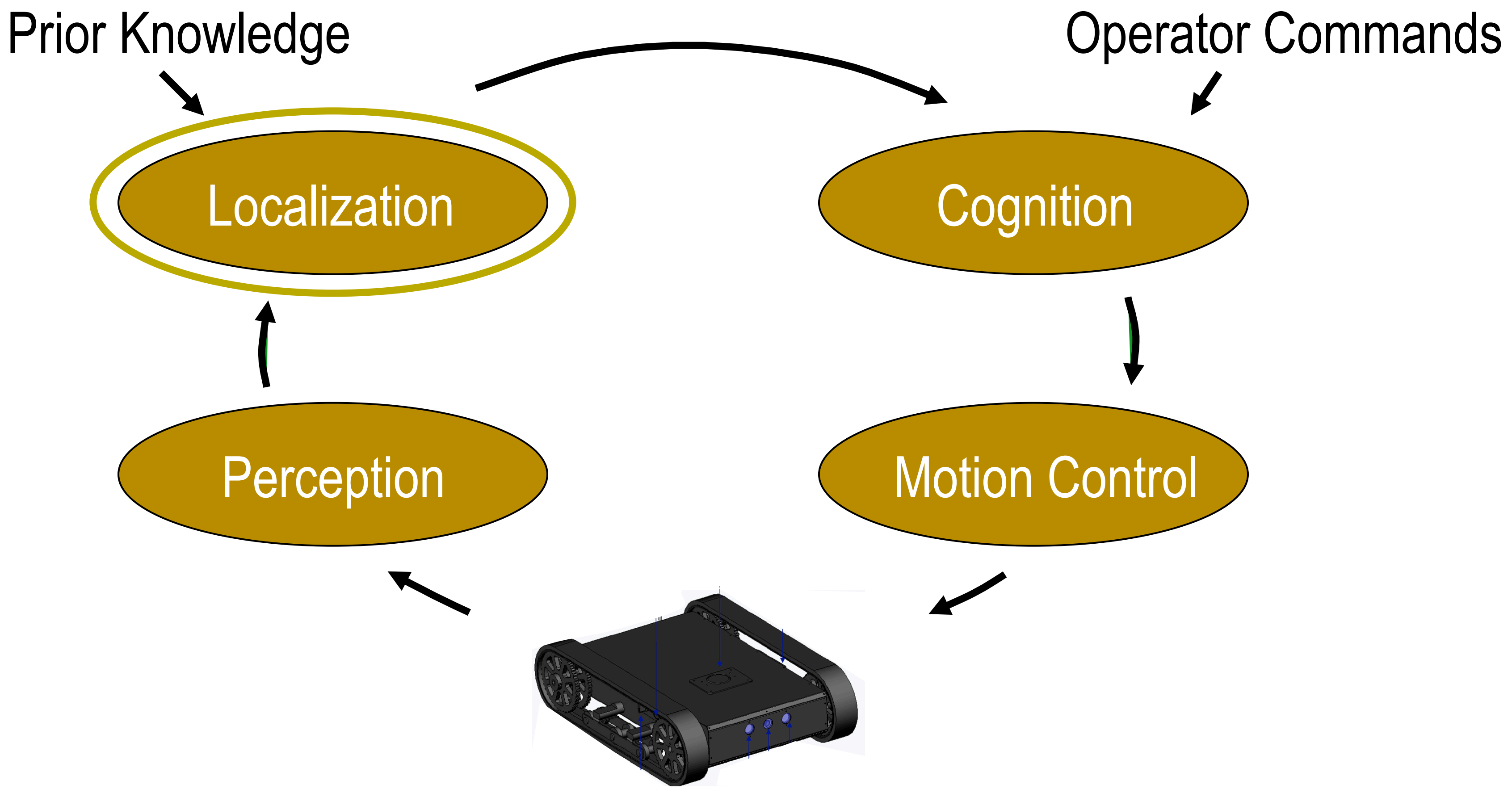
E160 – Lecture 9
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Spring 2016

