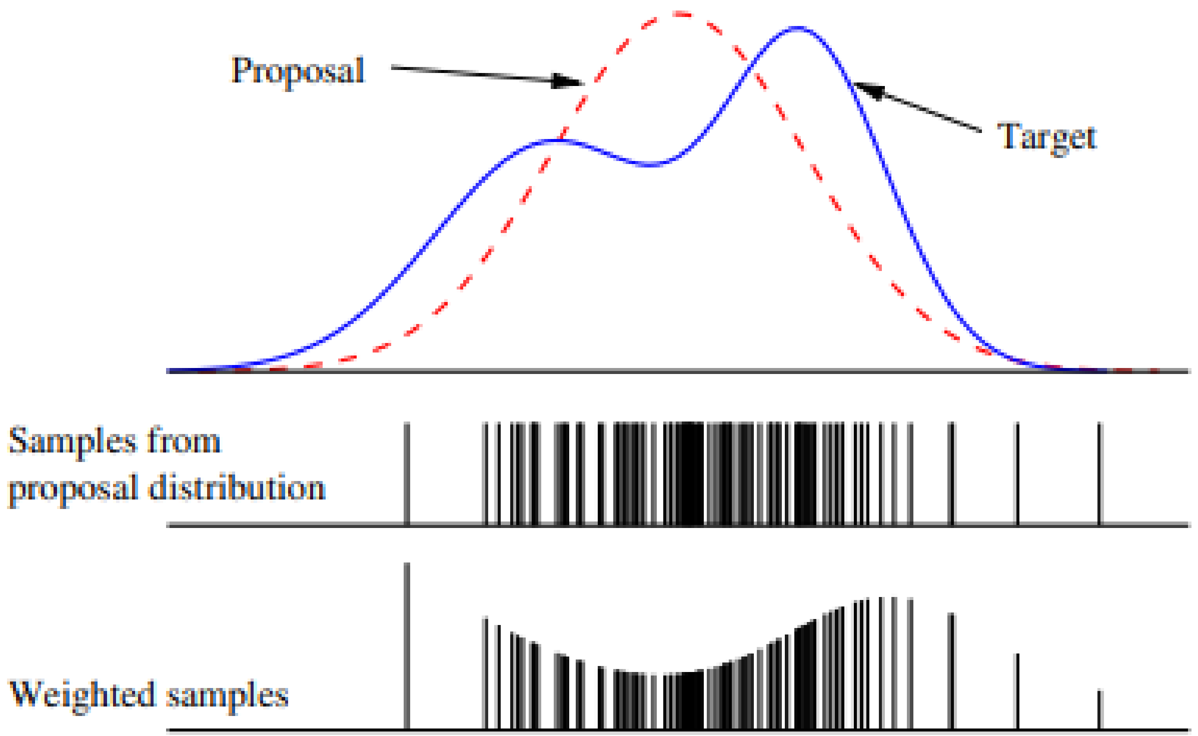
Weight



Localisation – Particle Filter

Resampling

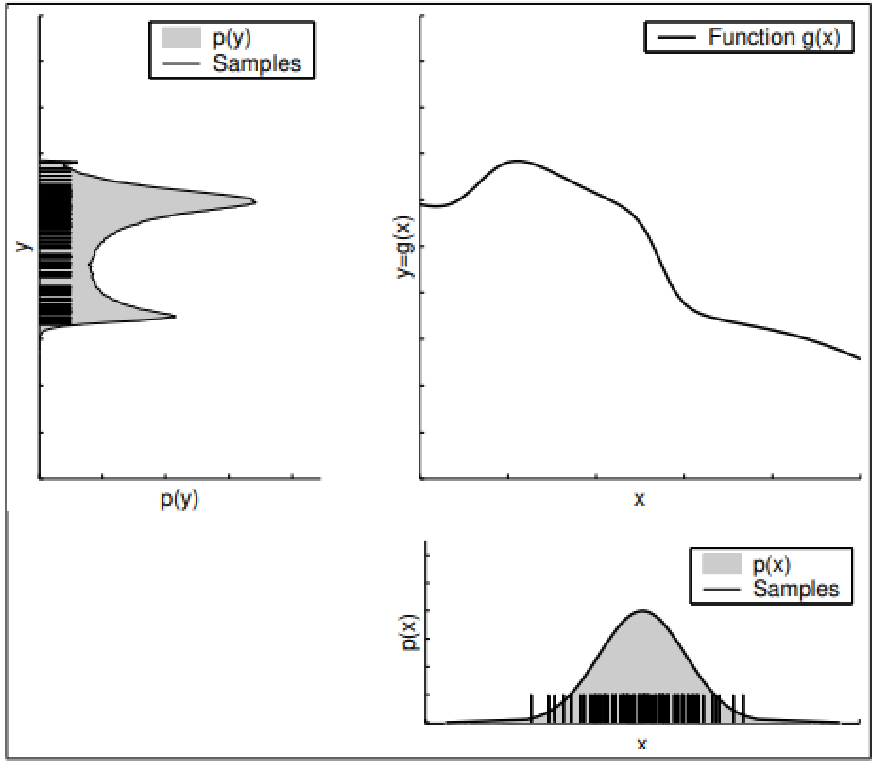
The process repeats!



Localisation – Particle Filter

Important considerations for particle filters:

- **Particle diversity**



Pros:

- Can easily model multi-modal distributions
- Easily parallelisable

Cons:

- The number of particles required is exponential in the dimensions of the state

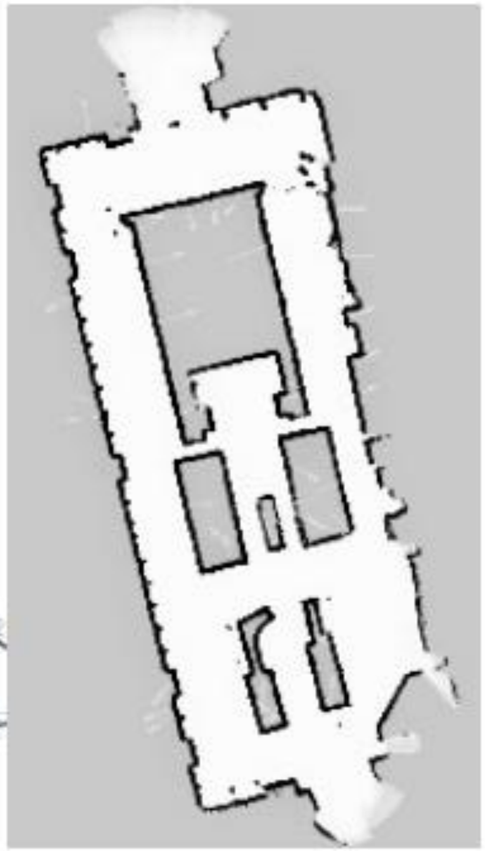
Challenges

Mapping – Challenges

Creating a map is easy if exact position of the robot is known.

Why can't we just use velocity to figure out the position of the robot?

Other potential solutions?



Mapping – Challenges

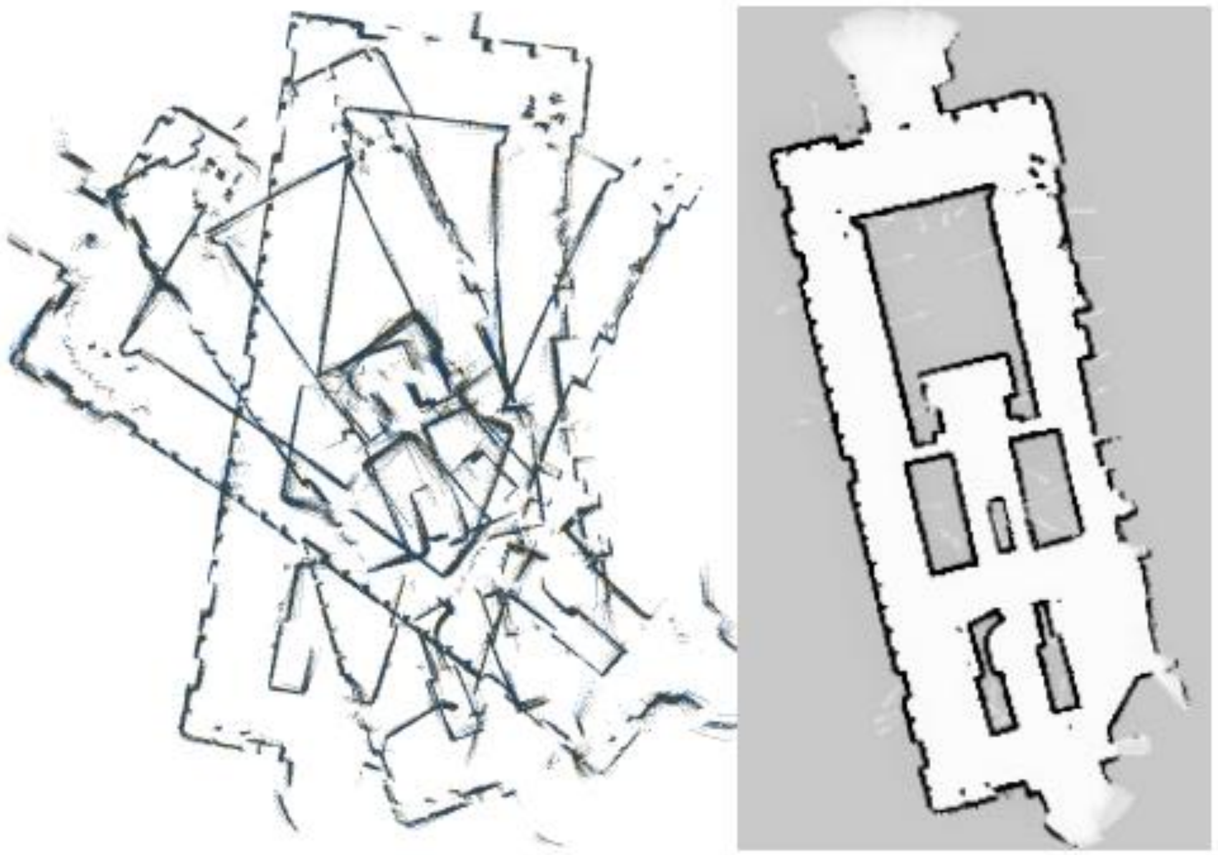
Creating a map is easy if exact position of the robot is known.

Why can't we just use velocity to figure out the position of the robot?

Other potential solutions?

