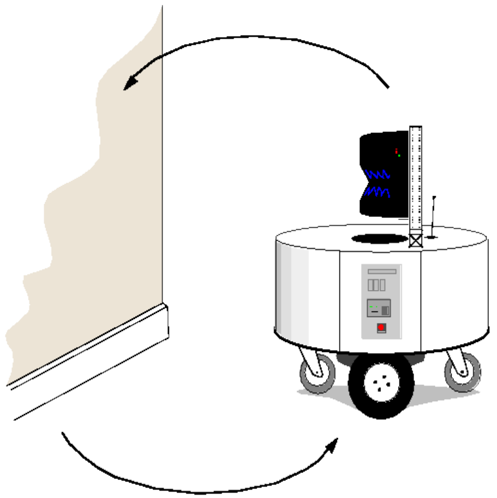
# What is SLAM?

- Estimate the pose of a robot and the map of the environment at the same time
- SLAM is hard, because
  - a map is needed for localization and
  - a good pose estimate is needed for mapping

- **Localization:** inferring location given a map
- **Mapping:** inferring a map given locations
- **SLAM:** learning a map and localizing the robot simultaneously

# The SLAM Problem

- SLAM has long been regarded as a **chicken-or-egg** problem:
  → a map is needed for localization and
  → a pose estimate is needed for mapping



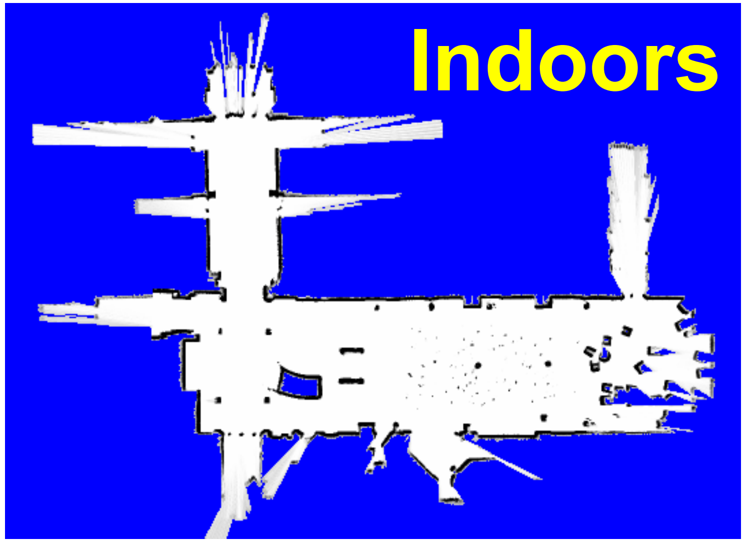
# SLAM Applications

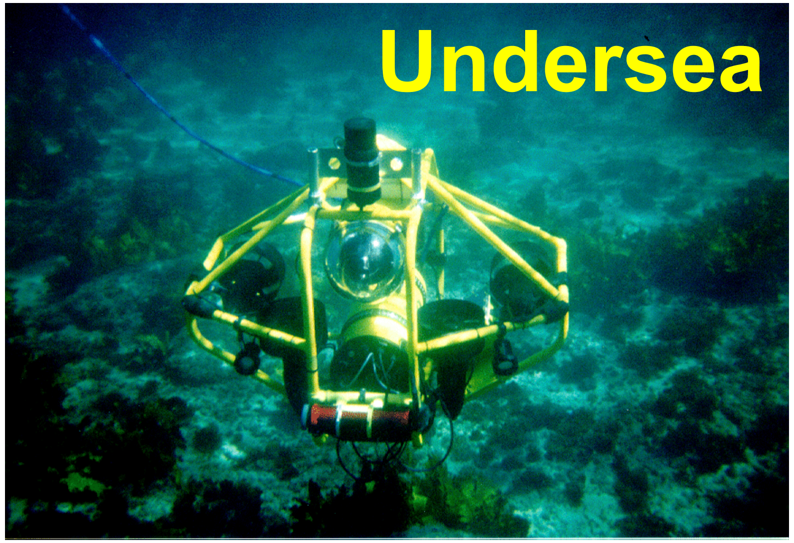
- SLAM is central to a range of indoor, outdoor, in-air and underwater applications for both manned and autonomous vehicles.

