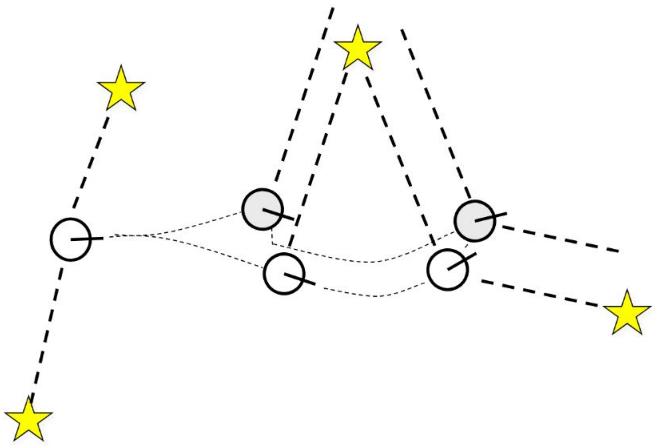
RRC Summer school

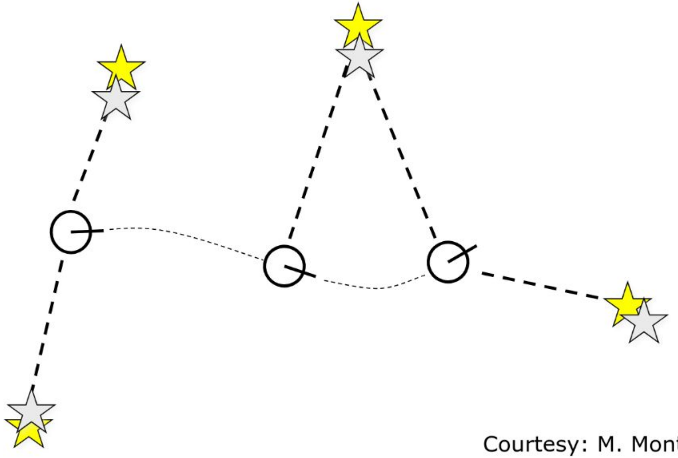
SLAM systems - 1

SLAM overview

For a robot to operate **autonomously** in an **unknown environment**, it must:

- Understand *where it is* (localization).
- Understand *it's environment* (mapping).





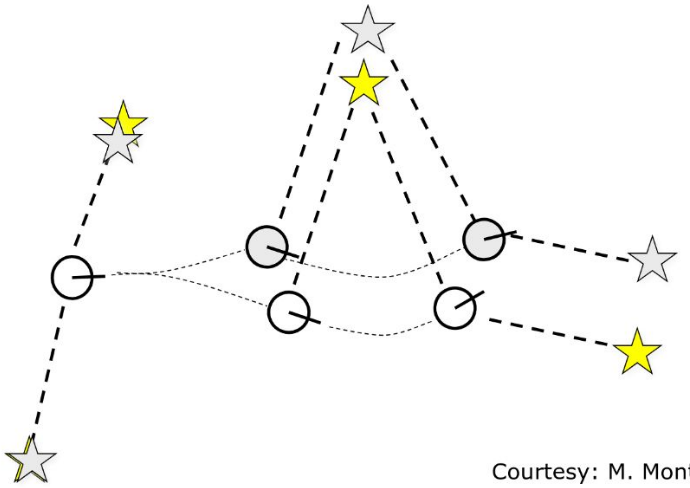
Courtesy: M. Montemerlo

SLAM overview

Without any prior information, how do you estimate any of these?

- Localization requires a map
  - How do you use your sensor inputs if you don't know where to look?
- Mapping requires localization
  - Without a precise location, sensor locations can come from anywhere

We can solve them **together**



Courtesy: M. Montemerlo

SLAM overview

Enter **Simultaneous Localization and Mapping** (SLAM).

Formally, SLAM systems estimate the robot's pose/trajectory and a map of the environment **simultaneously** from an **internal map/noisy control inputs** and **sensor measurements**.