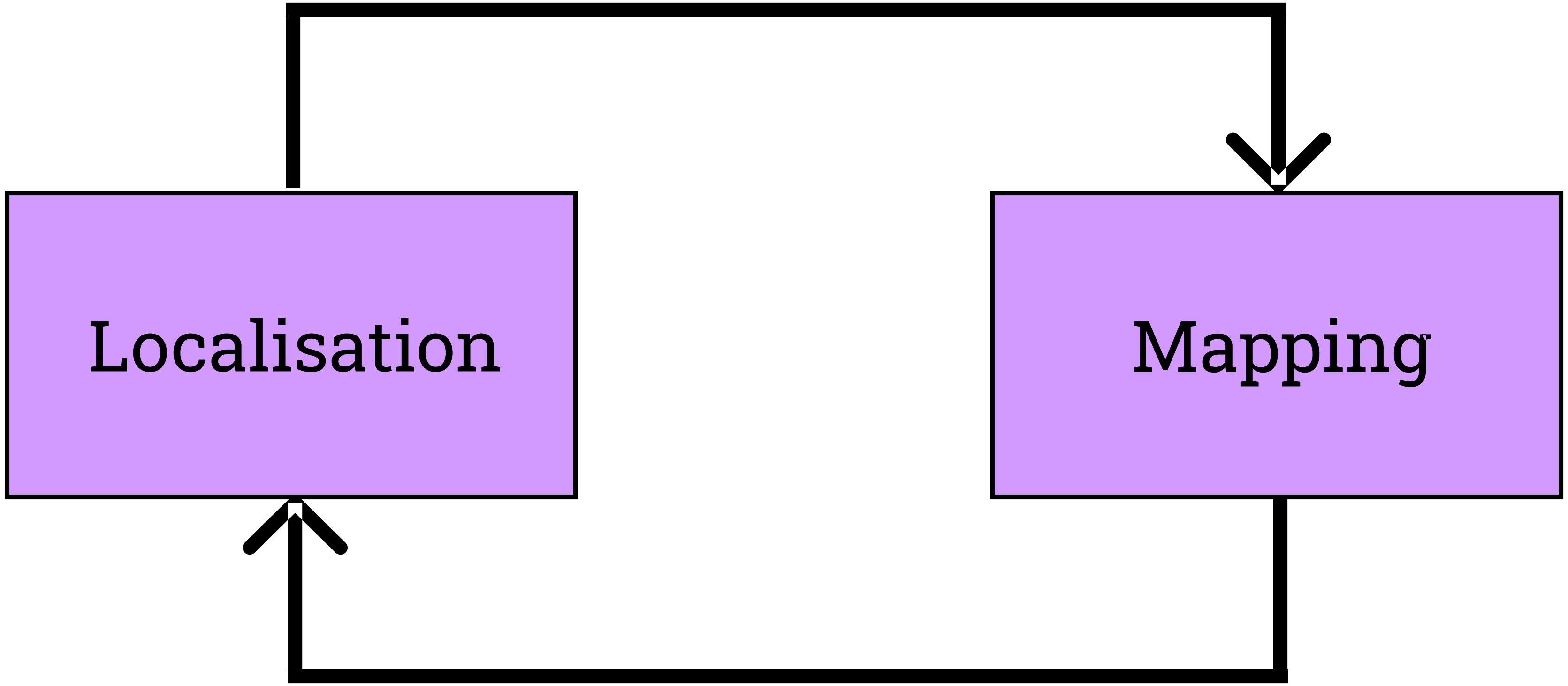
1. GPS (unless denied GPS environment)
2. **Localisation**

It would be nice if there were no reliability concerns!



SLAM

The Chicken and Egg Problem



SLAM – Simultaneous Localisation and Mapping

SLAM

SLAM

Simultaneous Localisation and Mapping (SLAM) looks to build a map of the environment whilst determining the location of the robot within the map.

Focus: probabilistic formulation of the SLAM problem.

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$$

Time-update

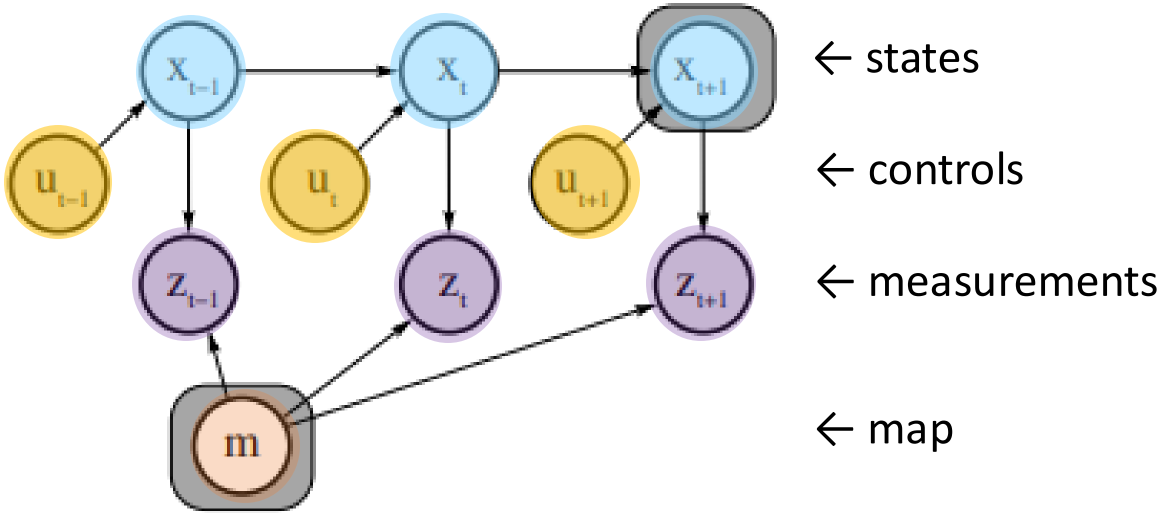
$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \int P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \times P(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1} \quad (4)$$

Measurement Update

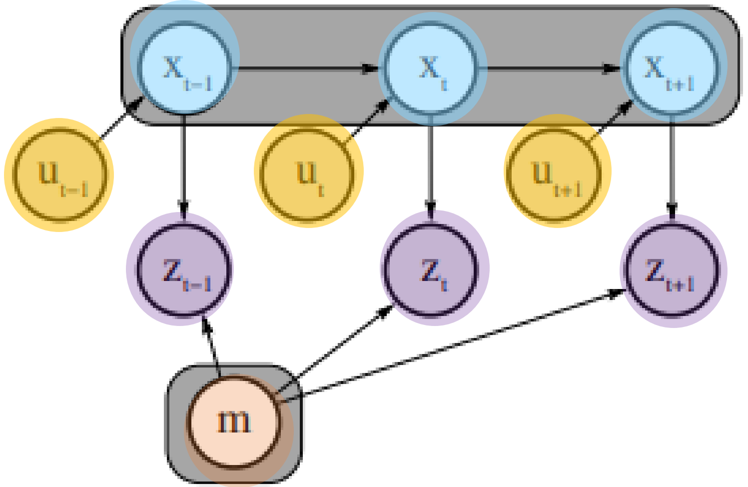
$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \frac{P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0)}{P(\mathbf{z}_k | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k})} \quad (5)$$