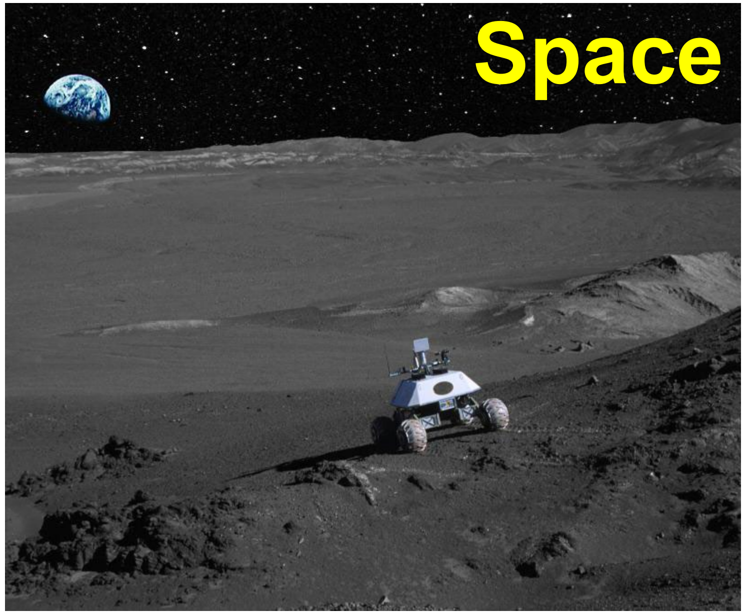
**Examples:**

- At home: vacuum cleaning, lawn mowing
- Air: surveillance with unmanned air vehicles
- Underwater: reef monitoring
- Underground: exploration of mines
- Space: terrain mapping for localization
- Cars: staying on the road