Figures courtesy of Siegwart & Nourbakhsh

Control Structures
Planning Based Control

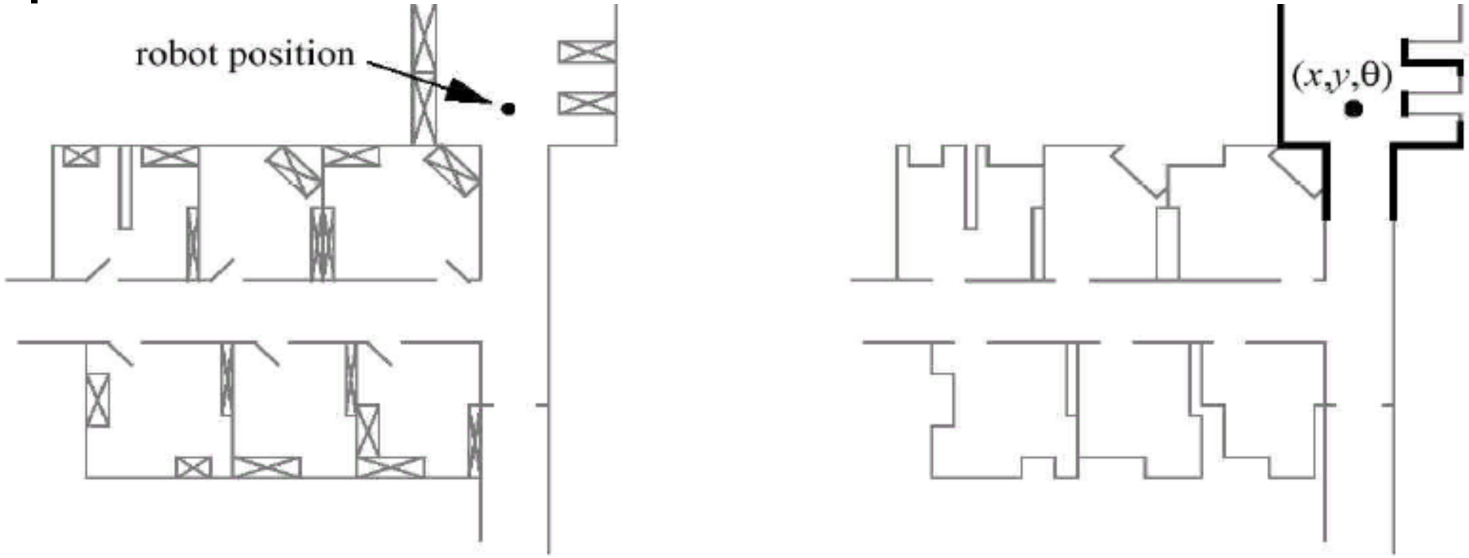


Outline - Localization

1. Localization Tools
   - Belief representation
   - Map representation
2. Overview of Algorithms
3. Markov Localization

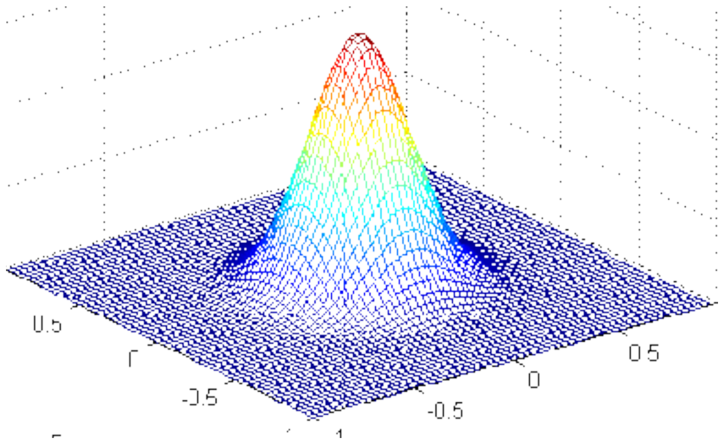
Belief Representation

- Our belief representation refers to the method we describe our estimate of the robot state.
- So far we have been using a **Continuous** Belief representation.