SLAM – Online vs Full

Online SLAM

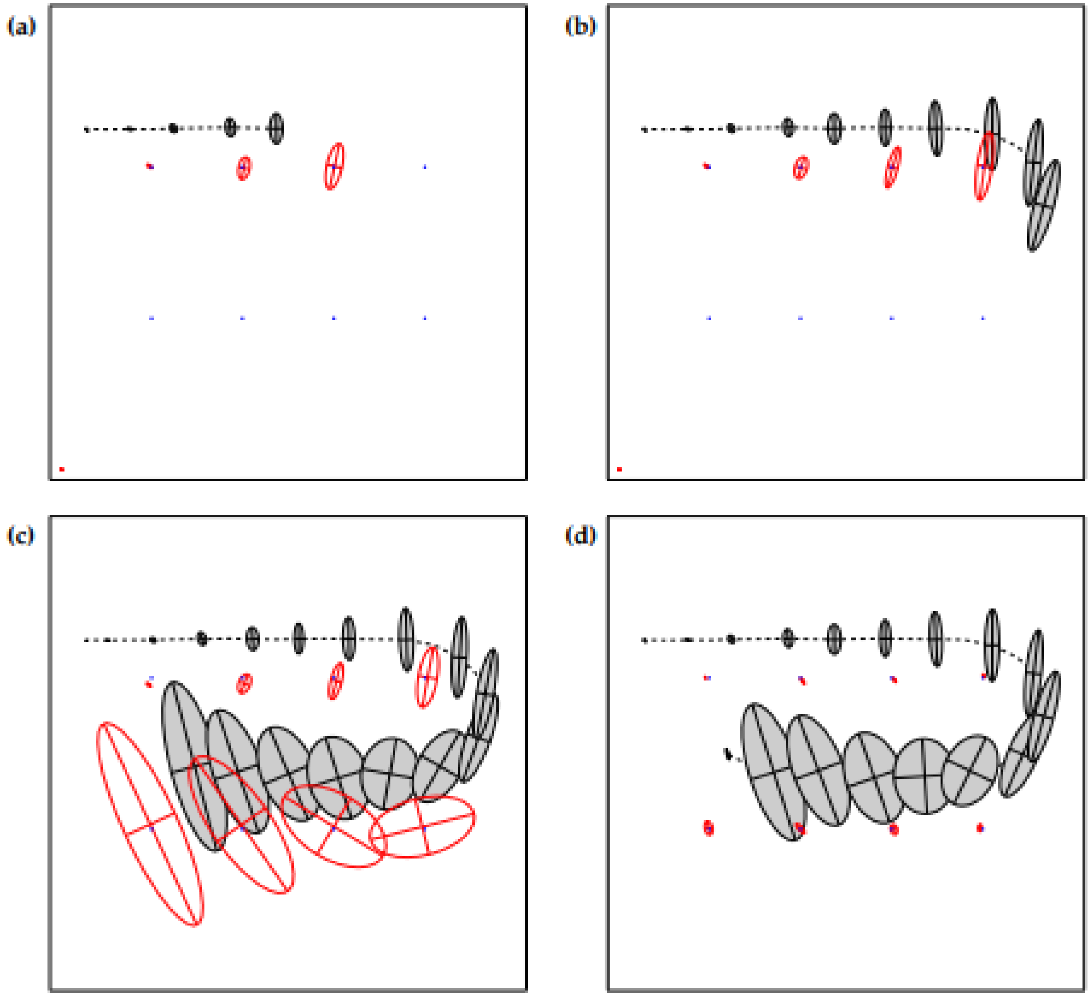


Full SLAM

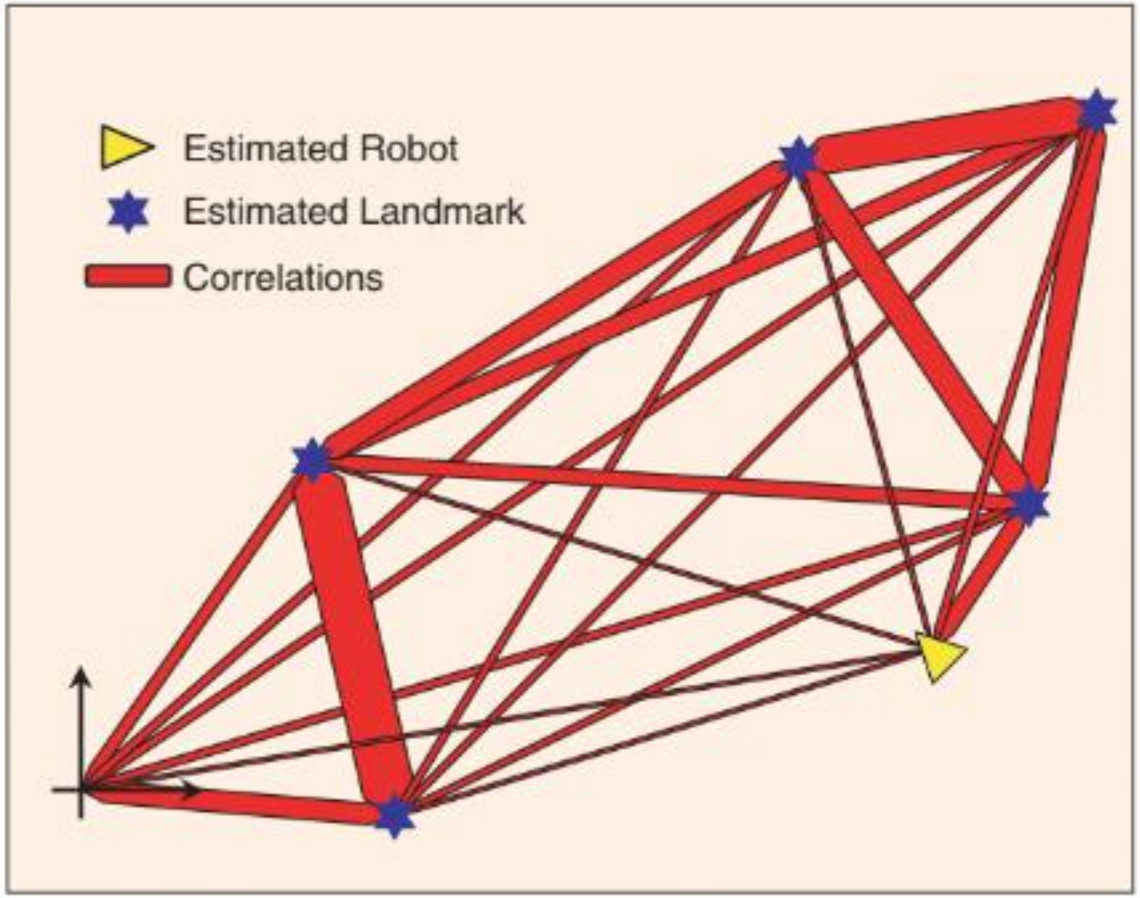


SLAM – Conceptual components

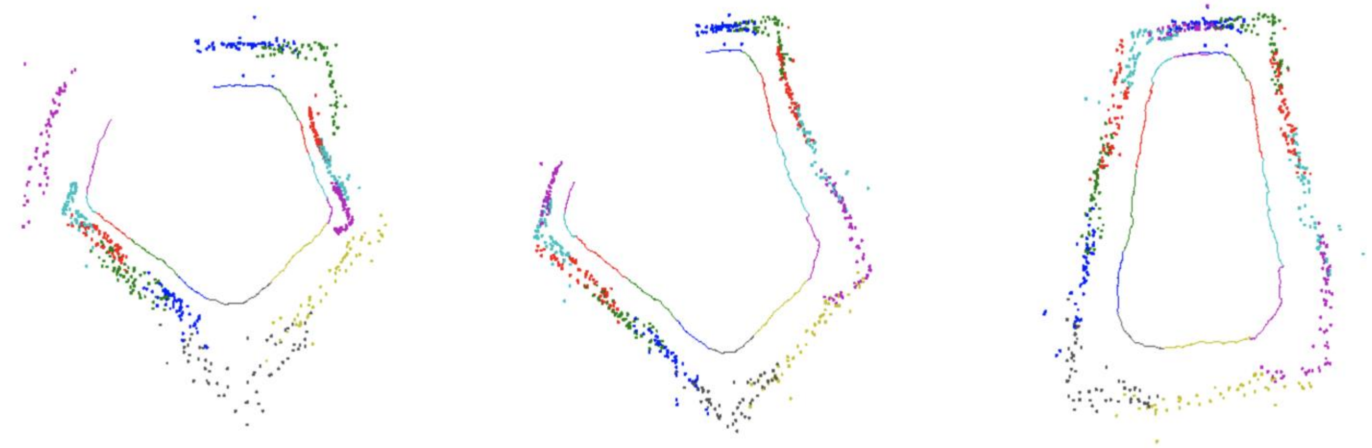
SLAM - Loop Closure



SLAM - Loop Closure



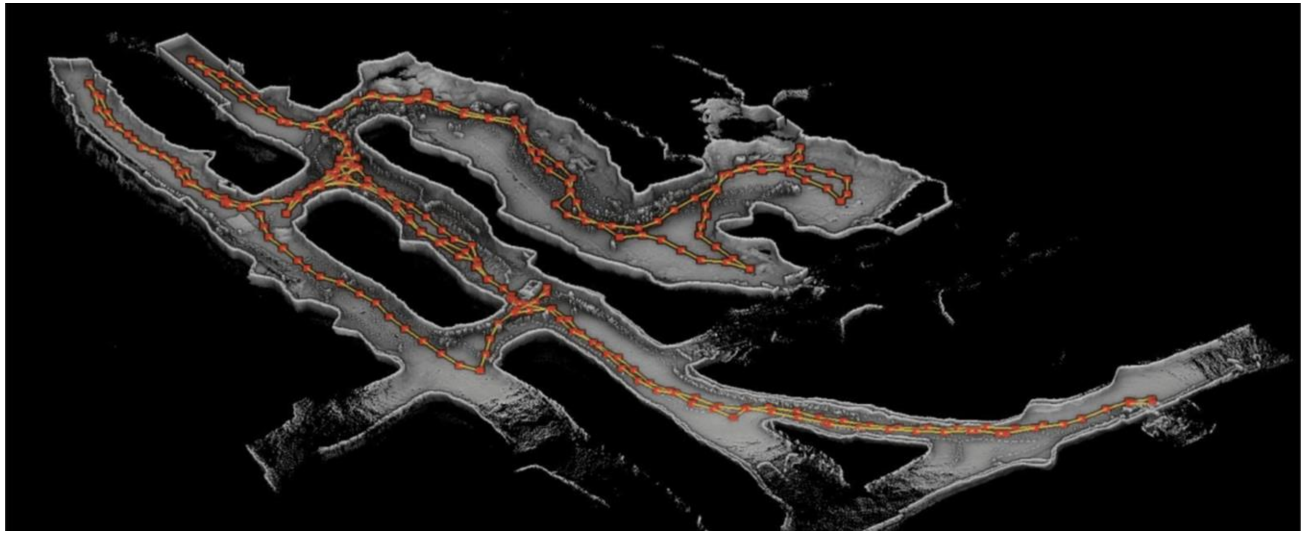
SLAM - Loop Closure



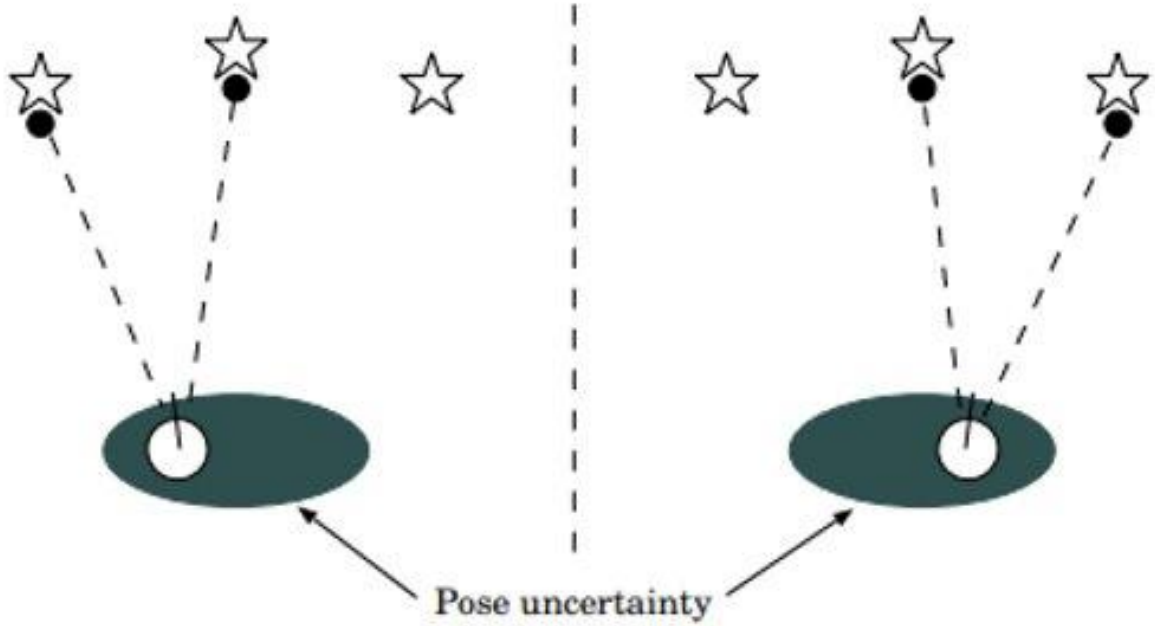
Local maps obtained

(b) Local maps auto-scaled

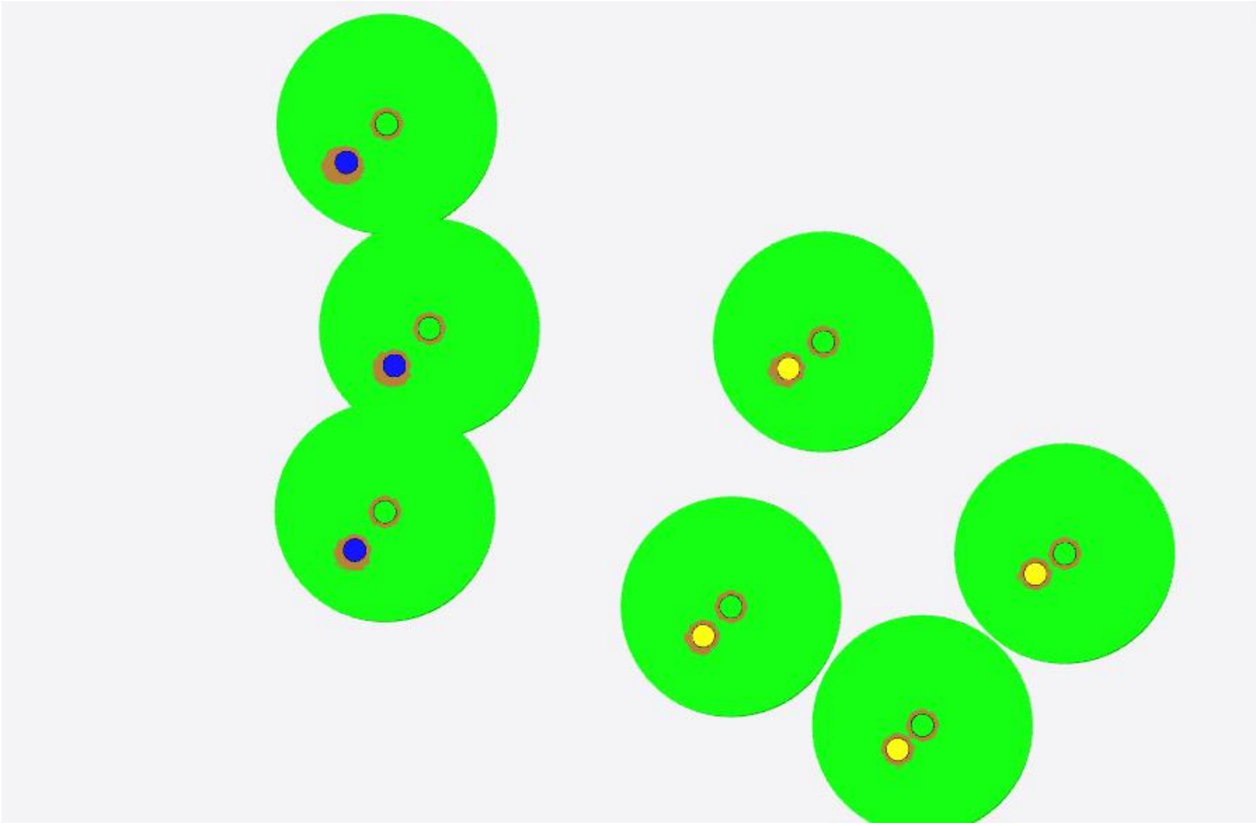
(c) After loop closing



SLAM - Data Association



SLAM - Data Association



FastSLAM

FastSLAM 1.0

Particle filter to predict robot pose

EKFs to update landmarks

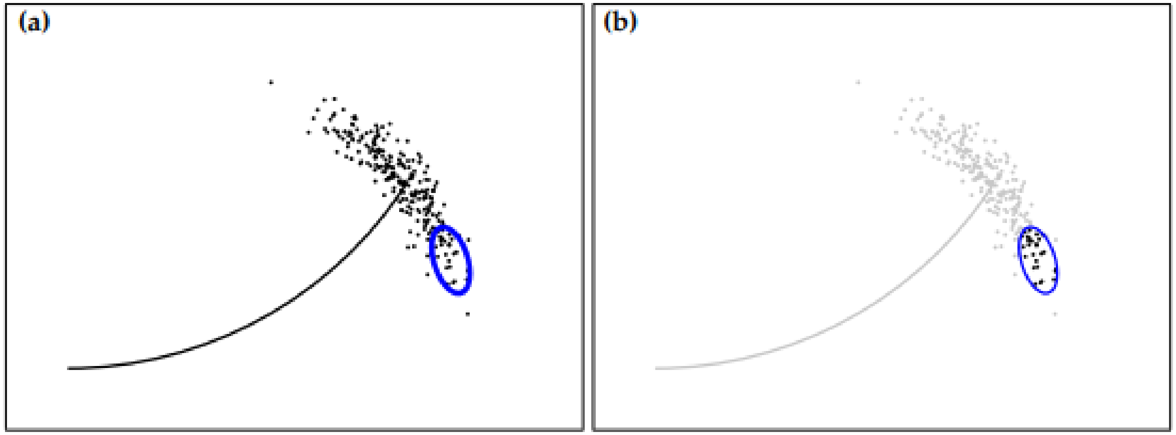
- One for each particle, for each landmark

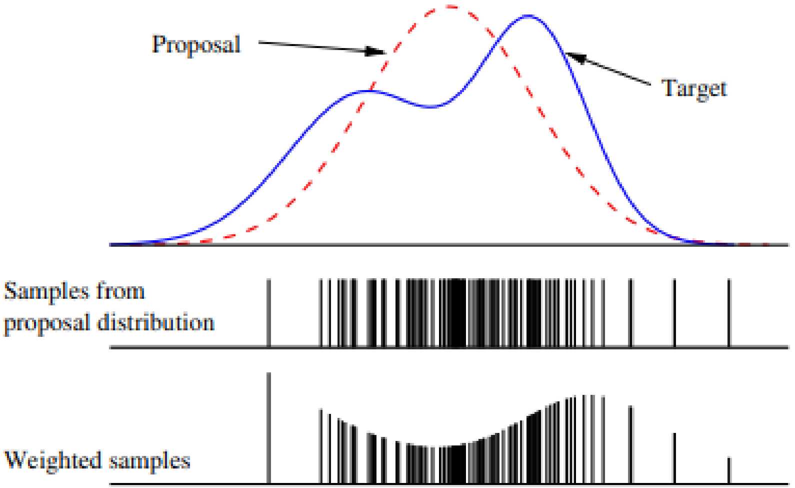
	robot path	feature 1	feature 2	…	feature N
Particle $k=1$	$x_{1:t}^{[1]} = \{(x\ y\ \theta)^T\}_{1:t}^{[1]}$	$\mu_1^{[1]}, \Sigma_1^{[1]}$	$\mu_2^{[1]}, \Sigma_2^{[1]}$	…	$\mu_N^{[1]}, \Sigma_N^{[1]}$
Particle $k=2$	$x_{1:t}^{[2]} = \{(x\ y\ \theta)^T\}_{1:t}^{[2]}$	$\mu_1^{[2]}, \Sigma_1^{[2]}$	$\mu_2^{[2]}, \Sigma_2^{[2]}$	…	$\mu_N^{[2]}, \Sigma_N^{[2]}$
		$\vdots$			
Particle $k=M$	$x_{1:t}^{[M]} = \{(x\ y\ \theta)^T\}_{1:t}^{[M]}$	$\mu_1^{[M]}, \Sigma_1^{[M]}$	$\mu_2^{[M]}, \Sigma_2^{[M]}$	…	$\mu_N^{[M]}, \Sigma_N^{[M]}$