# SLAM Applications

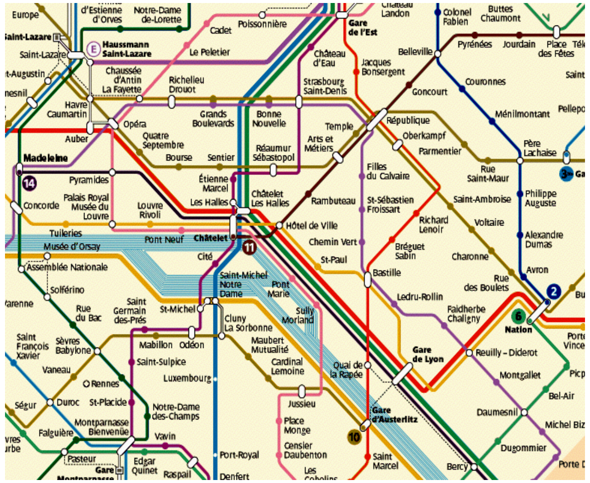


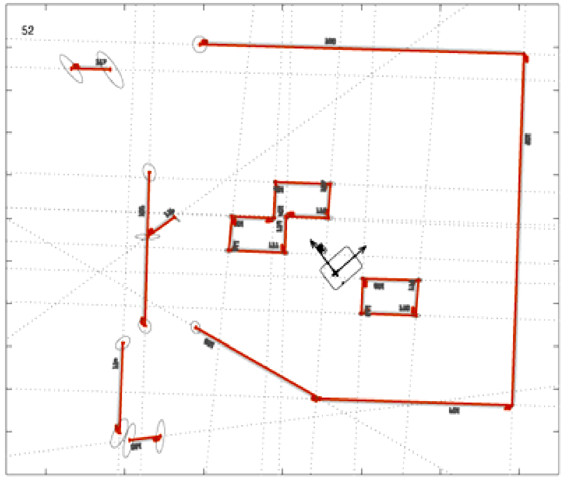




# Map Representations

**Examples:** Subway map, city map, landmark-based map

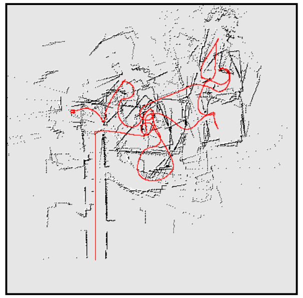


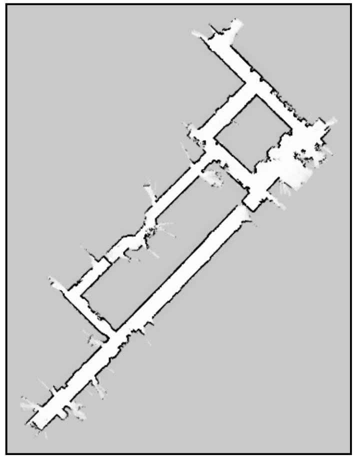


Here: Maps are **topological** and/or **metric models** of the environment

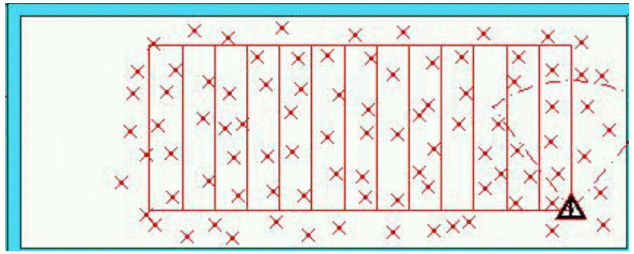
# Map Representations in Robotics

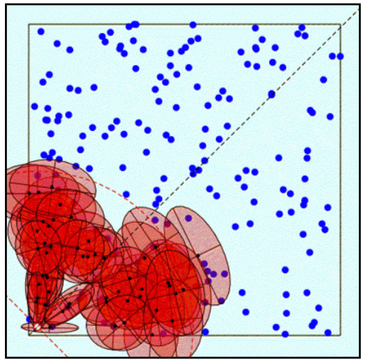
- Grid maps or scans, 2d, 3d





- Landmark-based





# The SLAM Problem

- SLAM is considered a fundamental problem for robots to become truly autonomous
- Large variety of different SLAM approaches have been developed
- The majority uses probabilistic concepts
- History of SLAM dates back to the mid-eighties

# Different Variants of the SLAM Problem

- Depending on the representation
  - Feature-based (landmarks or features)
  - Dense (2d/3d grids, signed distance functions, …)
- Depending on the sensor
  - Vision
  - Proximity sensors
- Depending on the sensor information
  - Range only
  - bearing only
  - Range and bearing
- Depending on the algorithm
  - Filtering-based
  - Optimization-based

# Feature-Based SLAM

## Given:

- The robot's controls

$$\boldsymbol{U}_{1:k} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k\}$$

- Relative observations

$$\boldsymbol{Z}_{1:k} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_k\}$$

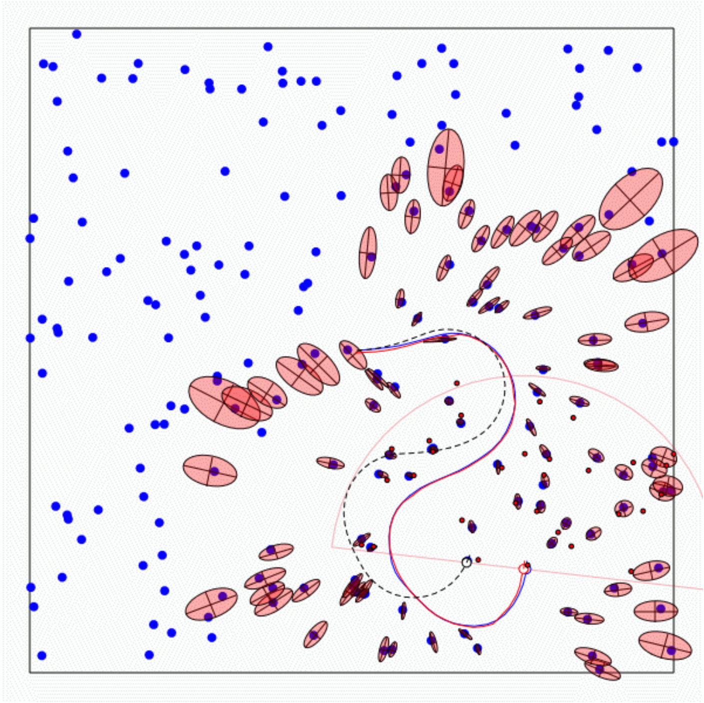
## Wanted:

- Map of features

$$\boldsymbol{m} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_n\}$$

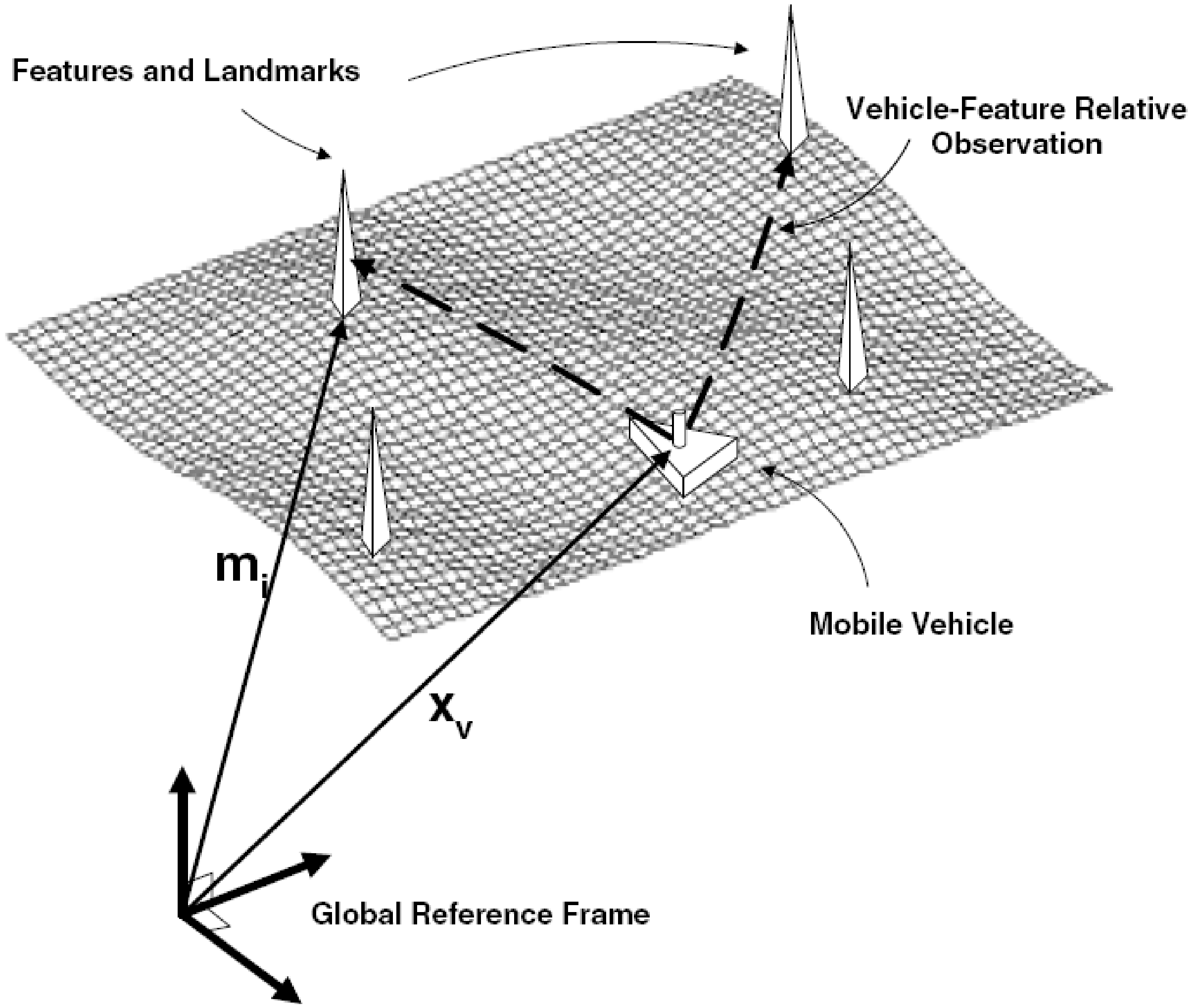
- Path of the robot

$$\boldsymbol{X}_{1:k} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$$



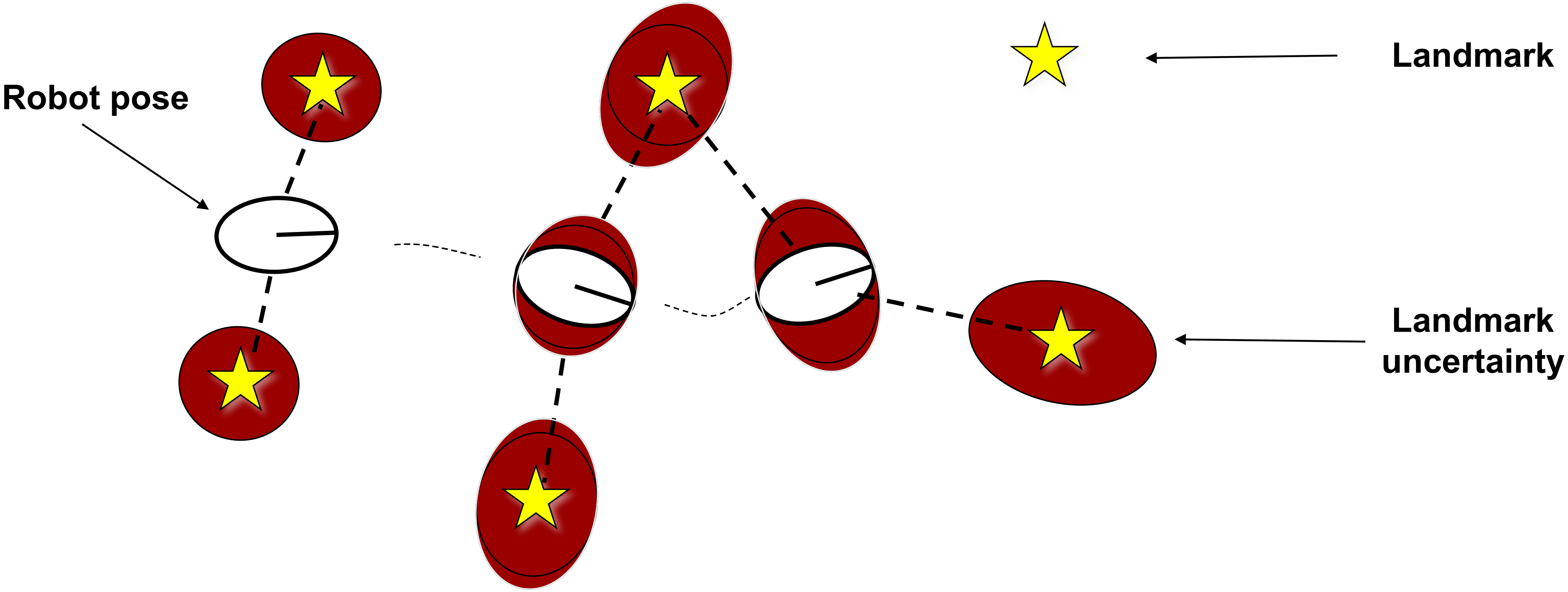
