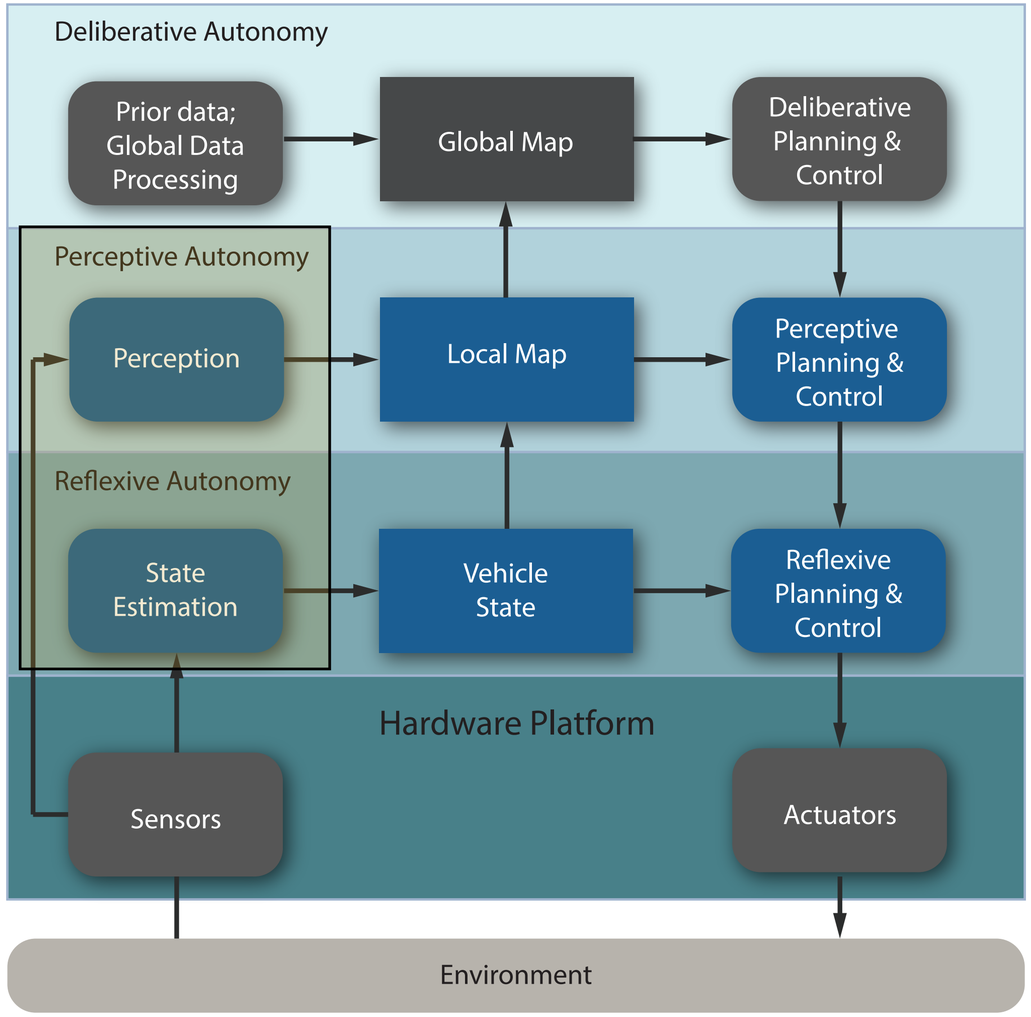
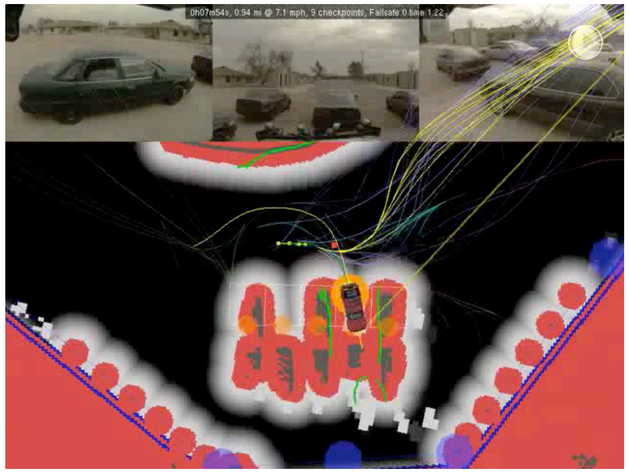
Hidden Markov Models

Planning on a Local Map

- We will now start working on the state estimation and perception tasks!
- Their goal is to create “state” or “map” that the planning systems can utilize.
- The world is uncertain, the sensors are noisy. So, we will aim to create a “probabilistic” understanding of the world!





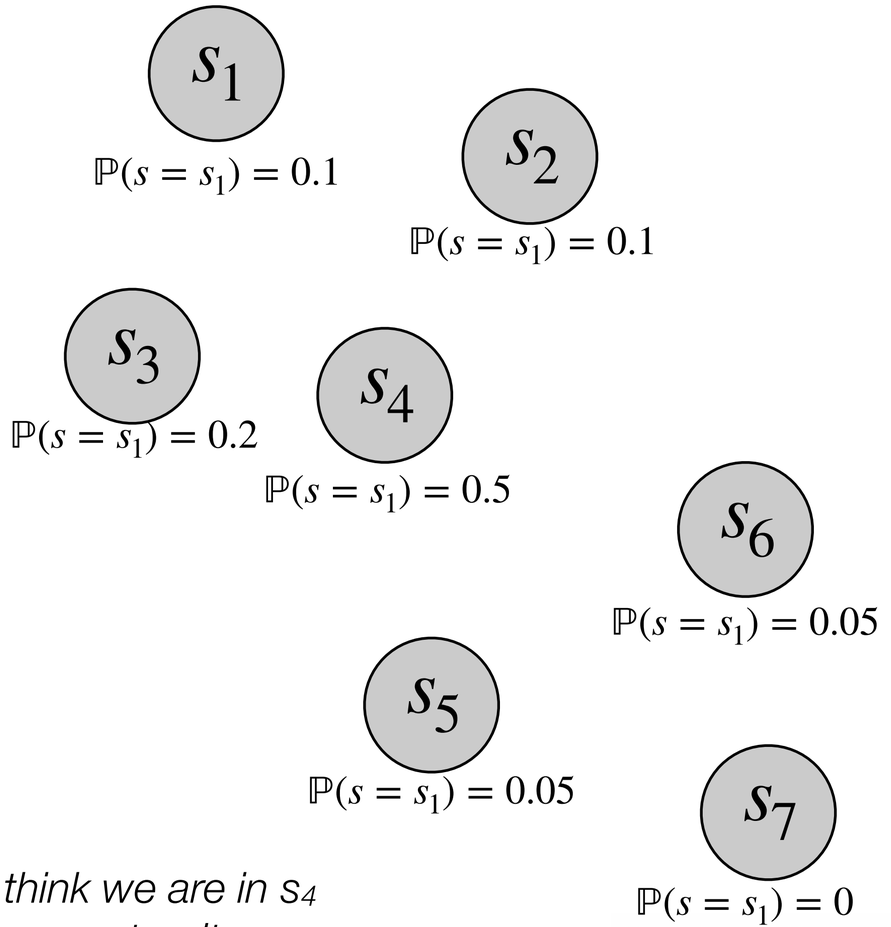
Modeling Uncertainty

- Consider a state space: *S*
  - Can be continuous or discrete

- We will not know exactly what state we are in. Hence, our goal is to assign a "probability distribution" over *S* to describe our understanding of the world right now!

- The goals of this lecture:
  - How do we describe this probability distribution?
  - How do we compute this probability distribution?