FastSLAM 2.0

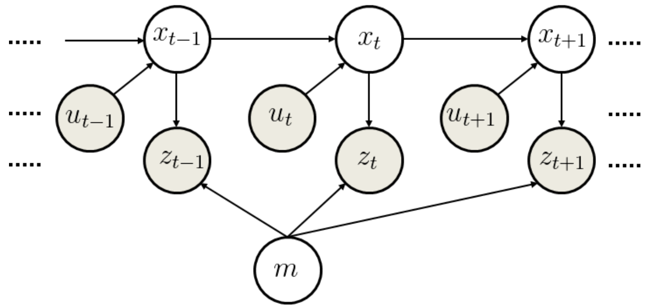
FastSLAM 1.0 does the prediction step purely using the control information.

FastSLAM 2.0 considers perception information in the prediction step.





FastSLAM – Algorithm Steps



REPEAT M TIMES

- Retrieval
- Prediction
- Measurement update
- Importance weight

THEN DO

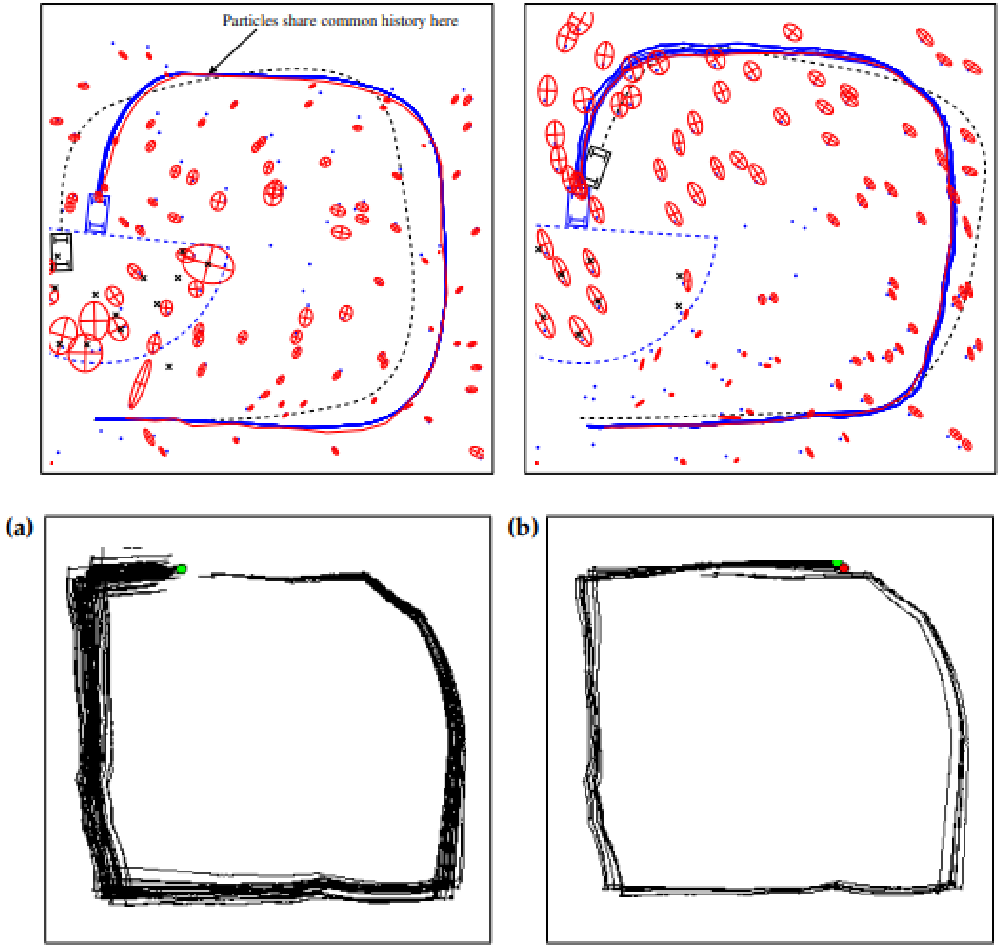
- Resampling (M particles)

	robot path	feature 1	feature 2	…	feature N
Particle $k = 1$	$x_{1:t}^{[1]} = \{(x\ \ y\ \ \theta)^T\}_{1:t}^{[1]}$	$\mu_1^{[1]}, \Sigma_1^{[1]}$	$\mu_2^{[1]}, \Sigma_2^{[1]}$	…	$\mu_N^{[1]}, \Sigma_N^{[1]}$
Particle $k = 2$	$x_{1:t}^{[2]} = \{(x\ \ y\ \ \theta)^T\}_{1:t}^{[2]}$	$\mu_1^{[2]}, \Sigma_1^{[2]}$	$\mu_2^{[2]}, \Sigma_2^{[2]}$	…	$\mu_N^{[2]}, \Sigma_N^{[2]}$
		$\vdots$			
Particle $k = M$	$x_{1:t}^{[M]} = \{(x\ \ y\ \ \theta)^T\}_{1:t}^{[M]}$	$\mu_1^{[M]}, \Sigma_1^{[M]}$	$\mu_2^{[M]}, \Sigma_2^{[M]}$	…	$\mu_N^{[M]}, \Sigma_N^{[M]}$

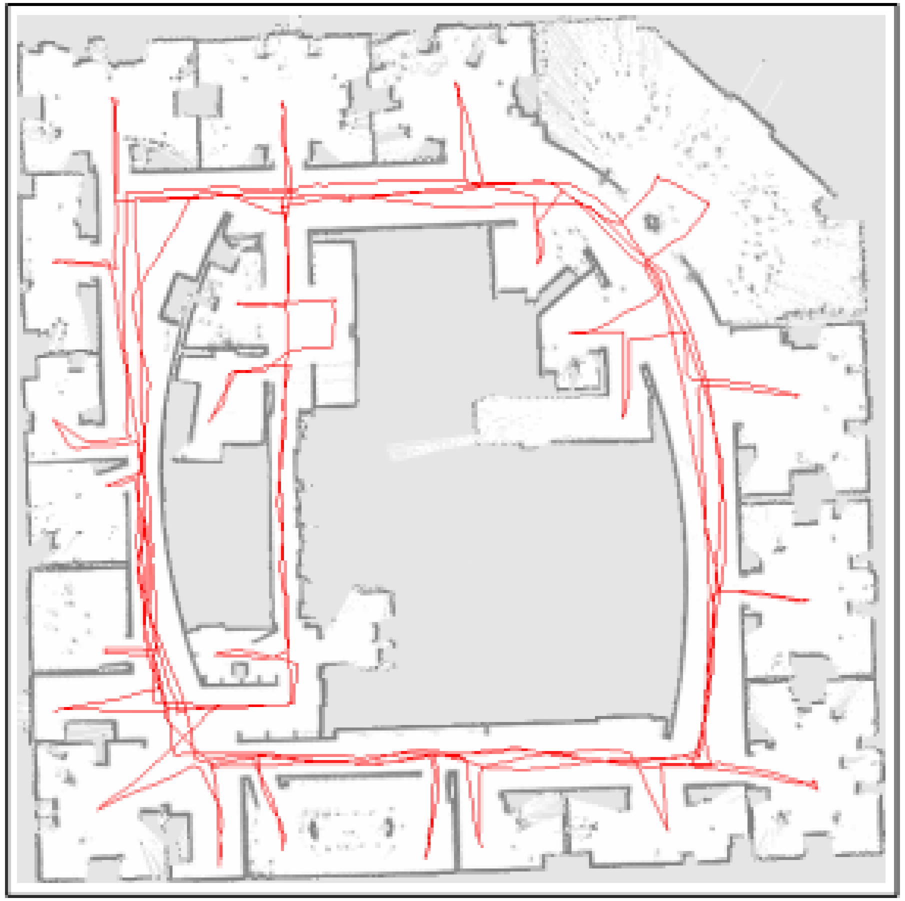
FastSLAM - Particle Resampling

FastSLAM - Loop Closure

When loop closure occurs, changes in the map can only occur up to the common ancestor!