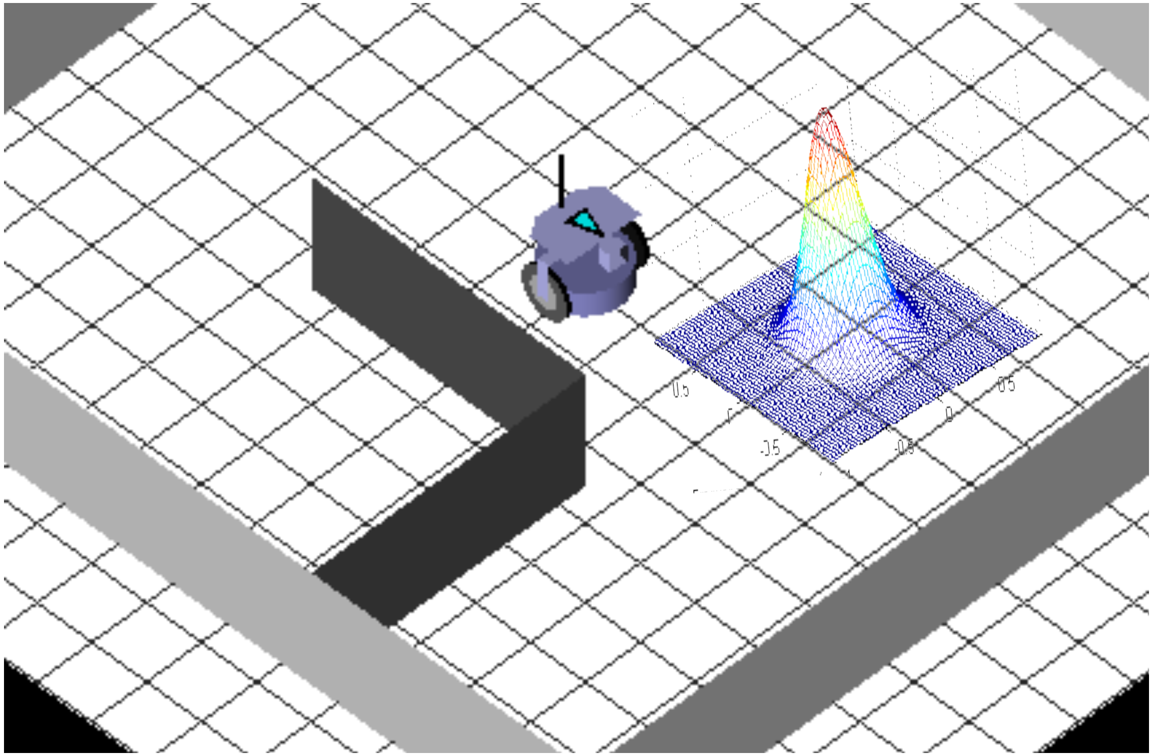
Belief Representation

- We can provide a description of the level of confidence we have in our estimate.
  - We typically use a Gaussian distribution to model the state of the robot.
  - We need to know the variance of this Gaussian!



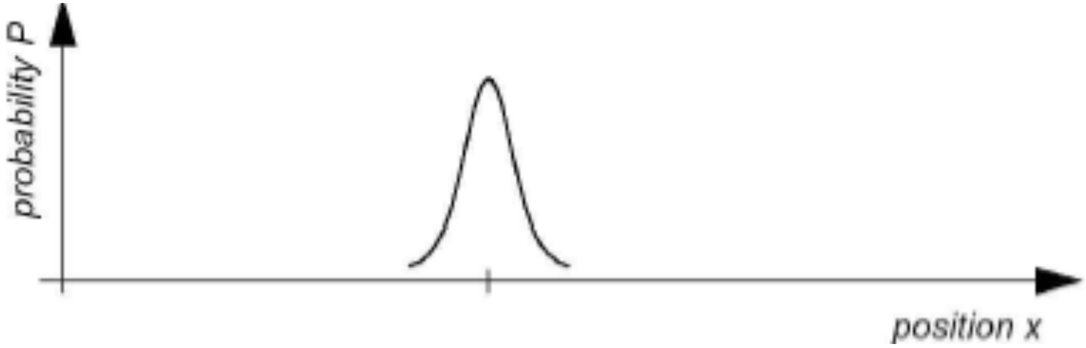
Belief Representation

- For example, consider modeling our robot's position with a 2D Gaussian:

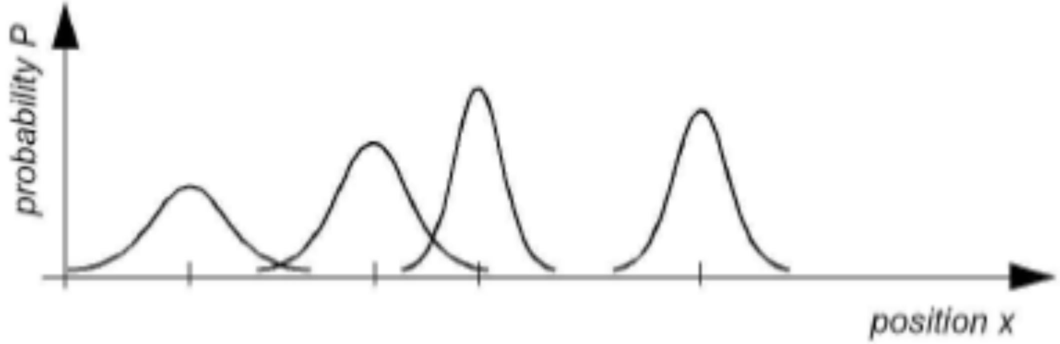


Belief Representation

- Continuous (single hypothesis)