Discrete example



We think we are in s_4
but we are not quite sure…

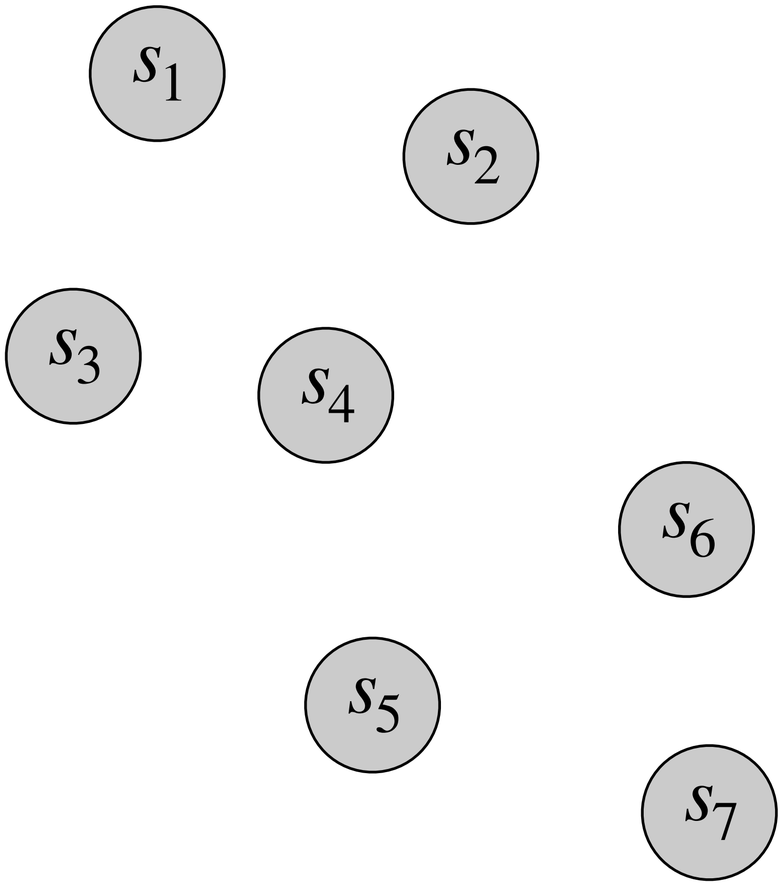
Conditional Probability and Modeling Observations

- The uncertainty stems from the fact that we can not observe the state directly.
- We will instead have a set of observations, O, where the observation will depend on the state we are in.
- Hence, we are concerned with the "probability of observing an observation o, given the state is s":

$$\mathbb{P}(O = o \,|\, S = s)$$

- In fact, we can use these probabilities to model our "sensor"

Discrete example

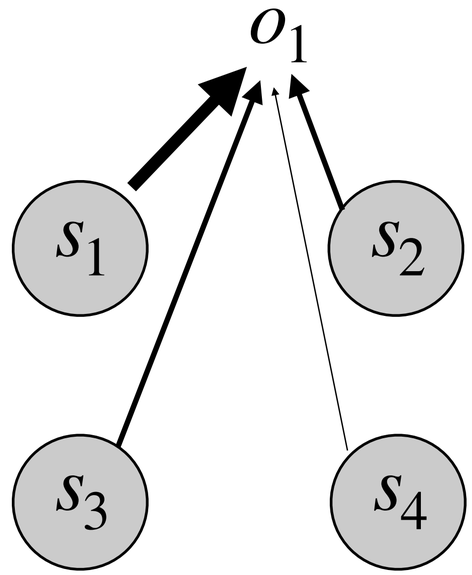


Simple example

- Consider a simple robot moving around a small grid, and let's assume the observations are "GPS" measurements that are noisy.

- Let's do it on the board together

$$S = \{s_1, s_2, s_3, s_4\}$$
$$O = \{o_1, o_2, o_3, o_4\}$$



More on Probability Foundations

- Joint distribution: the probability that you get two outcomes at the same time:

$$\mathbb{P}(O = o, S = s)$$

- Chain rule:

$$\mathbb{P}(O = o, S = s) = \mathbb{P}(O = o \,|\, S = s) \cdot \mathbb{P}(S = s)$$

- Why does it makes sense intuitively?
- What does it mean for our simple example?

More on Probability Foundations

- Joint distribution: the probability that you get two outcomes at the same time:

$$\mathbb{P}(O = o, S = s)$$

- Chain rule:

$$\mathbb{P}(O = o, S = s) = \mathbb{P}(O = o \,|\, S = s) \cdot \mathbb{P}(S = s)$$

- Why does it makes sense intuitively?
- What does it mean for our simple example?

More on Probability Foundations

- Marginalization: Means collapsing a dimension:

$$\mathbb{P}(O = o) = \sum_{S} \mathbb{P}(O = o, S = s)$$

- Why does it makes sense intuitively?
- What does it mean for our simple example?

More on Probability Foundations

- Bayes rule:

$$\mathbb{P}(S = s \,|\, O = o) = \frac{\mathbb{P}(O = o \,|\, S = s) \cdot \mathbb{P}(S = s)}{\mathbb{P}(o = o)}$$

- Why?

$$\mathbb{P}(O = o, S = s) = \mathbb{P}(O = o \,|\, S = s) \cdot \mathbb{P}(S = s)$$

$$\mathbb{P}(O = o, S = s) = \mathbb{P}(S = s \,|\, O = o) \cdot \mathbb{P}(O = o)$$

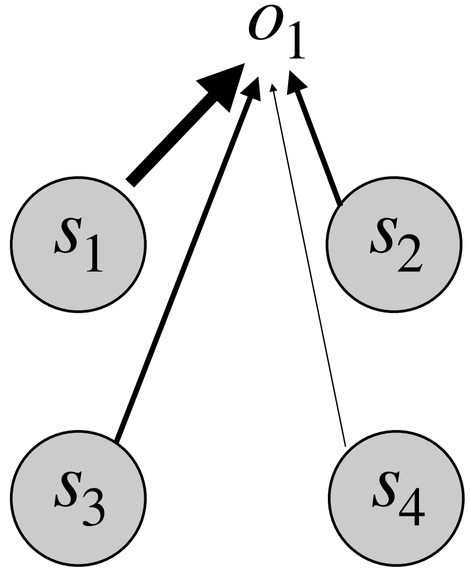
Robot state estimation and robot perception with probability concepts

- We will use probability to describe the uncertain world
  - We don't get to observe the state of the robot and the state of the world directly
  - We have noisy measurements of the world

$$S = \{s_1, s_2, s_3, s_4\}$$
$$O = \{o_1, o_2, o_3, o_4\}$$

- We will consider a probability distribution over the "state" of the robot
- We will model our measurements as a probability distribution over a set of observations given the state

- The Bayes rule will be our good friend: $\mathbb{P}(S = s \,|\, O = o) = \dfrac{\mathbb{P}(O = o \,|\, S = s)}{\mathbb{P}(S = s)}$



Hidden Markov Models

- We model the uncertain world with its states, its dynamics, and our observations of this world.

- States are set S
- Transitions are defined by conditional probabilities
- Observations are a set O and defined by conditional probabilities

Hidden Markov Model

States:

$$S = \{s_1, s_2, \ldots, s_n\}$$

State transition probabilities:

$$\mathbb{P}(S_{t+1} = s' \,|\, S_t = s) \text{ for all } s, s' \in S$$

Observations:

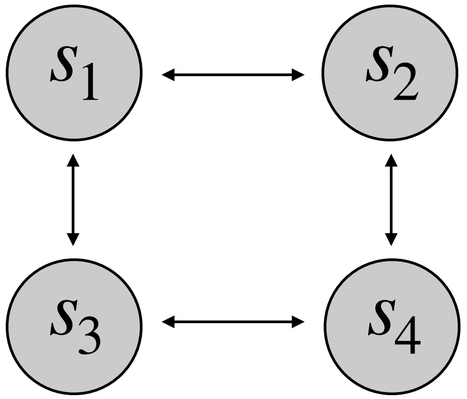
$$O = \{o_1, o_2, \ldots, o_m\}$$

$$\mathbb{P}(O = o \,|\, S = s) \text{ for all } o \in O \text{ and all } s \in S$$

Recall our simple example

- Let's do it together!