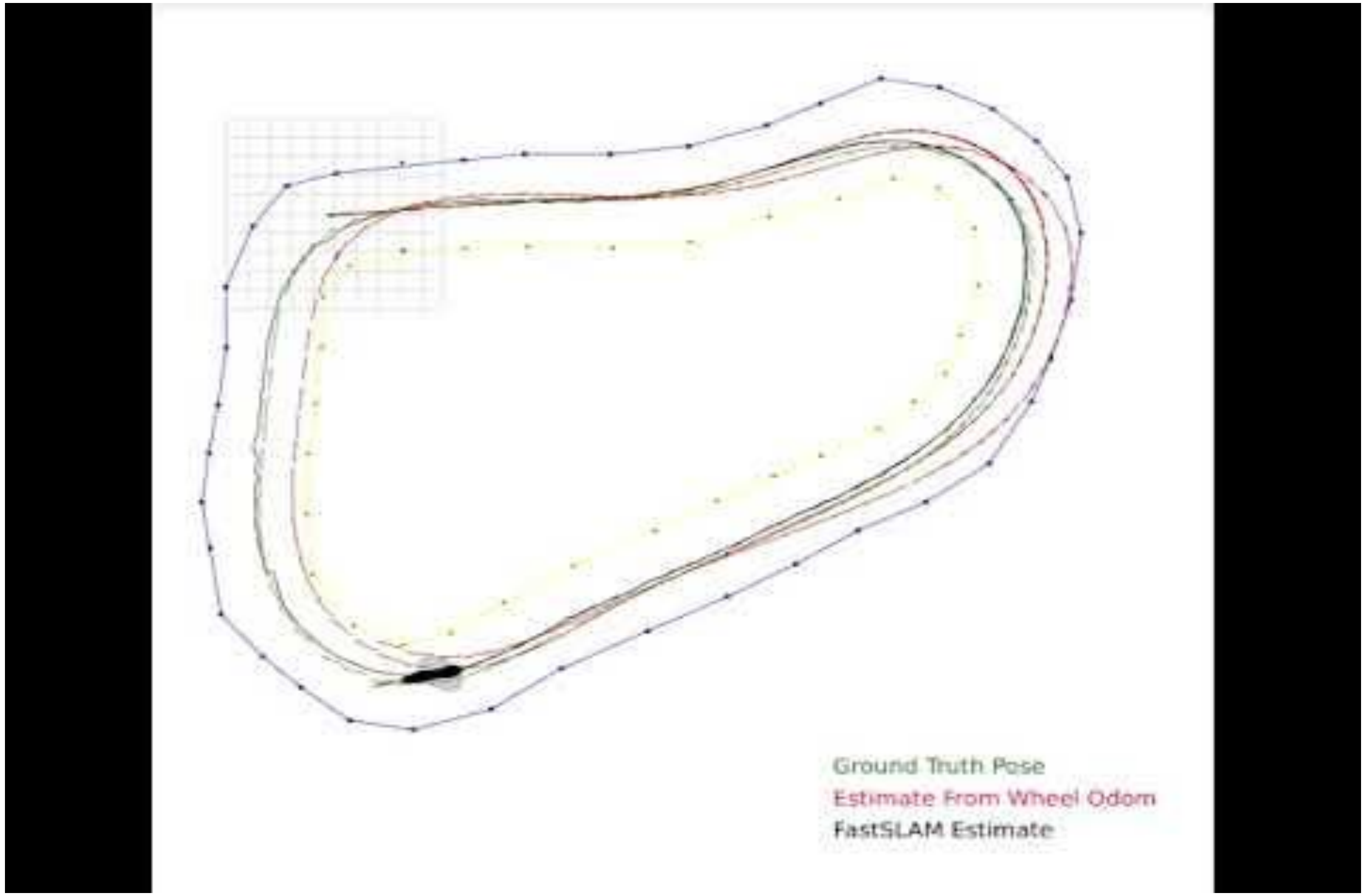
FastSLAM for Grid Maps



FastSLAM For EUFS

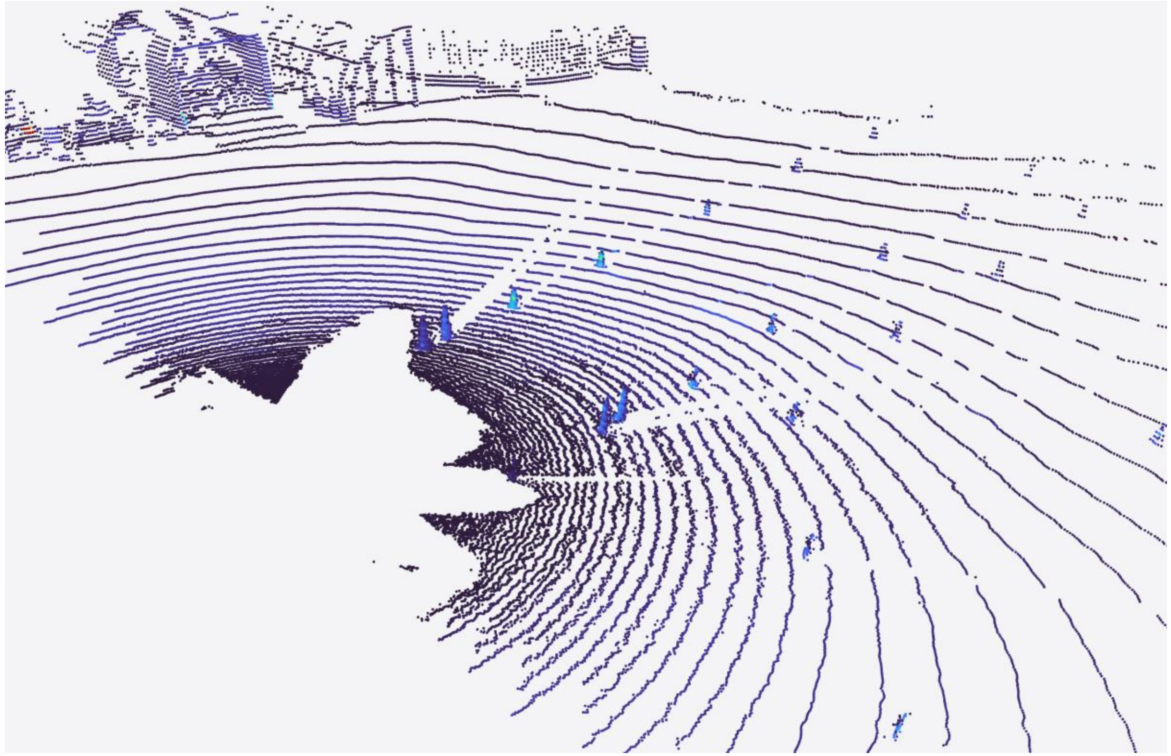
Why we like FastSLAM for EUFS:

1. Can create map in real time
2. Lower computational complexity
3. Can easily be parallelised
4. Lots of other teams use it!
5. Easily switch SLAM -> Localisation



https://www.youtube.com/watch?v=d6TwUduuY8o

FastSLAM for EUFS - What is a LiDAR?