Continuously use incoming sensor information to:

- Improve our localization estimate
- Generate a more accurate map

Incoming sensor information (from a variety of sensors) is treated as a set of constraints. More constraints allow for better estimation of robot pose and environment information

SLAM use cases

SLAM is considered a fundamental problem for truly autonomous robots, and is the basis of most navigation systems

- Search and Rescue: Robots explore collapsed buildings, mines – no GPS, no prior maps.
- Warehouse automation: Layouts may change, items are moved, and robots need to re-map areas dynamically.
- Planetary exploration: No prior environmental knowledge is available.
- Consumer robots (vacuums, drones): Cost-effective autonomy in unfamiliar home or office layouts.
- AR/VR and wearable computing: Headsets must localize in real time relative to their environment.

Commonly used sensors and sensor setups

Commonly used sensors:

Sensor Type	Data Provided	Strengths	Limitations
Monocular Camera	2D images	Low cost, high density	No scale, light-sensitive
Stereo depth	Depth information	Metric depth	Range-limited
Wheel/rotor encoders	Odometry	Simple, cheap	Slippage, accumulates error
IMU	Linear acc., angular vel.	High rate, compact	Drift over time
LiDAR	2D/3D point clouds	Accurate range	Expensive

Commonly used sensors and sensor setups

Commonly used setups:

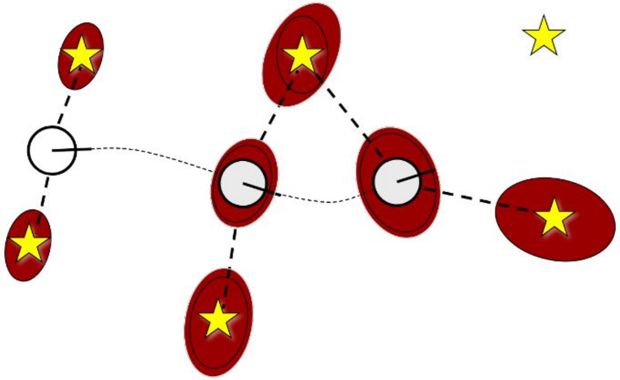
Setup	Components	Examples
Monocular	Single RGB camera	ORB-SLAM2 (no depth)
Stereo/RGB-D	Dual RGB / RGB-D	Stereo ORB-SLAM2/ElasticFusion, RTAB-Map
LiDAR-only	LiDAR only	Cartographer, LOAM
VIO	IMU + RGB	VINS-Fusion, ROVIO
LIO	IMU + LiDAR	LIO-SAM