- We need to define:
  - States
  - State transition function
  - Observations
  - Observation function



Key Problems for Hidden Markov Models

- We are interested in the following problems:
  - **Filtering/Inference**:
    - Distribution over current state given all measurements
  - **Smoothing**:
    - Distribution over all states given all measurements
  - **Decoding:**
    - Distribution over sequence of states given all measurements

Hidden Markov Model

States:

$$S = \{s_1, s_2, \ldots, s_n\}$$

State transition probabilities:

$$\mathbb{P}(S_{t+1} = s' \,|\, S_t = s) \text{ for all } s, s' \in S$$

Observations:

$$O = \{o_1, o_2, \ldots, o_m\}$$

$$\mathbb{P}(O = o \,|\, S = s) \text{ for all } o \in O \text{ and all } s \in S$$

Filtering Problem for Hidden Markov Models

Hidden Markov Model

States:

$$S = \{s_1, s_2, \ldots, s_n\}$$

State transition probabilities:

$$\mathbb{P}(S_{t+1} = s' \,|\, S_t = s) \text{ for all } s, s' \in S$$

Observations:

$$O = \{o_1, o_2, \ldots, o_m\}$$

$$\mathbb{P}(O = o \,|\, S = s) \text{ for all } o \in O \text{ and all } s \in S$$

- Given an HMM (on the left)
- Find:
  - $\mathbb{P}(S_t \,|\, o_t, o_{t-1}, \ldots, o_1)$

Filtering Problem for Hidden Markov Models

- Proceeds iteratively. Assumes we have an initial "prior" $\mathbb{P}(S_0)$
- At each iteration:
  - Given belief at time t: $\mathbb{P}(S_t \mid o_t, \ldots, o_1)$
  - Step 1: Prediction using known dynamics:
    - Apply the transition function to find the joint distribution:
      $\mathbb{P}(S_{t+1}, S_t \mid o_t, \ldots, o_1) = \mathbb{P}(S_{t+1} \mid S_t) \cdot \mathbb{P}(S_t \mid o_t, \ldots, o_1)$
    - Marginalize to compute the belief for next state before the measurement: $\mathbb{P}(S_{t+1} \mid o_t, \ldots, o_1) = \sum_{s \in S} \mathbb{P}(S_{t+1}, S_t = s \mid o_t, \ldots, o_1)$
  - Step 2: Integrate the measurement:
    - Apply Bayes rule to find the belief at time *t+1:*
      $\mathbb{P}(S_{t+1} \mid o_{t+1}, o_t, \ldots, o_1) = \dfrac{\mathbb{P}(O_{t+1} = o_{t+1} \mid S = s_{t+1}, o_t, \ldots, o_1) \cdot \mathbb{P}(S_{t+1} \mid o_t, \ldots, o_1)}{\mathbb{P}(O_{t+1} \mid o_t, \ldots, o_1)}$

Belief at time t

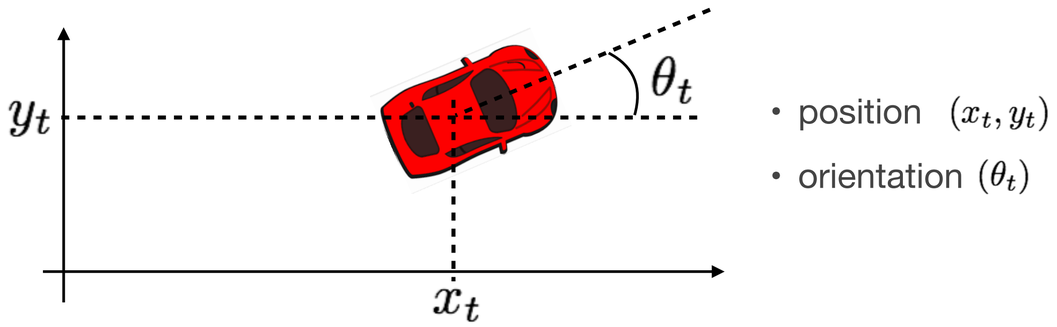
Transition function

Observation function

Normalization factor

Applications in Autonomy & Robotics:
1. Localization

- **Localization**: Given some representation of the environment, use sensing to determine the pose of the robot with respect to this environment.

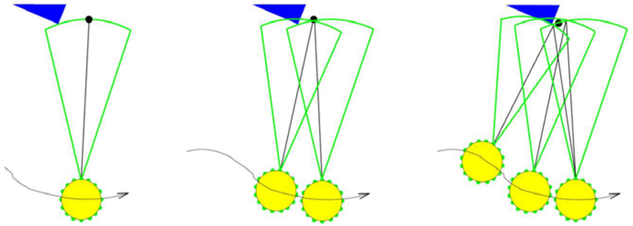


- Key aspects:
  - Sensing, actuation, and modeling uncertainty is embedded in any application.
  - We need exteroceptive sensors to localize.

Localization is an HMM where the states are not directly observed!

Applications in Autonomy & Robotics:
2. Mapping

- Mapping is possible assuming short-range dead reckoning is small.
- This technique has been used for centuries for mapping.



Pedro Reinel, 1504

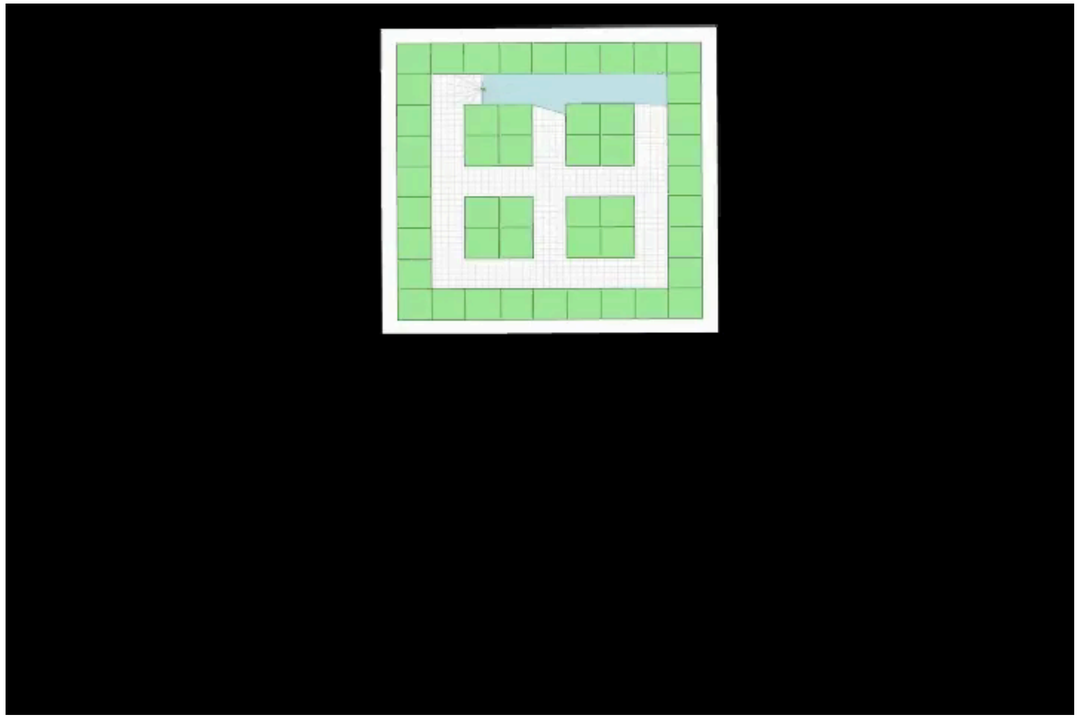
Dünya Haritası, Piri Reis
Topkapı Sarayı Müzesi Kütüphanesi. No. H. 1824
The World Map, Piri Reis
Library of Topkapı Palace Museum. No. H. 1824

Piri Reis, 1513

Applications in Autonomy & Robotics:
3. The Simultaneous Localization and Mapping (SLAM) Problem

- **Inputs**:
  - No external coordinate reference;
  - Instead, time series of proprioceptive and exteroceptive measurements acquired as robot moves through an initially unknown environment.