FastSLAM For EUFS – Troubleshooting

FSUK 24

LiDAR cones ->

<- map

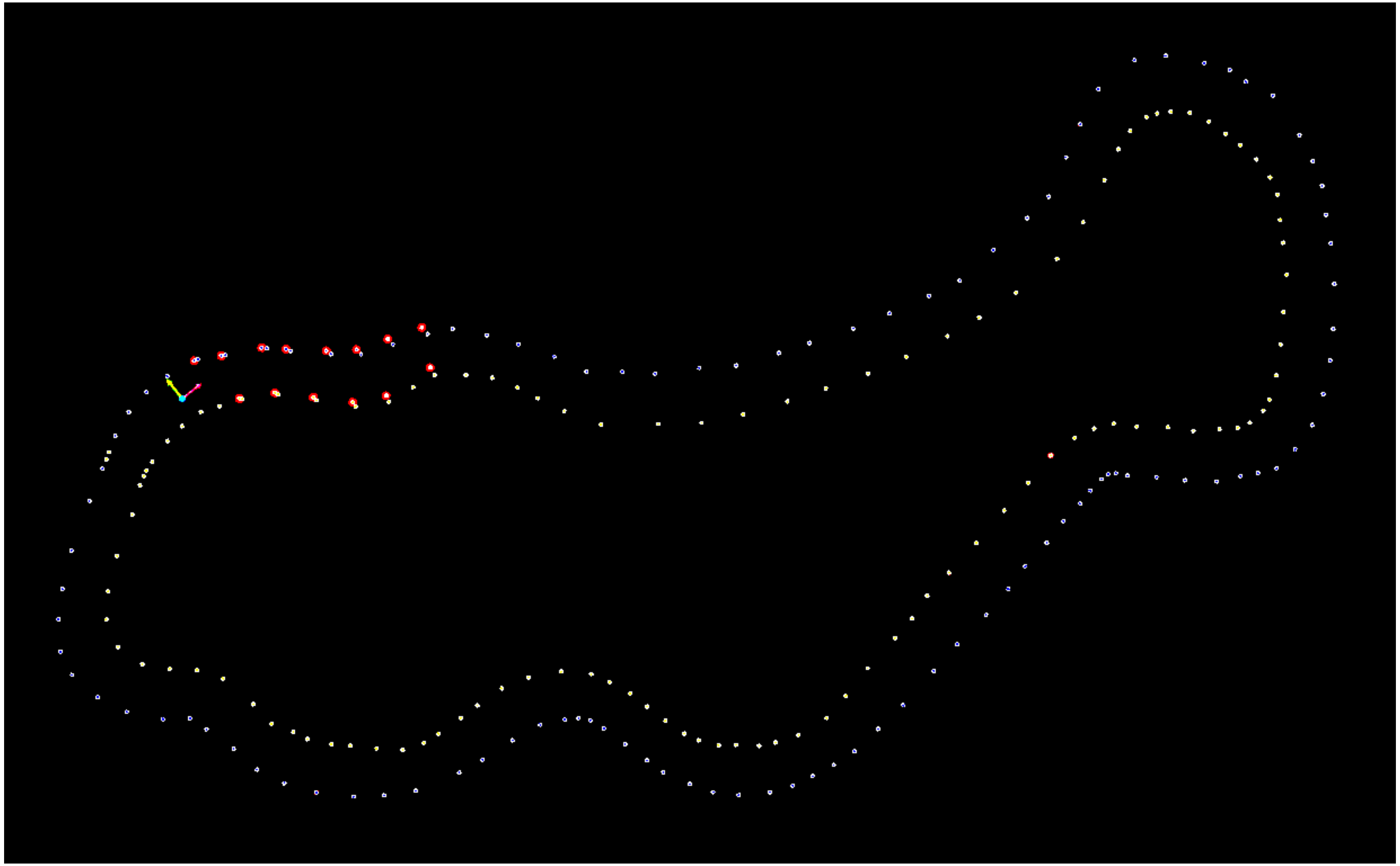
lidarconespack170.webm

FastSLAM 2.0 in Action at FS UK 23

FastSLAM 2.0 in Action at FS UK 23

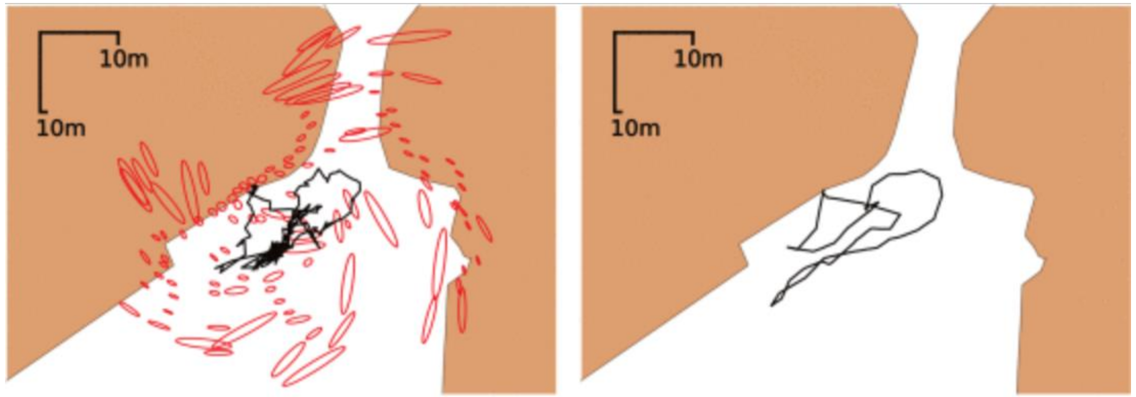
- Best lap of trackdrive
- SLAM map

- Do you think it resembles what you saw in the video?



FastSLAM 2.0 - Applications

NASA Viper:
- Polar region of moon in 2023
- First NASA missions to use ROS



(a) Estimated robot path. (b) Ground truth.

Mission Robotics Submersible Robot
- Coral reef mapping / species identification
- Mapping least known part of Earth
- Use wall features to perform SLAM

FastSLAM - Question

Conditioning on the most recent pose instead of the entire path is sufficient.

True or False?

FastSLAM - Question

False.

Conditioning on the most recent pose instead of the entire path is insufficient, as dependencies may arise through previous poses.

EKF SLAM

EKF SLAM

SLAM using Extended Kalman Filters was the first SLAM algorithm published.

$$\begin{bmatrix} x \\ y \\ \theta \\ m^1_x \\ m^1_y \\ \vdots \end{bmatrix} \begin{bmatrix} & \cdots & \\ \vdots & \ddots & \\ & & \end{bmatrix}$$

EKF SLAM - Limitations

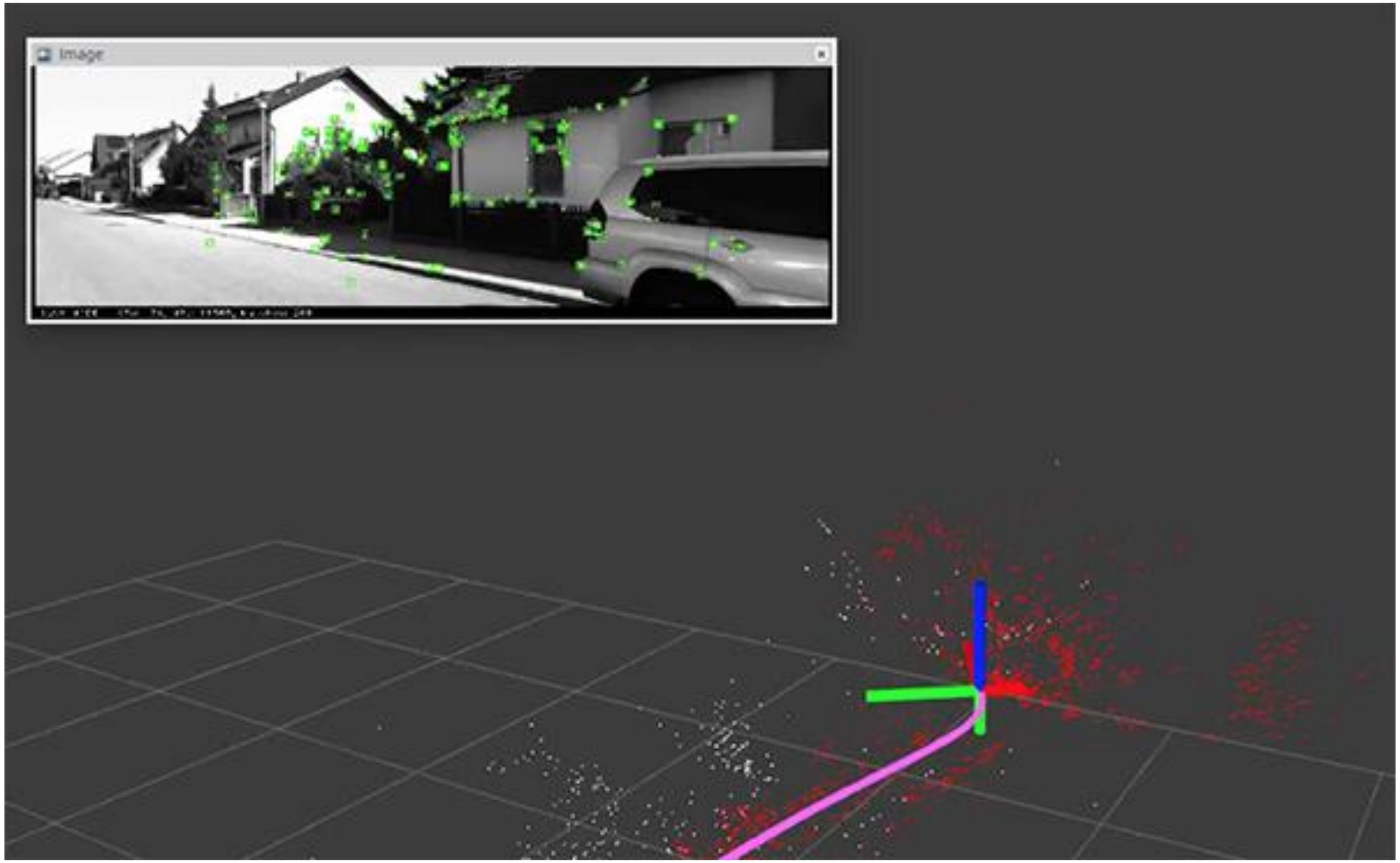
Requires:

- Enormous update complexity
- Brittle to incorrect data association due to linearisation

Keeps track of all uncertainty!

$$
\begin{bmatrix} x \\ y \\ \theta \\ m^1_x \\ m^1_y \\ \vdots \end{bmatrix}
\begin{bmatrix} & \cdots & \\ \vdots & \ddots & \\ & & \end{bmatrix}
$$

EKF SLAM - Applications



EKF SLAM - Question

EKF SLAM applies the extended Kalman filter to the full SLAM problem.

True or False?

EKF SLAM - Question

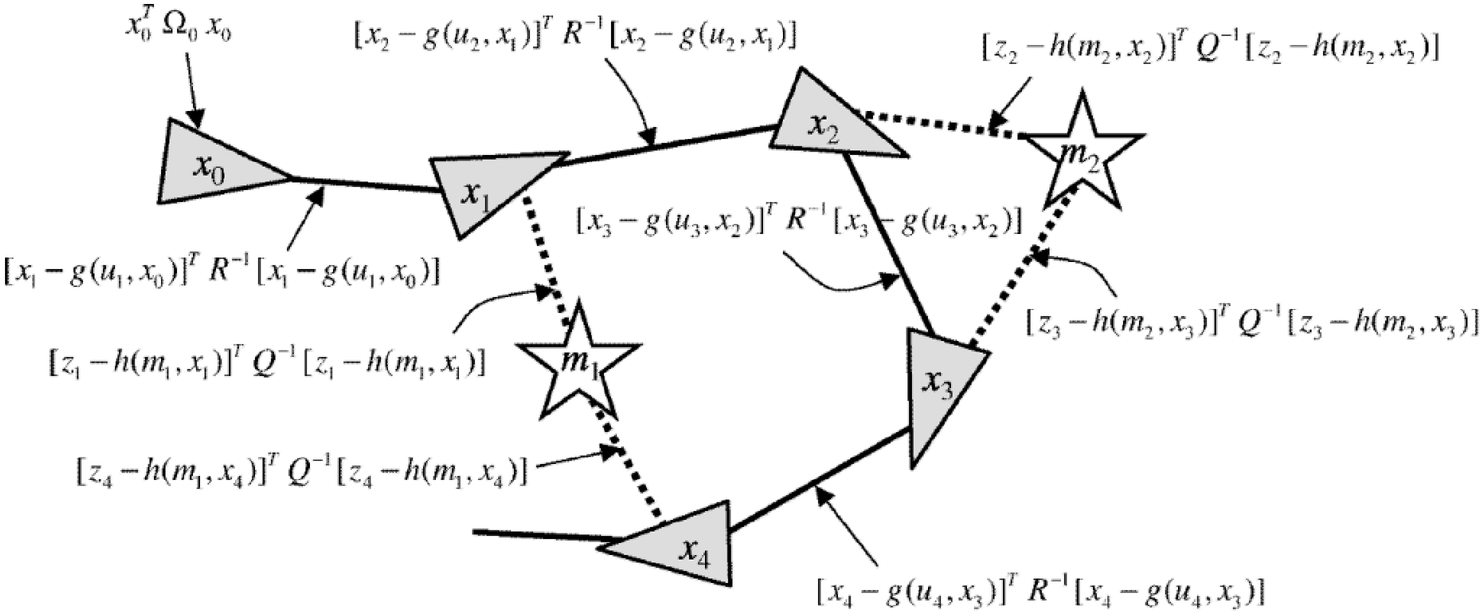
False

EKF SLAM applies the extended Kalman filter to the online SLAM problem.