Challenges faced by SLAM systems

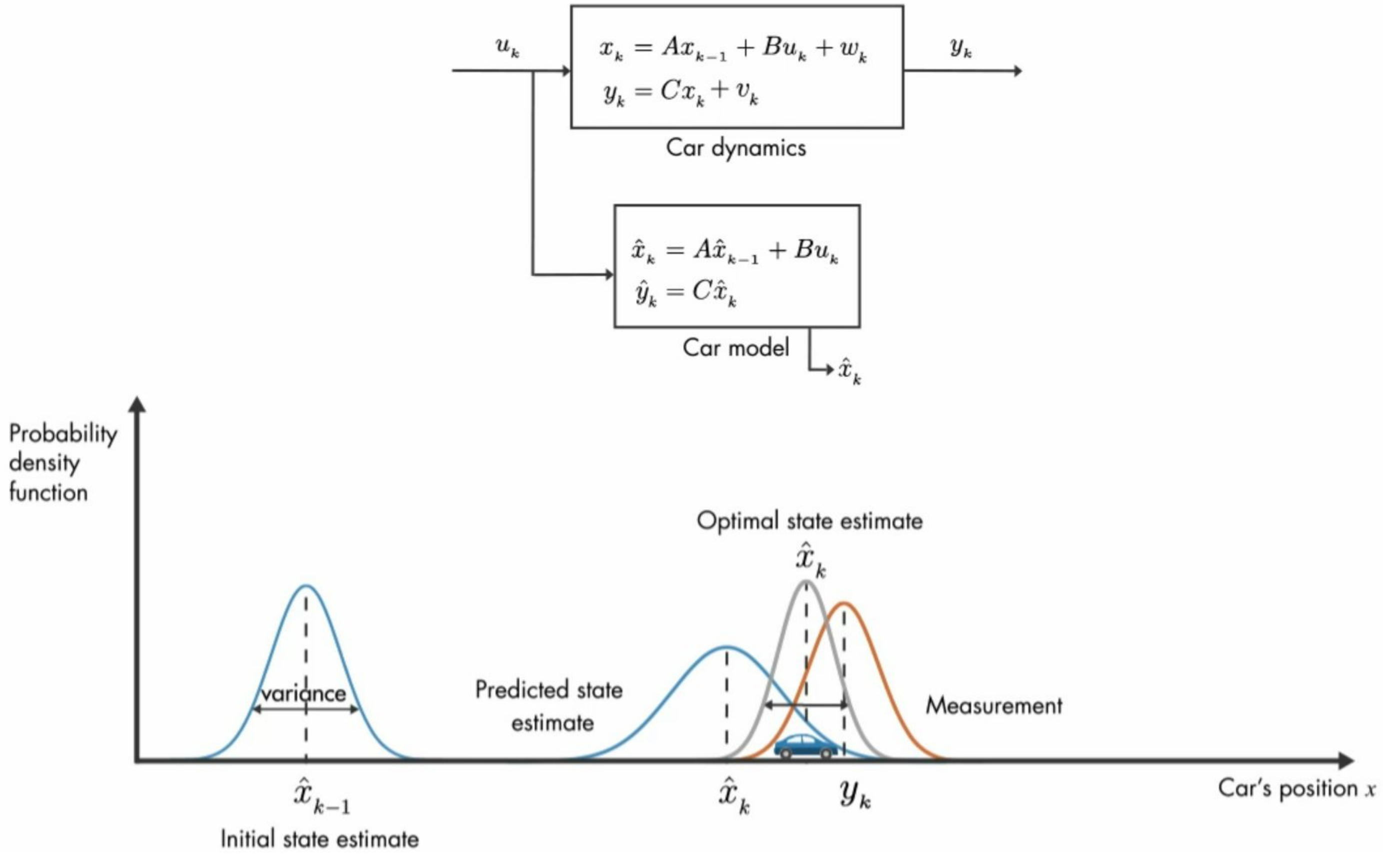
- Sensor noise
- Odometry drift
- High computational needs (real time processing)
- Incorrect edges and loop closure errors
- Long term mapping

Probabilistic methods

- Uncertainty and noise in the robots odometry and sensor readings is modelled mathematically as Gaussians
- Represent the robots pose as a probability distribution
- Keep updating this distribution as the robot moves
  - The robot moving will “widen” this distribution as noise accumulates
  - Loop closures or repeats “narrow” it
- Loop of predicting and correcting



Probabilistic methods



Pose graph methods

- Instead of a probabilistic approach, model the sum of knowledge so far as a graph
- Nodes are beliefs about the robots pose and the map
- Edges are pieces of information that link nodes, aka. “constraints”
- We can encode all constraints into a matrix, minimize its Hessian
- Loop closures serve as vital sources of information

- The error in the pose graph is to be minimized
- Minimization typically done by efficient solvers like G2O and Ceres
- Non linear least squares