- **Outputs**:
  - A map of the environment;
  - A robot pose estimate associated with each measurement, in the coordinate system in which the map is defined.

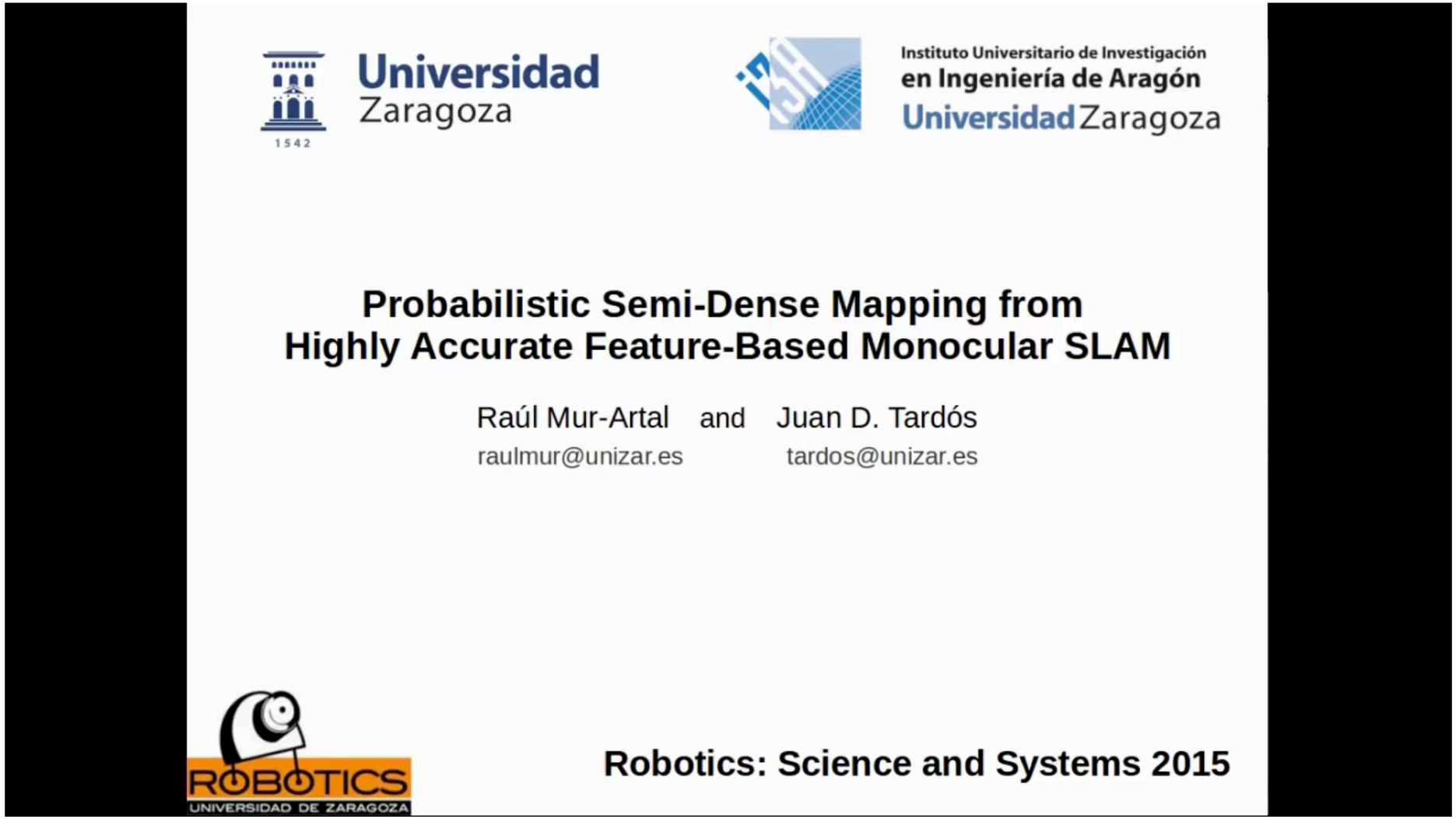


Top: ground truth

Left: PF-SLAM

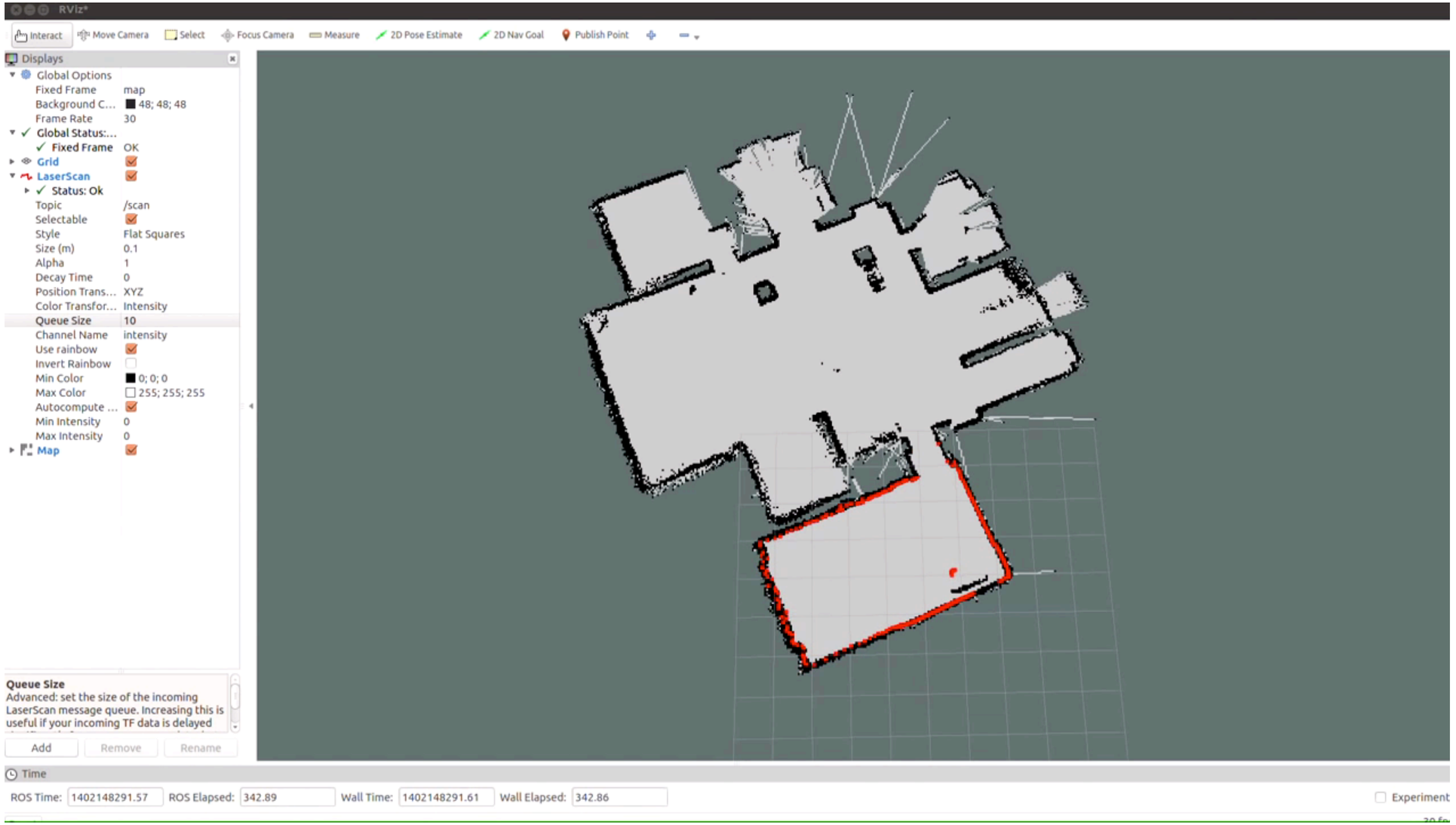
Right: mapping from odometry

SLAM: Example 1: ORB SLAM



SLAM: Example 2: Elastic Fusion

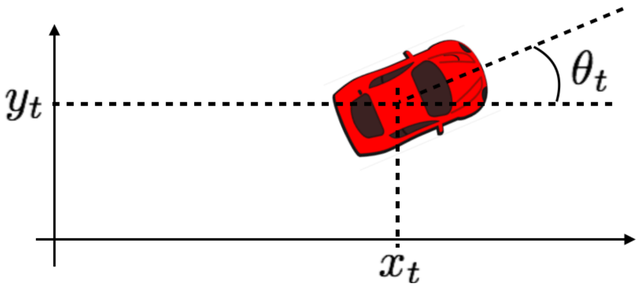
SLAM: Example 3: Using LiDAR data only



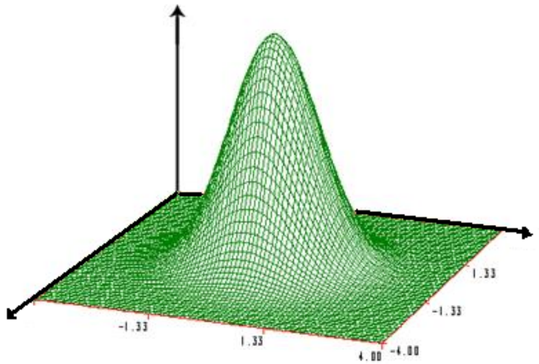
SLAM: Example 4: Using RGB camera only

* Additional material for those interested

Pose Belief



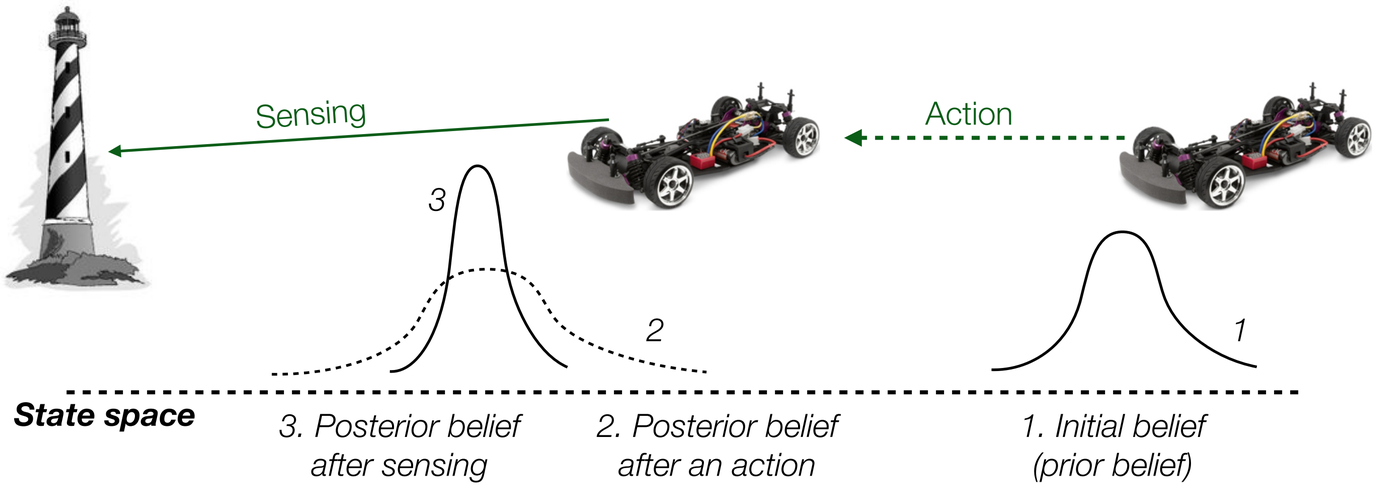
$$p_t = (x_t, y_t, \theta_t)$$



$$\mathbb{P}\left(p_t | u_{1:t}\right) = \mathbb{P}\left(x_t, y_t, \theta_t | u_{1:t}\right)$$

- **Definition (Pose):** Position and orientation of an object.
- **Definition (Pose Belief):** The set of all probability distributions over the set of all poses.

Localization: Evolution of the Pose Belief



- **Initial pose belief** models uncertainty in the robots knowledge of its pose.