# Feature-Based SLAM

- **Absolute** robot poses
- **Absolute** landmark positions
- But only **relative** measurements of landmarks



# Why this SLAM Problem is Hard

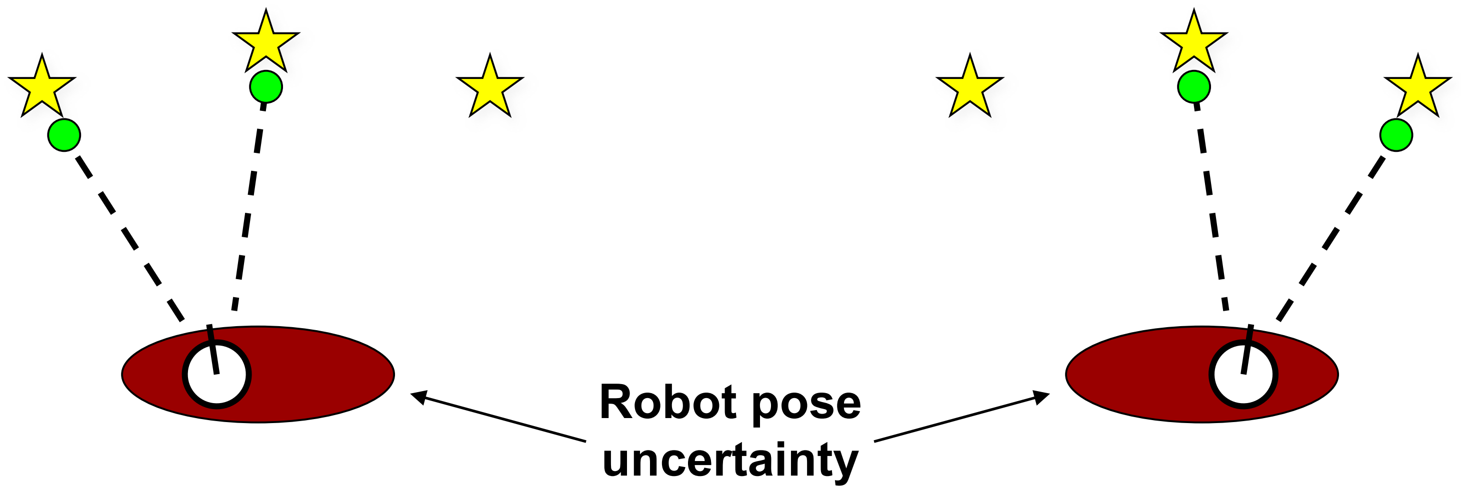
1. Robot path and map are both **unknown**