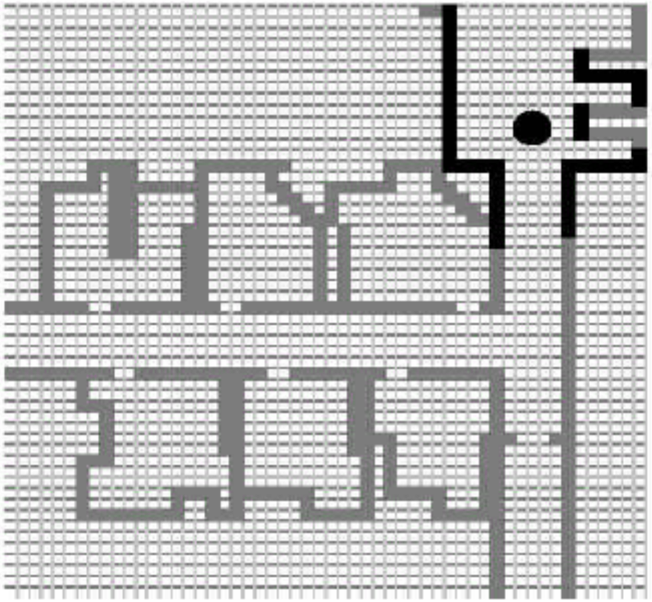
- Continuous (multiple hypothesis)



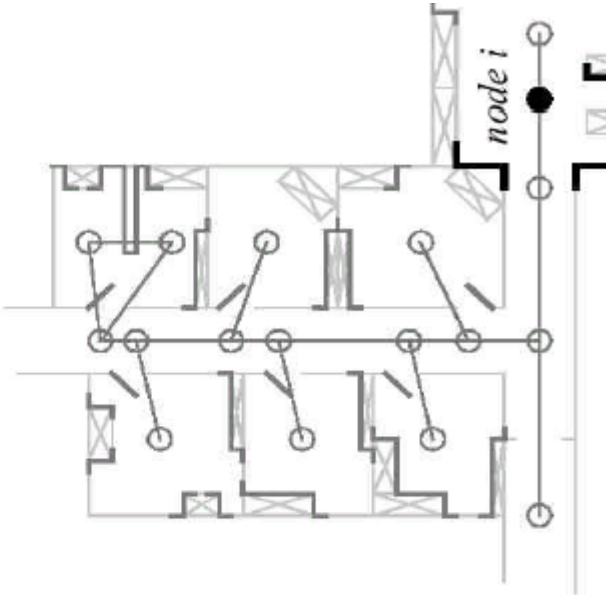
Belief Representation

- Or, we could assign a probability of being in some discrete locations:

Grid

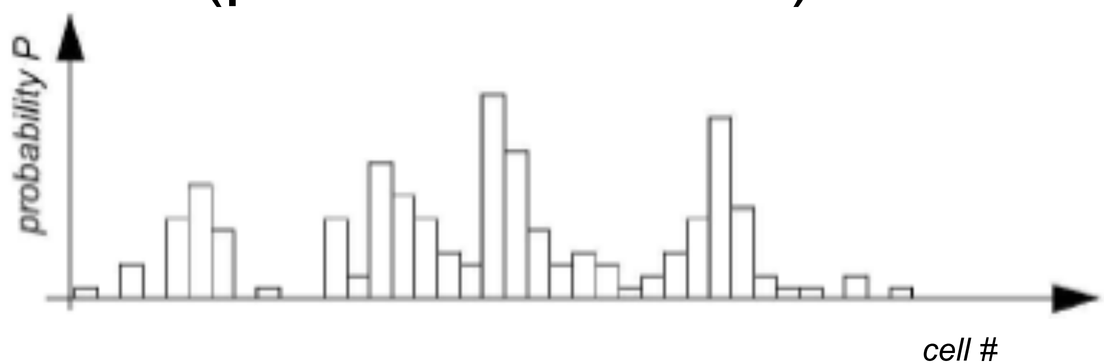


Topological

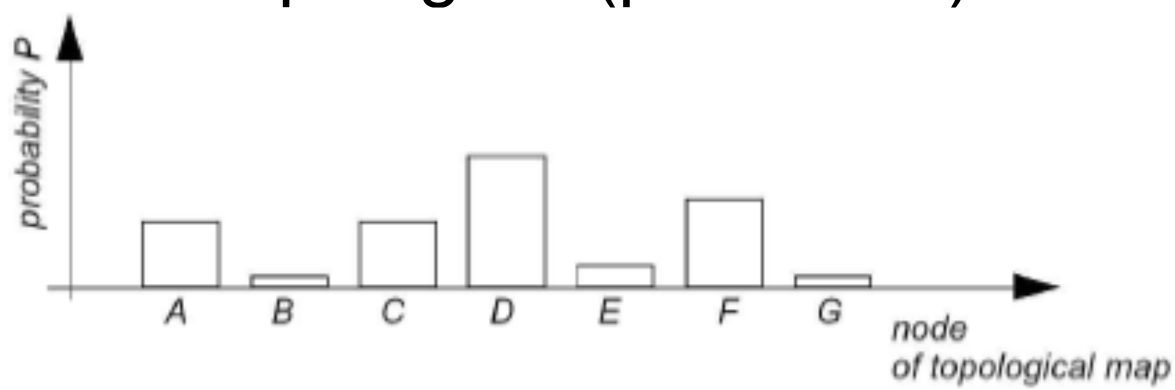


Belief Representation

- Discretized (prob. Distribution)