- **Posterior pose belief after an action** is "wider" as forward integration accumulates the (now modeled) actuation errors
- **Posterior pose belief after sensing** is "narrower" as sensing helps the robot better understand its position.
- But, sensing does not allow the robot know its position exactly. Why?

Localization: The Bayes Filter

- Pose at time *t*: x_t
- Action at time t: a_t
- Measurement at time t: z_t

1. Pose belief at time *t*:

$$p(x_t|a_1, z_1, a_2, z_2, \ldots, a_t, z_t)$$

2. Separate “measurement” and “dynamics” models. According to the Bayes rule:

$$\alpha\, p(z_t|x_t, a_1, z_1, a_2, z_2, \ldots, a_t)\, p(x_t|a_1, z_1, a_2, z_2, \ldots, a_t)$$

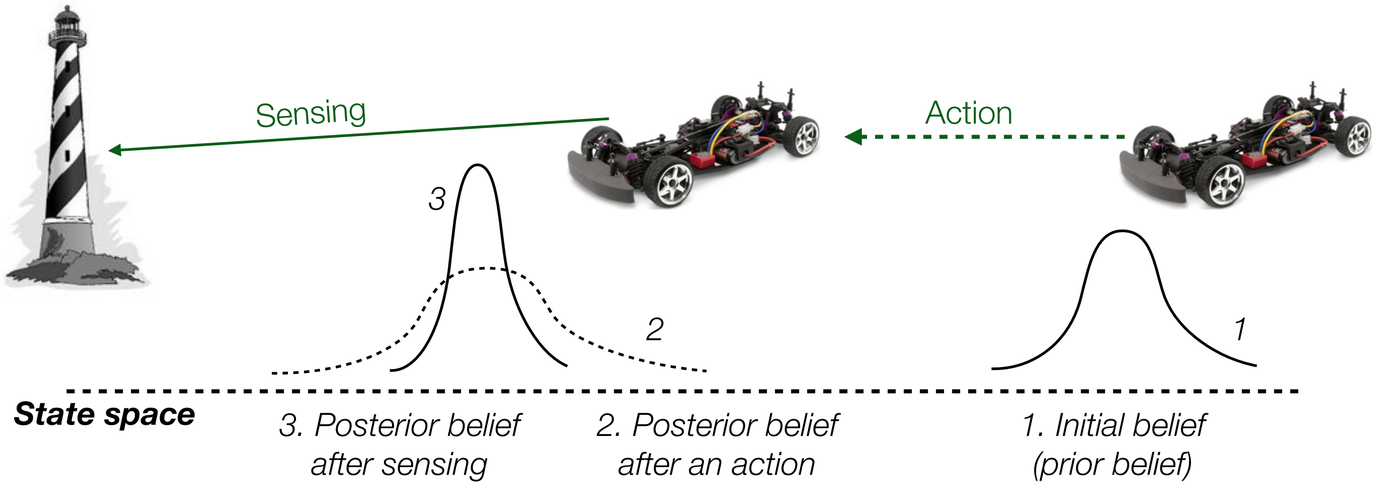
3. Assume that the measurement depends only on the current pose (Markov):

$$\alpha\, p(z_t|x_t)\, p(x_t|a_1, z_1, a_2, z_2, \ldots, a_t)$$

4. Introducing the pose at time t-1 as an auxiliary variable:

$$\alpha\, p(z_t|x_t) \int_X p(x_t|a_t, x_{t-1})\, p(x_{t-1}|a_1, z_1, a_2, z_2, \ldots, a_{t-1}, z_{t-1}) dx_{t-1}$$

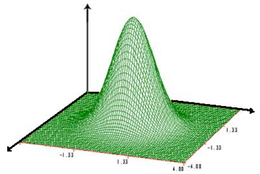
Localization: Evolution of the Pose Belief