2. Errors in map and pose estimates correlated

# Why this SLAM Problem is Hard

- The **mapping between observations and landmarks is unknown**
- Picking **wrong** data associations can have **catastrophic** consequences (divergence)



# SLAM: Simultaneous Localization And Mapping

- Full SLAM:

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t})$$

Estimates entire path and map!

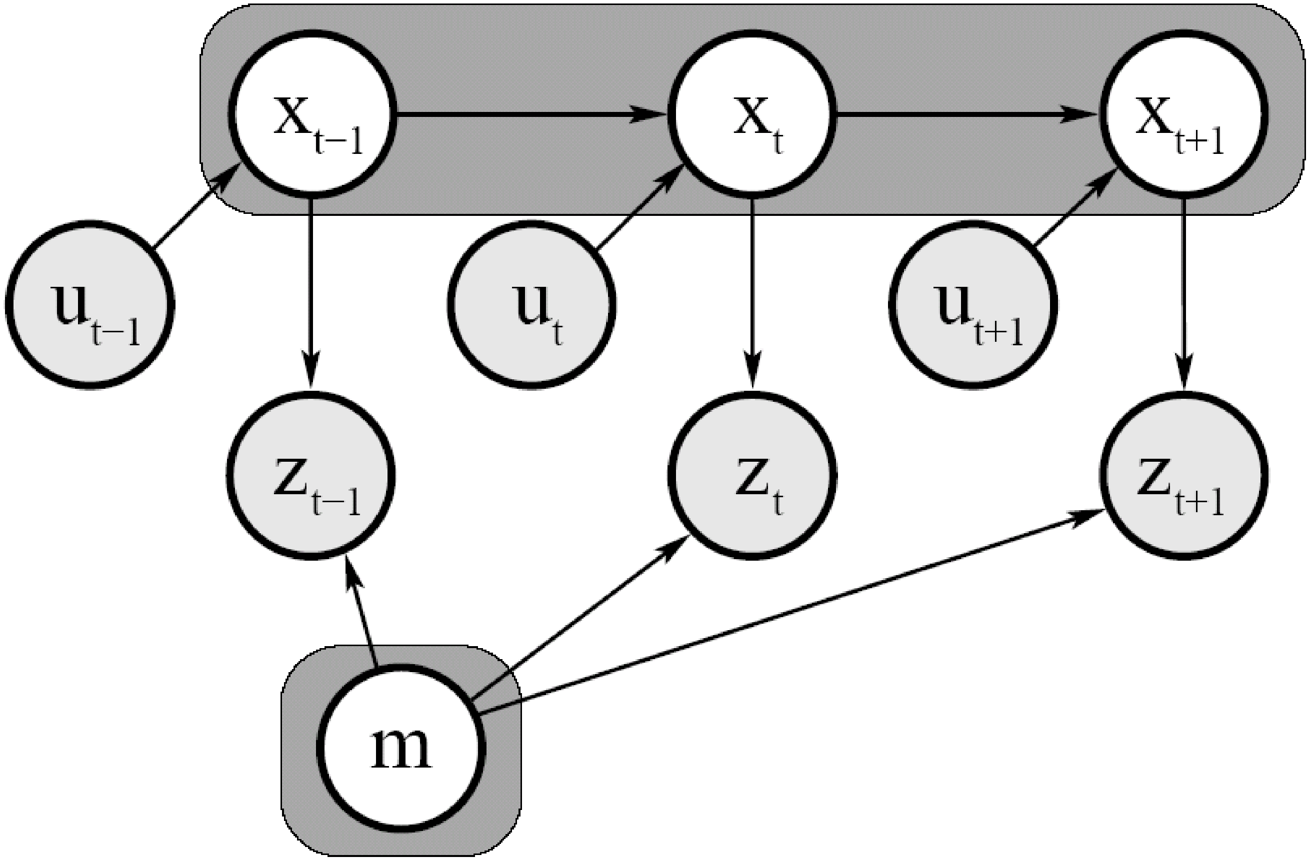
- Online SLAM:

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int\int \ldots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) dx_1 dx_2 \ldots dx_{t-1}$$

Estimates most recent pose and map!