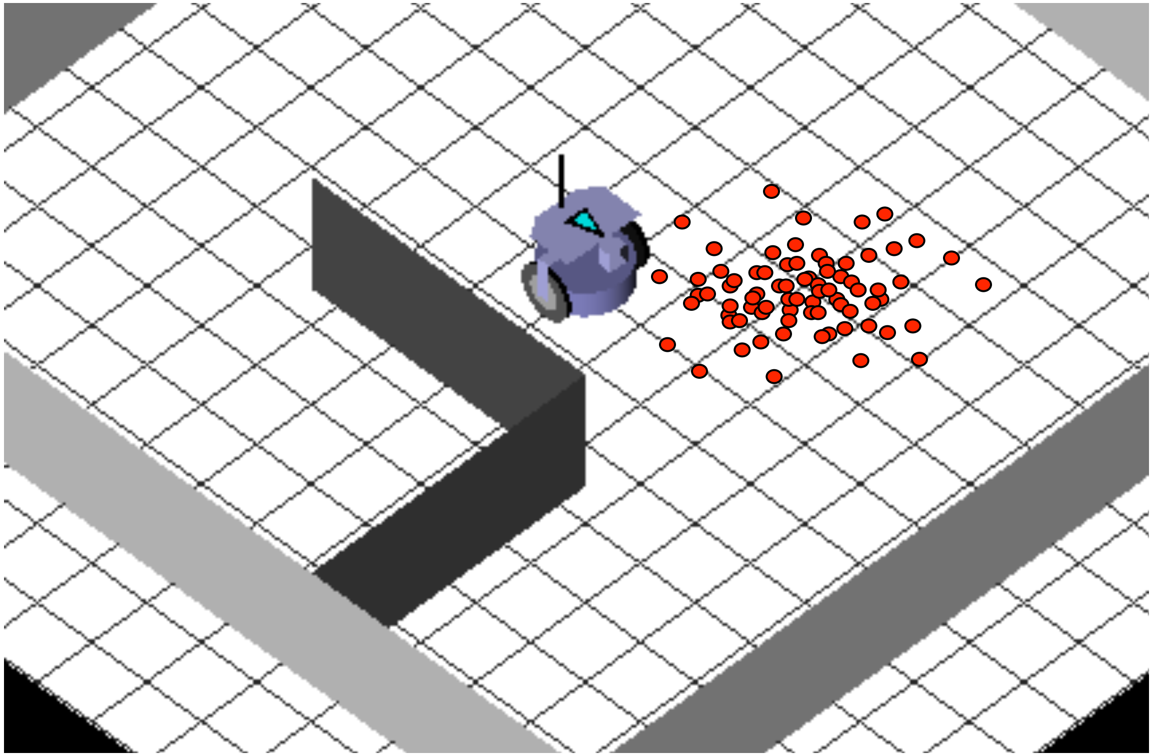


- Discretized Topological (prob. dist.)