Posterior pose belief

$$p(x_t|a_1, z_1, a_2, z_2, \ldots, a_t, z_t)$$

$$= \alpha\, p(z_t|x_t) \int_X p(x_t|a_t, x_{t-1})\, p(x_{t-1}|a_1, z_1, a_2, z_2, \ldots, a_{t-1}, z_{t-1}) dx_{t-1}$$



Measurement model

Dynamic model

Initial pose belief

x_t State

a_t Action

z_t Measurement

Bayes Filter for Localization and for SLAM

Bayes Filter for Localization

Posterior pose belief

$$p(x_t|a_1, z_1, a_2, z_2, \ldots, a_t, z_t)$$

$$= \alpha\, p(z_t|x_t) \int_X p(x_t|a_t, x_{t-1})\, p(x_{t-1}|a_1, z_1, a_2, z_2, \ldots, a_{t-1}, z_{t-1}) dx_{t-1}$$

Measurement model

Dynamic model

Initial pose belief

Bayes Filter for SLAM

$$p(x_t, m_t|a_1, z_1, \ldots, a_t, z_t)$$

$$= \alpha\, p(z_t|x_t, m_t) \int_X p(x_t, m_t|a_t, x_{t-1}, m_{t-1}) p(x_{t-1}, m_{t-1}|a_1, z_1, \ldots, a_{t-1}, z_{t-1}) d(x_{t-1}, m_{t-1})$$

The Bayes Filter (for continuous probability distributions)

Bayes Filter

Posterior pose belief

$$p(x_t|a_1, z_1, a_2, z_2, \ldots, a_t, z_t)$$

$$= \alpha\, p(z_t|x_t) \int_X p(x_t|a_t, x_{t-1})\, p(x_{t-1}|a_1, z_1, a_2, z_2, \ldots, a_{t-1}, z_{t-1}) dx_{t-1}$$

Measurement model

Dynamic model

Initial pose belief

- How can we discretize so that we can apply computing? A few routes:
  - Discretize the states, actions and observations
  - Represent continuous states, actions and observations with finitely-parametrized distributions
  - Discretize the distributions for states, actions and observations themselves

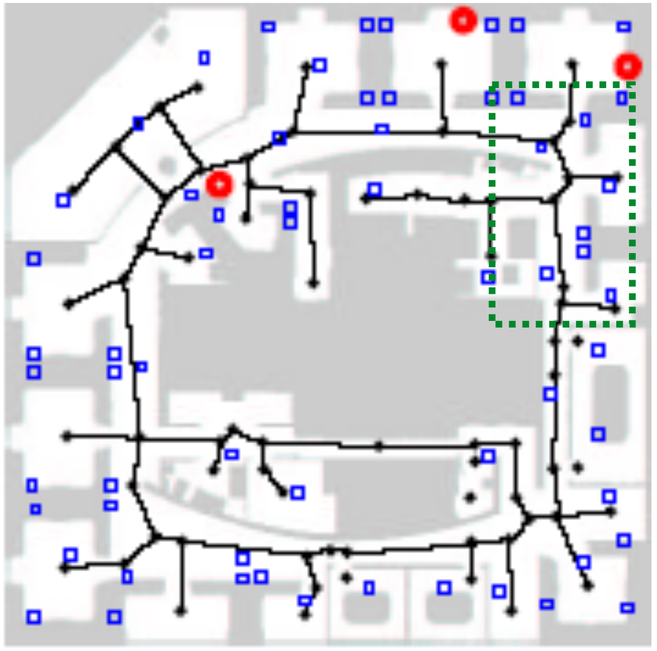
Approach 1: Discrete states, actions and observations

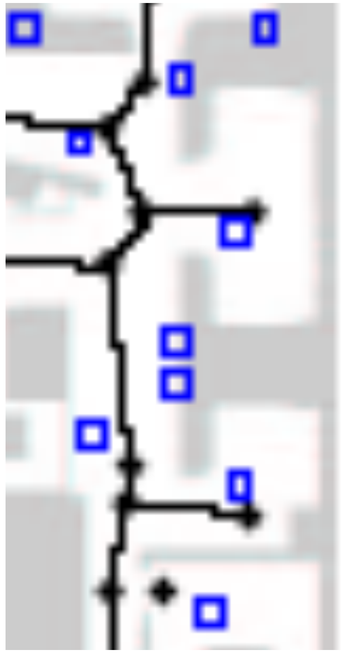
$$p(x_t) = p(z_t|x_t) \sum_X p(x_t|a_t, x_{t-1})\, p(x_{t-1}) \qquad \text{for all } x \in X$$

- Even when the state space is continuous, we discretize the state space (e.g., in a grid)

- Note that the run time of the algorithm is quadratic in the number of states in X.

Approach 1: Discrete states, actions and observations





- This kind of localization can be applied for localization in a *topological map*.

- **Robot pose** is the room the robot is in.
- **Measurements** are landmarks (e.g., corridor features, junctions, objects)

Discrete states, actions, and observations:

$$p(x_t) = p(z_t|x_t) \sum_X p(x_t|a_t, x_{t-1})\, p(x_{t-1}) \qquad \text{for all } x \in X$$

- Realistic examples include localization on a topological map!

Approach 1: Discrete states, actions and observations

- What are the problems with discretizing states, actions and observations themselves?

Approach 2: Finitely-parametrized Distributions

- The key example is the Kalman filter which assumes that the distribution is a Gaussian!

- Assumptions:
  - Linear process models
  - Gaussian initial state
  - Gaussian noise

- Pose at time *t*: x_t
- *Input* at time t: u_t
- Measurement at time t: z_t
- Process noise at time t: q_t
- Sensing noise at time t: r_t

$$x_t = A\,x_{t-1} + Bu_{t-1} + q_{t-1}$$
$$z_t = H\,x_t + r_t$$

- **Key result by Kalman:** Under these assumptions, the posterior pose belief is Gaussian.

Approach 2: Finitely-parametrized Distributions

- The key example is the Kalman filter which assumes that the distribution is a Gaussian!

- Assumptions:
  - Linear process models
  - Gaussian initial state
  - Gaussian noise

- Pose at time *t*: x_t
- *Input* at time t: u_t
- Measurement at time t: z_t
- Process noise at time t: q_t
- Sensing noise at time t: r_t

$$x_t = A\,x_{t-1} + Bu_{t-1} + q_{t-1}$$
$$z_t = H\,x_t + r_t$$

- **Key result by Kalman:** Under these assumptions, the posterior pose belief is Gaussian.

Kalman's result: We only need the first two moments.
Mean: $\mu_t = \mathbb{E}[x_t]$
Variance: $\Sigma_t = \mathbb{E}[(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T]$

1. **Process Model:**
$$\mu_t^- = A\,\mu_{t-1} + B\,u_{t-1}$$
$$\Sigma_t^- = A\,\Sigma_{t-1}\,A^T + Q$$

2. **Measurement Model:**
$$K_t = \Sigma_t^-\,H^T\,(H\,\Sigma_t^-\,H^T + R)^{-1}$$
$$\mu_t = \mu^- + K_t(z_t - H\,\mu_t^-)$$
$$\Sigma_t = (I - K_t\,H)\,\Sigma_t^-$$

Approach 2: Finitely-parametrized Distributions

- The key example is the Kalman filter which assumes that the distribution is a Gaussian!

- Pose at time *t*: x_t
- *Input* at time t: u_t
- Measurement at time t: z_t
- Process noise at time t: q_t
- Sensing noise at time t: r_t

$$x_t = A\,x_{t-1} + B u_{t-1} + q_{t-1}$$
$$z_t = H\,x_t + r_t$$

Mean: $\mu_t = \mathbb{E}[x_t]$
Variance: $\Sigma_t = \mathbb{E}[(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T]$

Kalman Filter

1. **Process Model:**
$$\mu_t^- = A\,\mu_{t-1} + B\,u_{t-1}$$
$$\Sigma_t^- = A\,\Sigma_{t-1}\,A^T + Q$$

2. **Measurement Model:**
$$K_t = \Sigma_t^-\,H^T\,(H\,\Sigma_t^-\,H^T + R)^{-1}$$
$$\mu_t = \mu^- + K_t(z_t - H\,\mu_t^-)$$
$$\Sigma_t = (I - K_t\,H)\,\Sigma_t^-$$

Approach 2: Finitely-parametrized Distributions

- What if the dynamics model and measurement model are nonlinear?