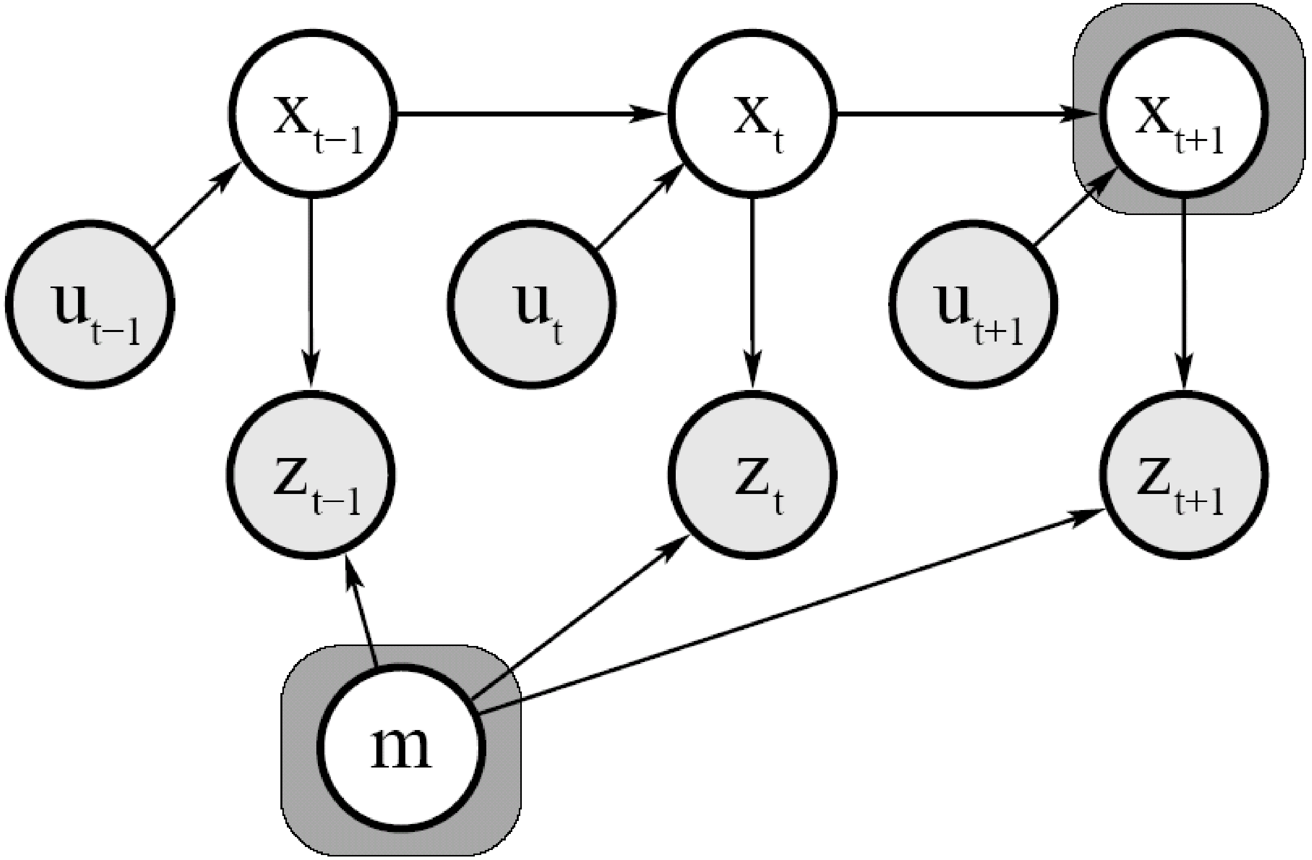
- Integrations (marginalization) typically done recursively, one at a time

# Graphical Model of Full SLAM




$$p(x_{1:t+1}, m \mid z_{1:t+1}, u_{1:t+1})$$

# Graphical Model of Online SLAM




$$p(x_{t+1}, m \mid z_{1:t+1}, u_{1:t+1}) = \int\int \ldots \int p(x_{1:t+1}, m \mid z_{1:t+1}, u_{1:t+1}) dx_1 dx_2 \ldots dx_t$$