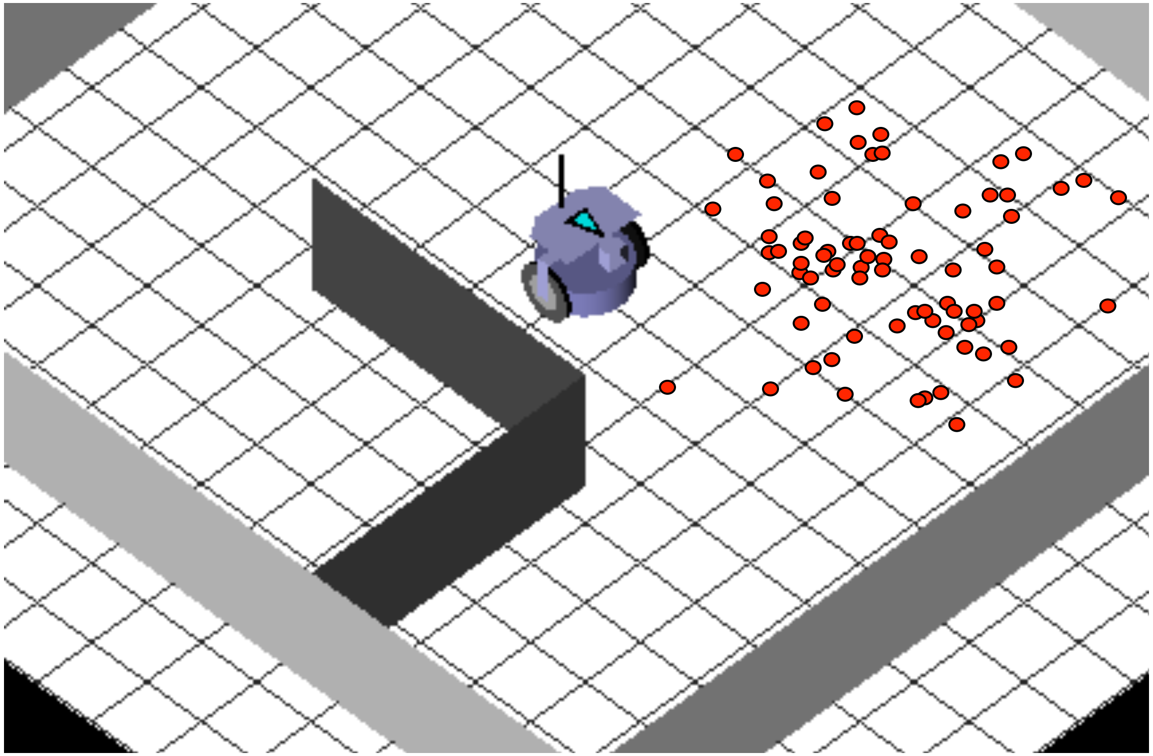
Belief Representation

- Discretized: Particles



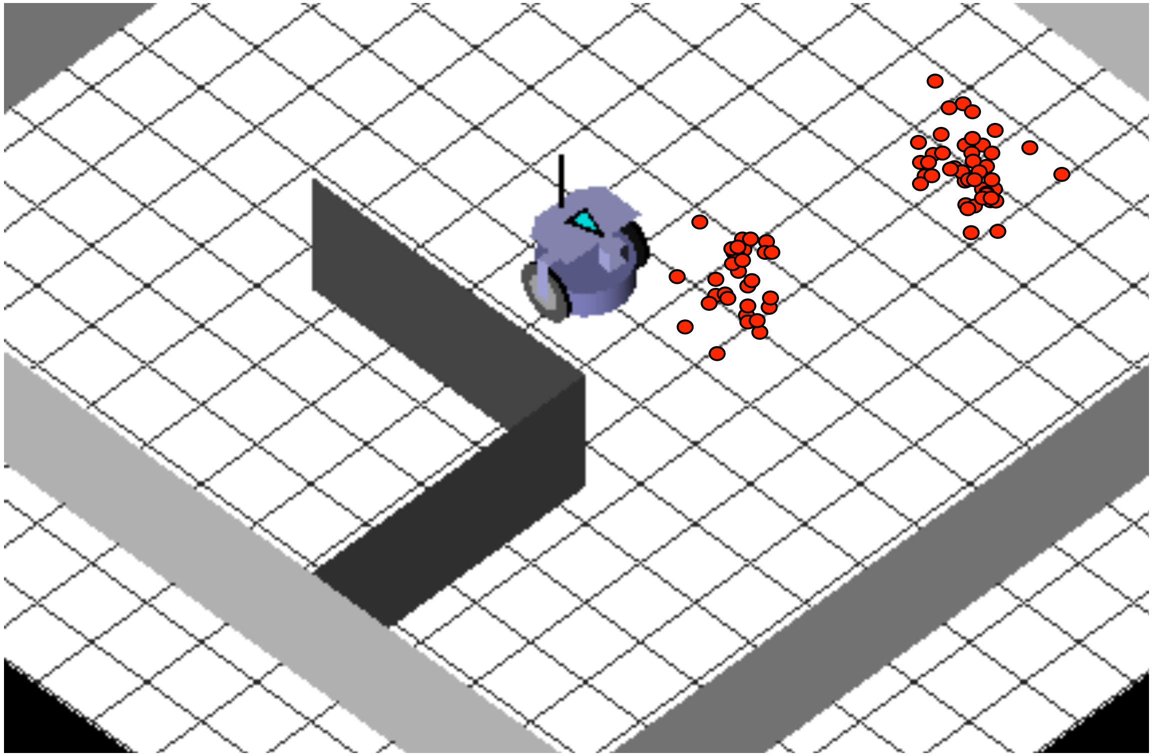
Belief Representation

- Discretized: Particles



Belief Representation

- Discretized: Particles



Belief Representation

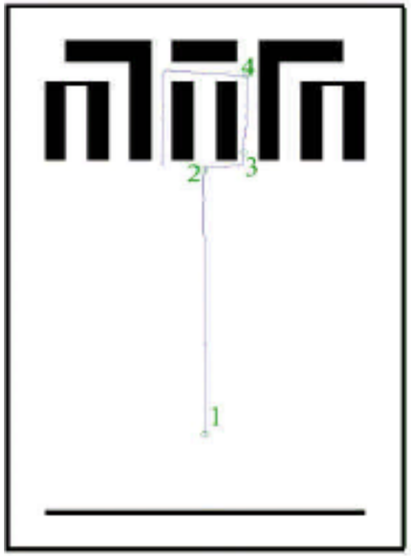
- Continuous
  - Precision bound by sensor data
  - Typically single hypothesis pose estimate
  - Lost when diverging (for single hypothesis)
  - Compact representation
  - Reasonable in processing power

Belief Representation

- Discrete
  - Precision bound by resolution of discretization