GraphSLAM

GraphSLAM

A network of soft constraints:



<u>Sum of all constraints:</u>

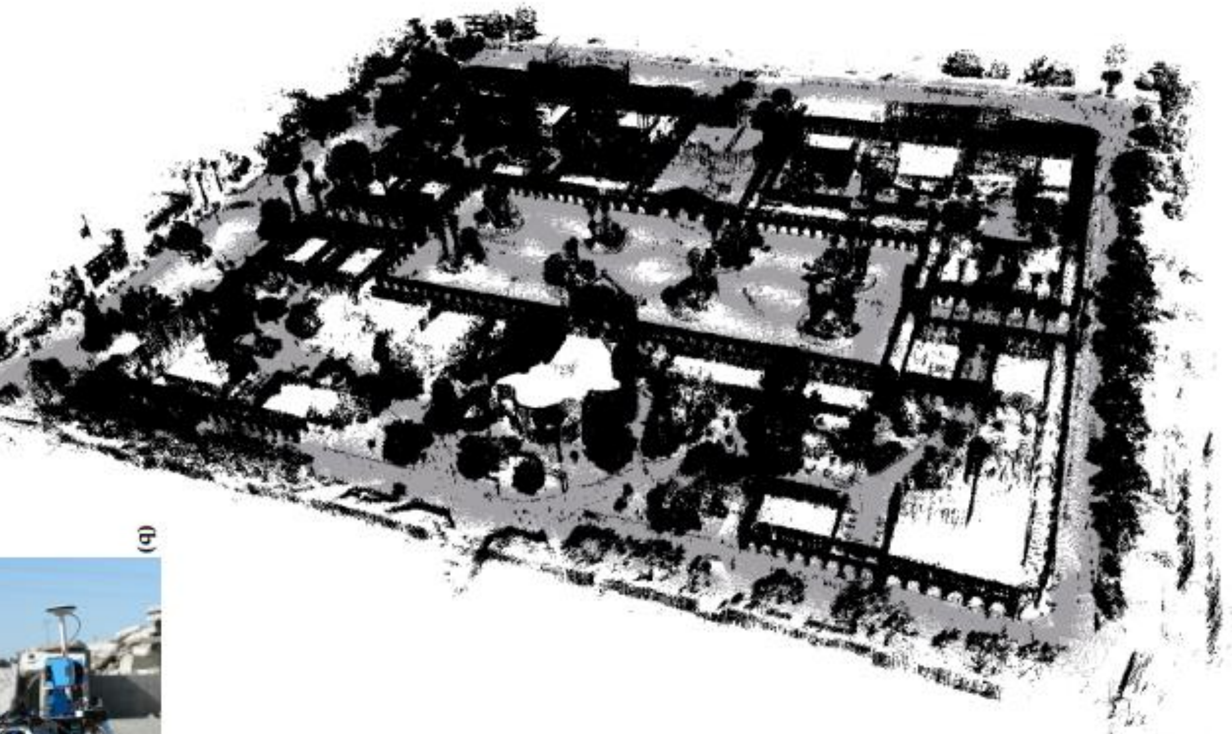
$$J_{\text{GraphSLAM}} = x_0^T \Omega_0 x_0 + \sum_t [x_t - g(u_t, x_{t-1})]^T R^{-1} [x_t - g(u_t, x_{t-1})] + \sum_t [z_t - h(m_{c_t}, x_t)]^T Q^{-1} [z_t - h(m_{c_t}, x_t)]$$

GraphSLAM

Sum constrains and minimize to get both the map and the full trajectory.

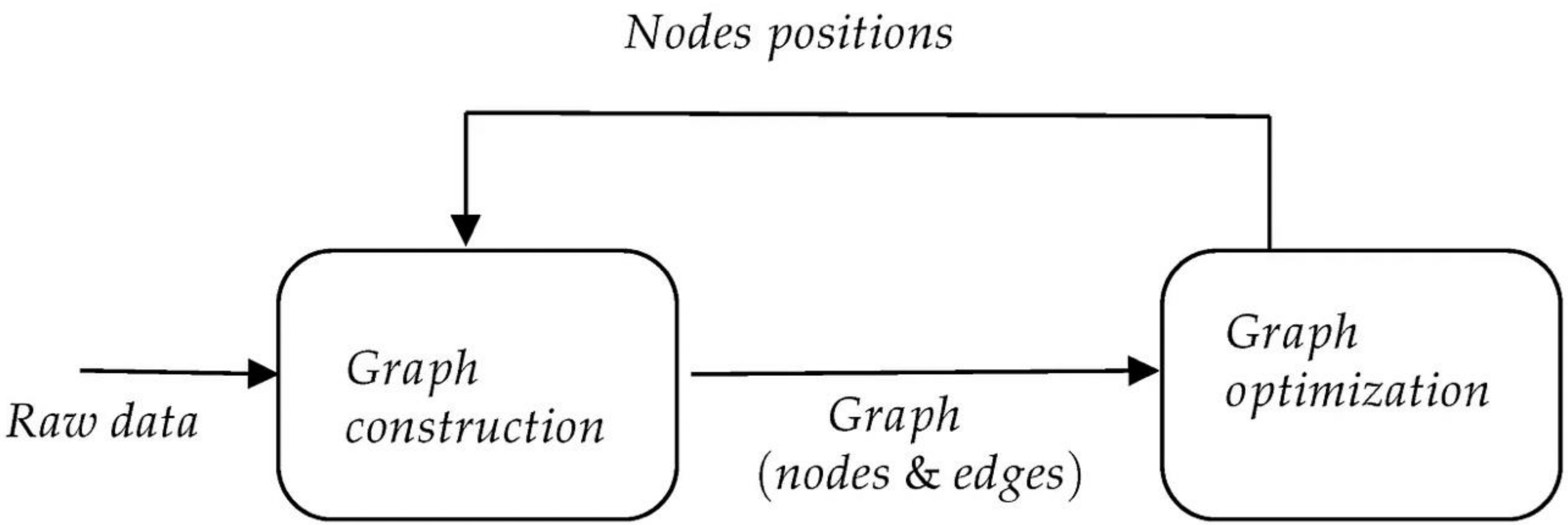
Solves the *full SLAM* problem.

(a)



(b)

GraphSLAM – Flow of data

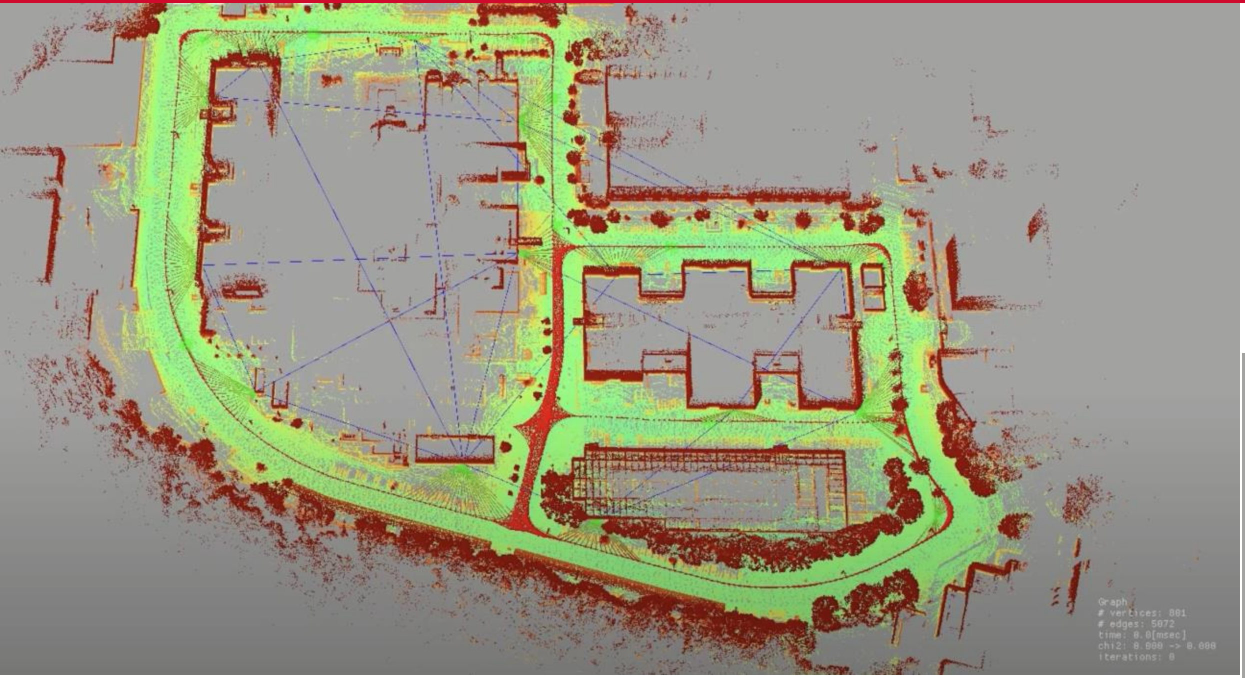


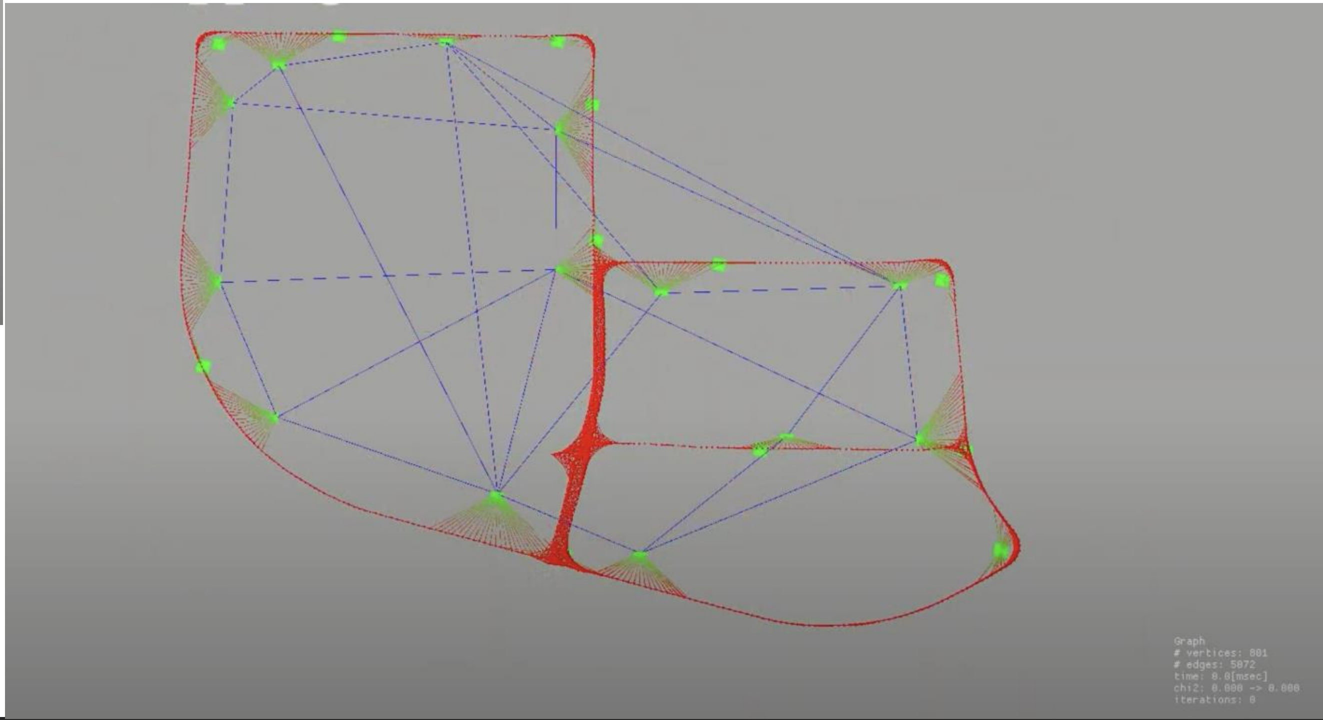
GraphSLAM - Data Association

There is no requirement to process the observed features sequentially.

Decisions can be undone.

GraphSLAM - Visualisations





GraphSLAM - Question

The distance between any two landmarks will always converge to the correct distance.

True or False?

GraphSLAM - Question

True

LiDAR-based SLAM

LiDAR-based SLAM - Examples

GMapping is a well-known implementation of the Rao-Blackwellized Particle Filter (RBPF).

slam_gmapping

slam_gmapping is a wrapper around the GMapping SLAM library. It reads laser scans and odometry and computes a map. This map can be written to a file using e.g.

"rosrun map_server map_saver static_map:=dynamic_map"

gmapping

This package contains a ROS wrapper for OpenSlam's Gmapping. The gmapping package provides laser-based SLAM (Simultaneous Localization and Mapping), as a ROS node called slam_gmapping. Using slam_gmapping, you can create a 2-D occupancy grid map (like a building floorplan) from laser and pose data collected by a mobile robot.