Pose graph optimisation

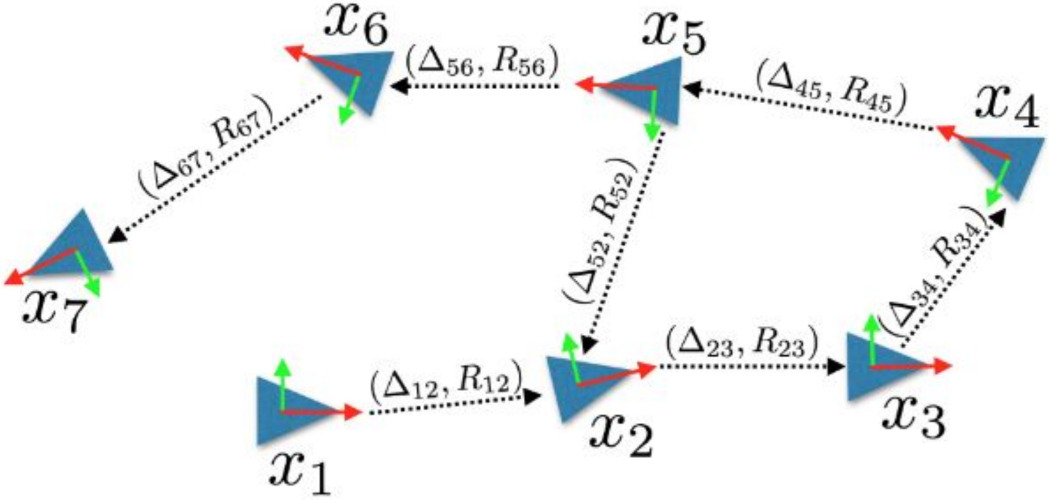
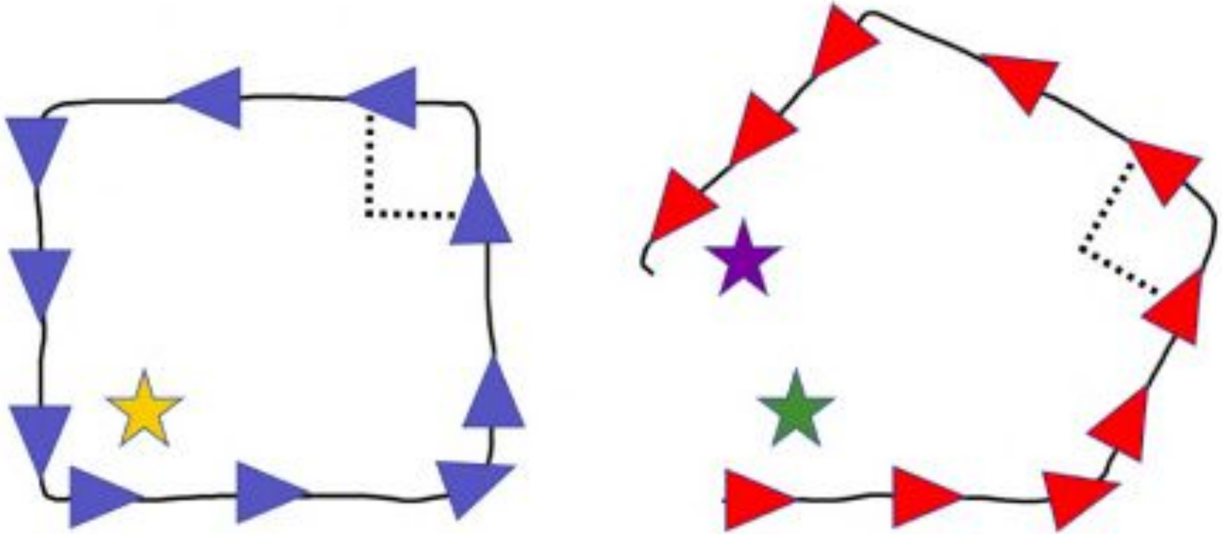


Figure 2. Schematic representation of Pose Graph Optimization: the objective is to associate a pose $\boldsymbol{x}_i$ to each node of a directed graph, given relative pose measurements $(\boldsymbol{\Delta}_{ij}, \boldsymbol{R}_{ij})$ for each edge (i, j) in the graph.

Loop closures

- Repetitions in the environment can clarify noisy information and form accurate maps



SLAM pipeline