  - Typically multiple hypothesis pose estimate
  - Rarely lost (when diverges/converges to another cell).
  - Memory and processing power needed (unless topological map used)
  - Aids discrete planner implementation

Belief Representation

- Multi-Hypothesis Example



Path of the robot

Belief states at positions 2, 3 and 4

Outline - Localization

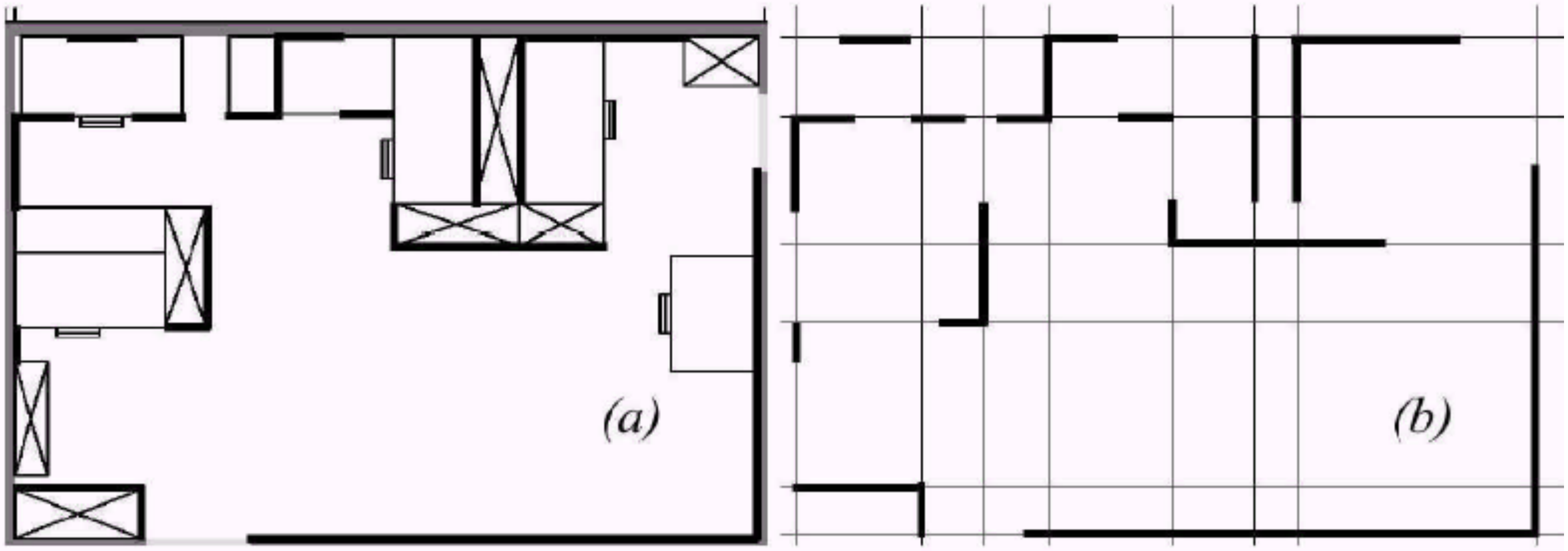
1. Localization Tools
   - Belief representation
   - Map representation
2. Overview of Algorithms
3. Markov Localization

Map Representation

- Similar to belief representations, there are two main types:
  - Continuous
  - Discretized