- Pose at time *t*: x_t
- Input at time t: u_t
- Measurement at time t: z_t
- Process noise at time t: q_t
- Sensing noise at time t: r_t

Nonlinear model

$$x_t = f(x_{t-1}, u_{t-1}, q_{t-1})$$
$$z_t = h(x_t, r_t)$$

Linearized model

$$\tilde{x}_t = f(x_{t-1}, u_{t-1}, 0)$$
$$x_t \approx \tilde{x}_t + A\,(x_{t-1} - \tilde{x}_{t-1}) + W\,q_{t-1}$$
$$z_t \approx h(\tilde{x}_t, 0) + H\,(x_t - \tilde{x}_t) + V\,r_t$$

Linear Kalman Filter

1. **Process Model:**

$$\mu_t^- = A\,\mu_{t-1} + B\,u_{t-1}$$
$$\Sigma_t^- = A\,\Sigma_{t-1}\,A^T + Q$$

2. **Measurement Model:**

$$K_t = \Sigma_t^-\,H^T\,(H\,\Sigma_t^-\,H^T + R)^{-1}$$
$$\mu_t = \mu^- + K_t(z_t - H\,\mu_t^-)$$
$$\Sigma_t = (I - K_t\,H)\,\Sigma_t^-$$

Extended Kalman Filter

1. **Process Model:**

$$\mu_t^- = f(\mu_{t-1}, u_{t-1}, 0)$$
$$\Sigma_t^- = A_t\,\Sigma_{t-1}\,A_t^T + W_t\,Q_t\,W_t^T$$

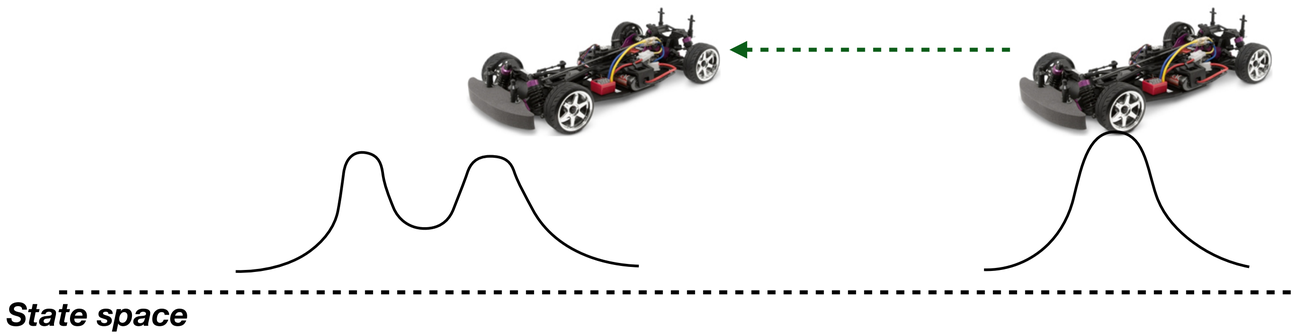
2. **Measurement Model:**

$$K_t = \Sigma_t^-\,H_t^T\,(H_t\,\Sigma_t^-\,H_t^T + V_t\,R_t\,V_t^T)^{-1}$$
$$\mu_t = \mu_t^- + K_t\,(z_t - h(\mu_t^-, 0))$$
$$\Sigma_t = (I - K_t\,H_t)\,\Sigma_t^-$$

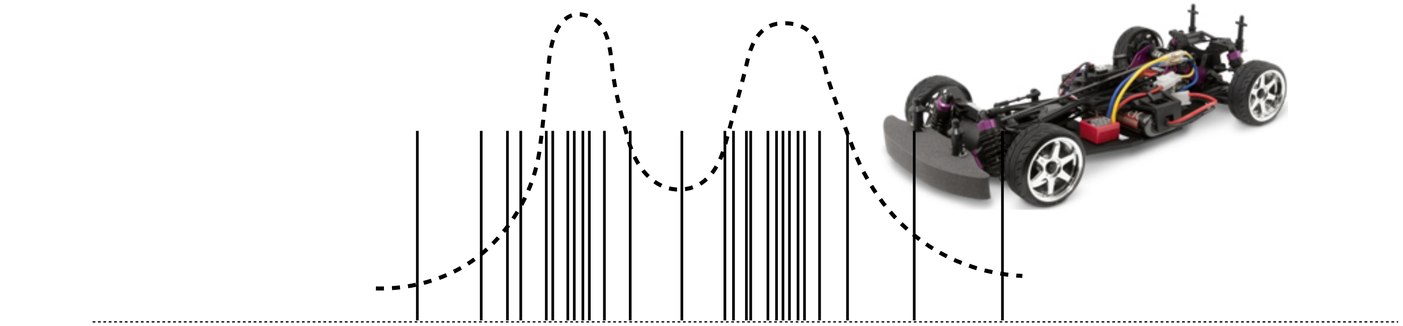
- Extended Kalman filter uses the nonlinear model to handle the mean.

Approach 2: Finitely-parametrized Distributions

- What are the problems with Kalman filters?
  - Recall the assumptions of the (linear) Kalman Filter:
    - Linear process models
    - Gaussian initial state
    - Gaussian noise
  - But, the pose belief is still modeled as a Gaussian.
    Why is this a problem?



Approach 3: Monte Carlo Localization / Particle Filter



State space

- **Key idea:** Represent the belief distribution as a set of particles.
  - These particles must simulate a set that would be sampled from the underlying distribution!

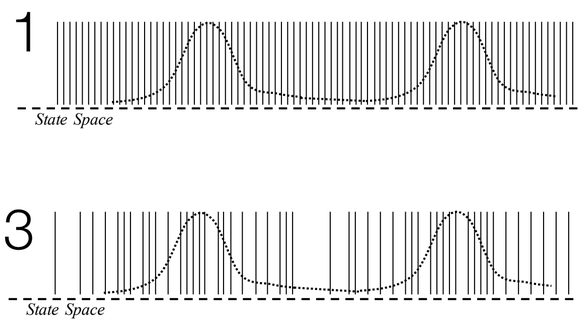
- "propagate" particles forward according to the action and the dynamics model.
- "update the likelihood" of a particle according to the measurement and the measurement model.

Approach 3: Monte Carlo Localization / Particle Filter

- **Key idea:** Represent the belief distribution as a set of particles.
  - These particles must simulate a set that would be sampled from the underlying distribution!

- "propagate" particles forward according to the action and the dynamics model.
- "update the likelihood" of a particle according to the measurement and the measurement model.

- **Rejection sampling:** To sample from p(.)
  1. Sample *x* from the uniform distribution
  2. Compute the weight *w(x)* of each sample as its probability according to *p(.)*
  3. Keep particles with probability *w(x)* hence, reject with probability *1-w(x)*



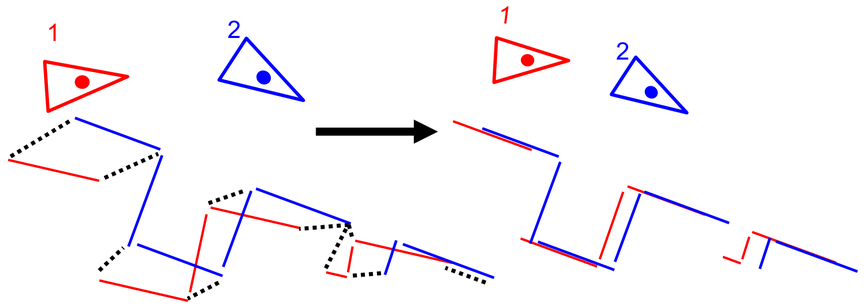
Approach 3: Monte Carlo Localization / Particle Filter

How does it match the scans?