- Homepage: http://wiki.ros.org/gmappin

ROS topics

Subscribes to (name/type):

- **"scan"/** sensor_msgs/LaserScan : data from a laser range scanner
- **"/tf"**: odometry from the robot

Publishes to (name/type):

- **"/tf"/tf/tfMessage**: position relative to the map

services

- **"~dynamic_map"** : returns the map

ROS parameters

Reads the following parameters from the parameter server

Parameters used by our GMapping wrapper:

- **"~throttle_scans"**: [int] throw away every nth laser scan
- **"~base_frame"**: [string] the tf frame_id to use for the robot base pose
- **"~map_frame"**: [string] the tf frame_id where the robot pose on the map is published
- **"~odom_frame"**: [string] the tf frame_id from which odometry is read
- **"~map_update_interval"**: [double] time in seconds between two recalculations of the map

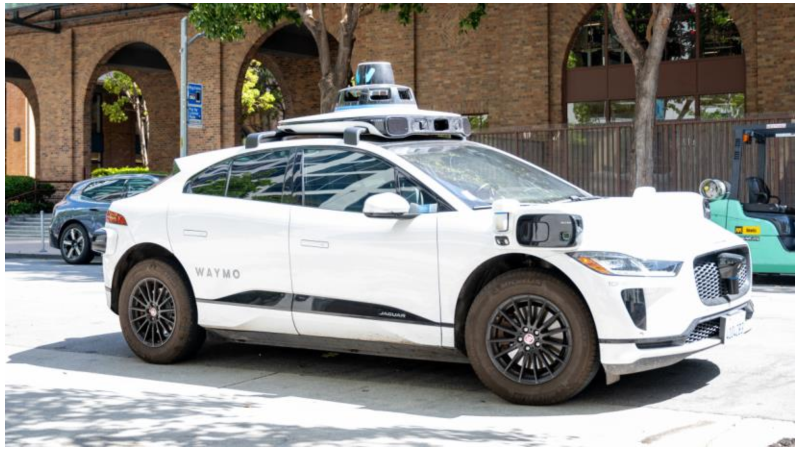
Parameters used by GMapping itself:

Laser Parameters:

- **"~/maxRange"** [double] maximum range of the laser scans. Rays beyond this range get discarded completely. (default:

LiDAR-based SLAM - Uses

- Self-driving cars
- Cleaning robots
- Drones - various

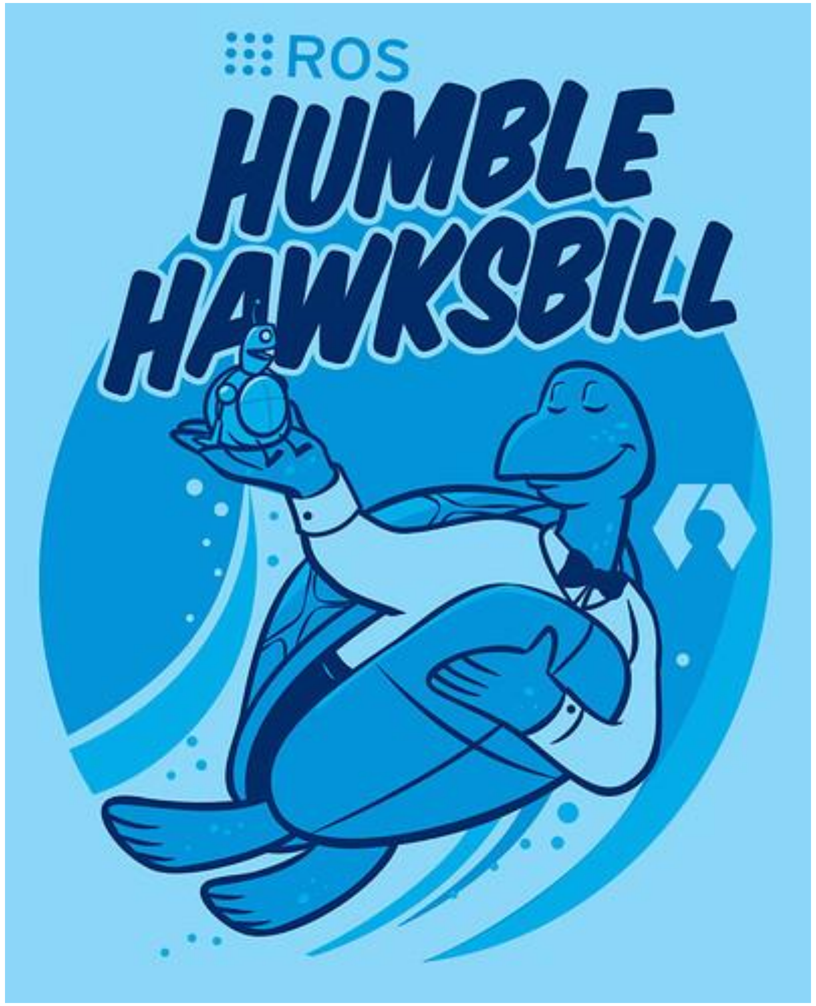


Link to ROS

Link to ROS

- Let's consider implementation using ROS

- Topics



Subscribers

- What topics would you need to subscribe to?

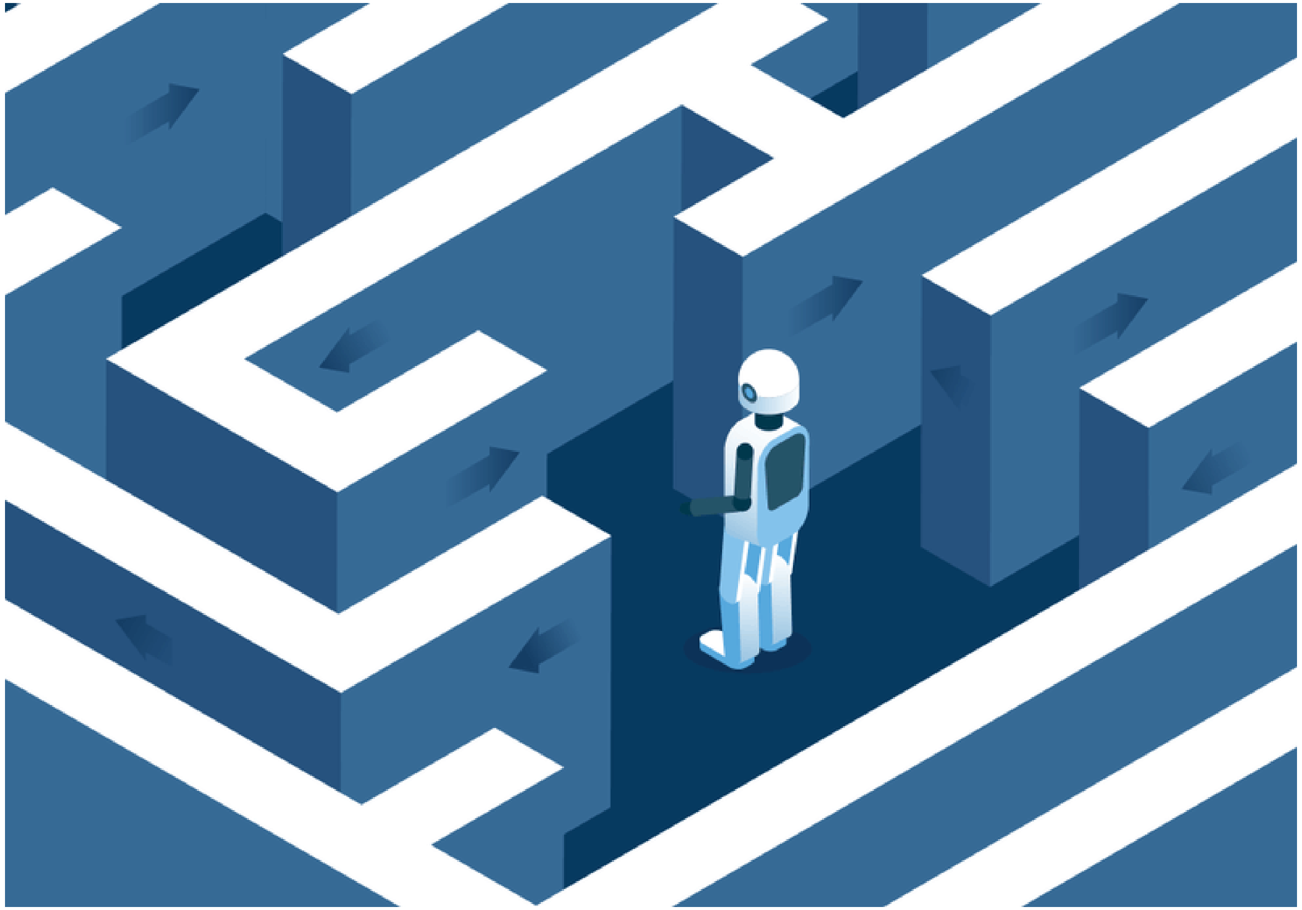
Publishers

- What topics would you need to publish to?

Extra Applications

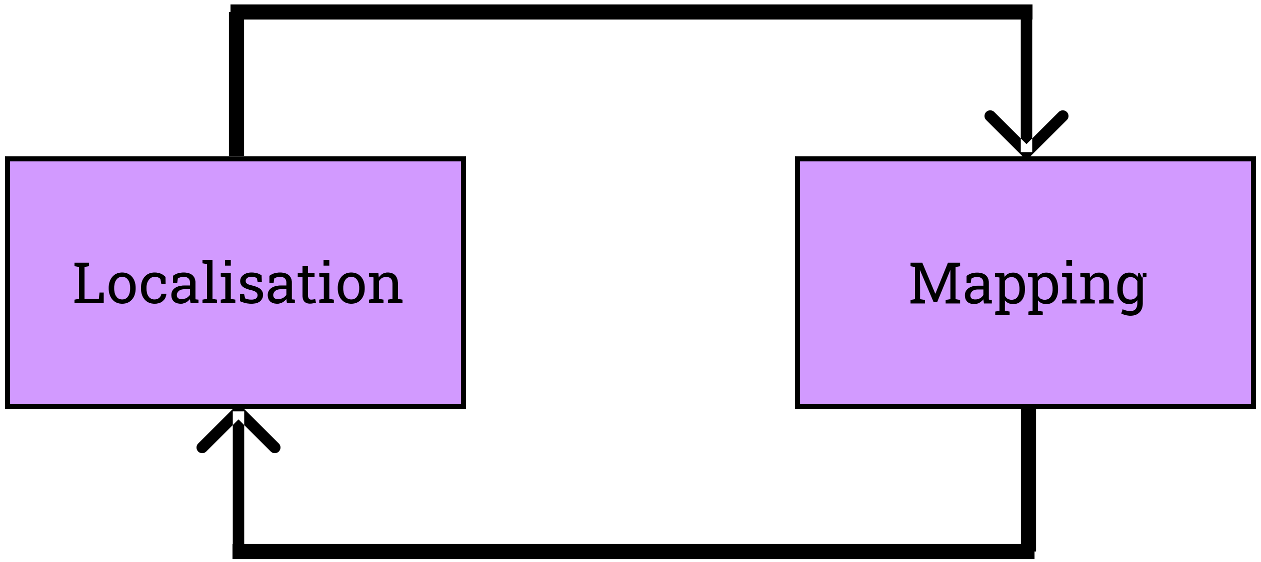
SLAM - Other Cool Applications

- Farming
- Store rooms – organize and collect stock
- Medicine – for small surgeries
- Augmented Reality
- Construction
- On the moon
- Exploring the oceans



Summary

Summary - SLAM



Interested in finding out more?