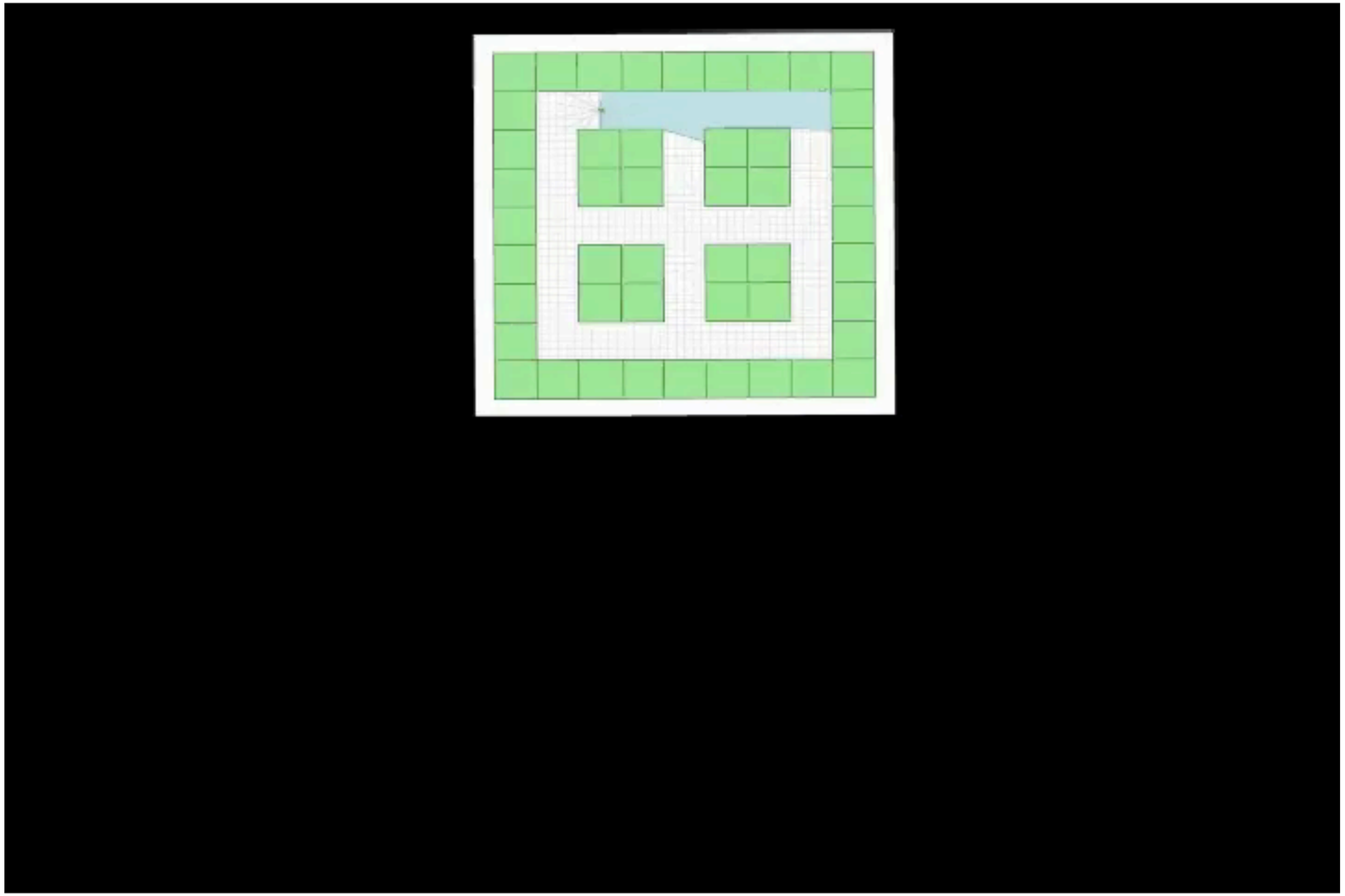
Iterative Closest Point (ICP)

- **In iterations:**
  - **For each point in scan 1, find the closest point in scan 2 (how?)**
  - **Find the transformation that best aligns the matching sets of points**

- **Repeat until convergence.**



Simultaneous Localization and Mapping (SLAM)



Top:
ground truth

Left:
PF-SLAM

Right:
mapping
from
odometry

Bayes Filter for Localization and for SLAM

Bayes Filter for Localization

Posterior pose belief

$$p(x_t|a_1, z_1, a_2, z_2, \ldots, a_t, z_t)$$

$$= \alpha\, p(z_t|x_t) \int_X p(x_t|a_t, x_{t-1})\, p(x_{t-1}|a_1, z_1, a_2, z_2, \ldots, a_{t-1}, z_{t-1}) dx_{t-1}$$

Measurement model

Dynamic model

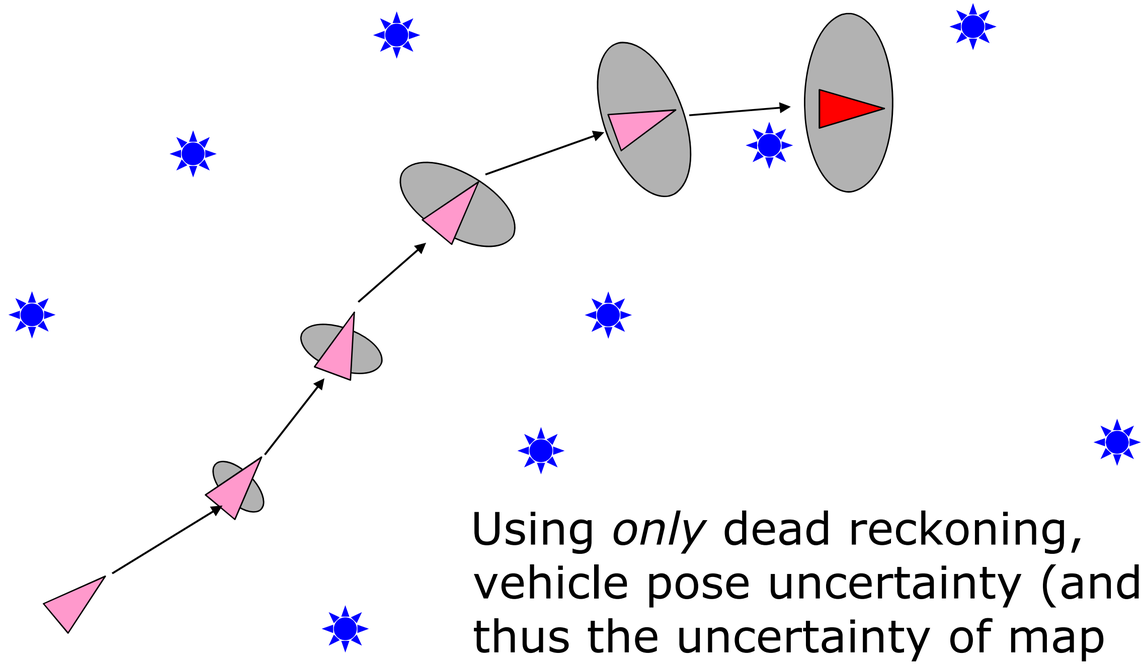
Initial pose belief

Bayes Filter for SLAM

$$p(x_t, m_t|a_1, z_1, \ldots, a_t, z_t)$$

$$= \alpha\, p(z_t|x_t, m_t) \int_X p(x_t, m_t|a_t, x_{t-1}, m_{t-1}) p(x_{t-1}, m_{t-1}|a_1, z_1, \ldots, a_{t-1}, z_{t-1}) d(x_{t-1}, m_{t-1})$$

Localization and then Mapping (not Simultaneously)



Using *only* dead reckoning, vehicle pose uncertainty (and thus the uncertainty of map features) grows without bound