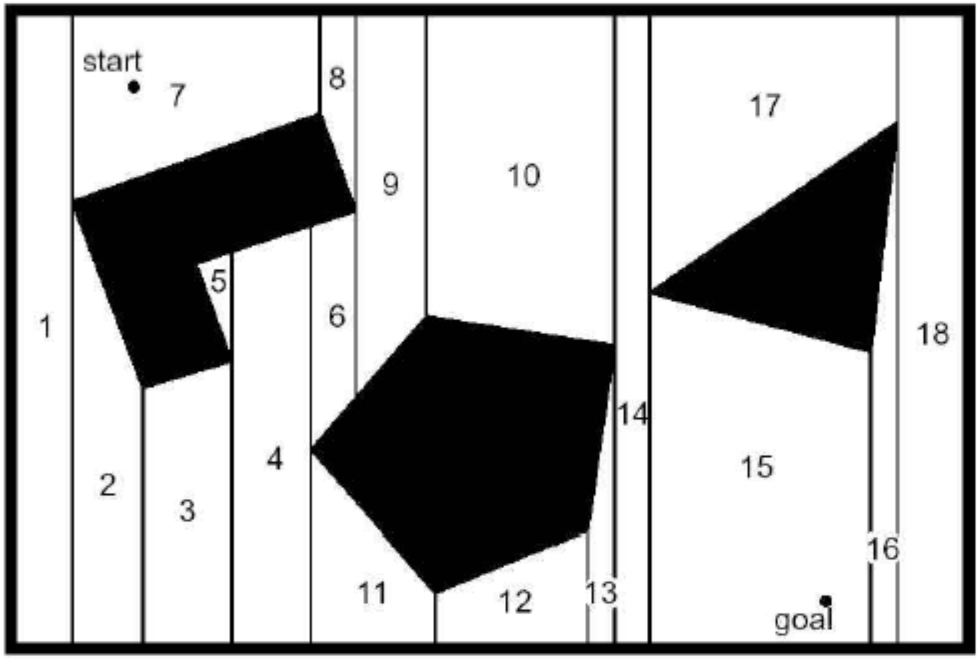
Map Representation

- Continous line-based



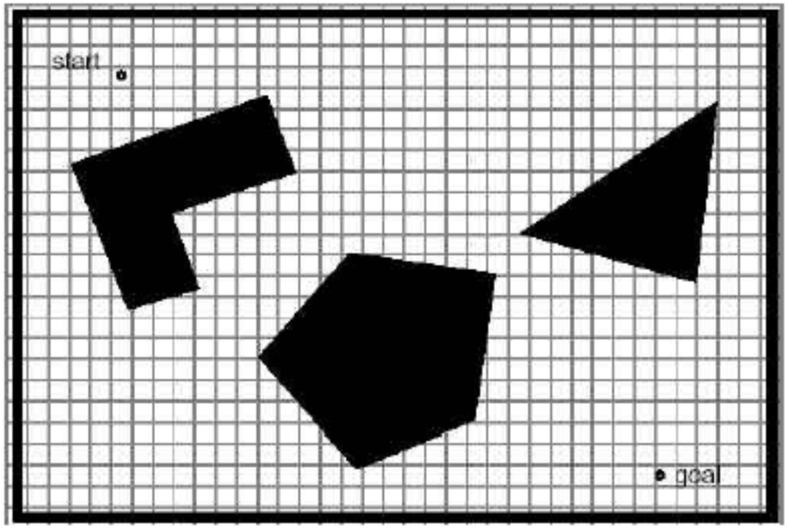
Map Representation

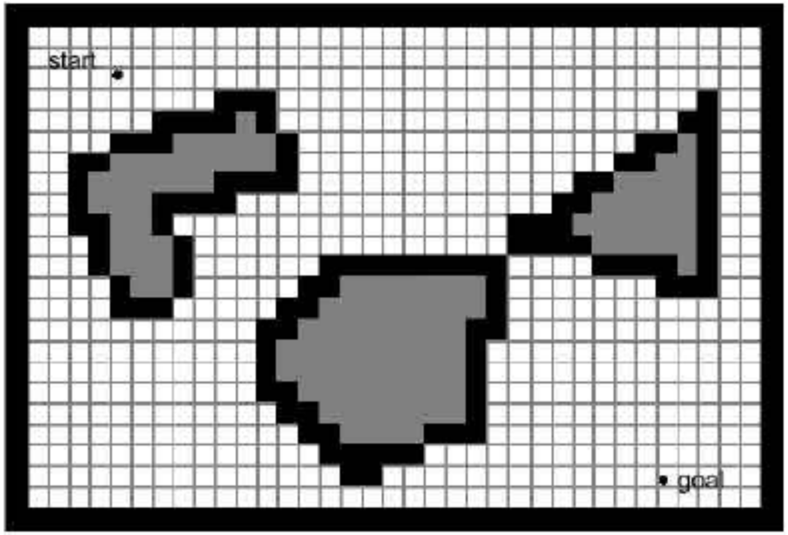
- Exact cell decomposition



Map Representation

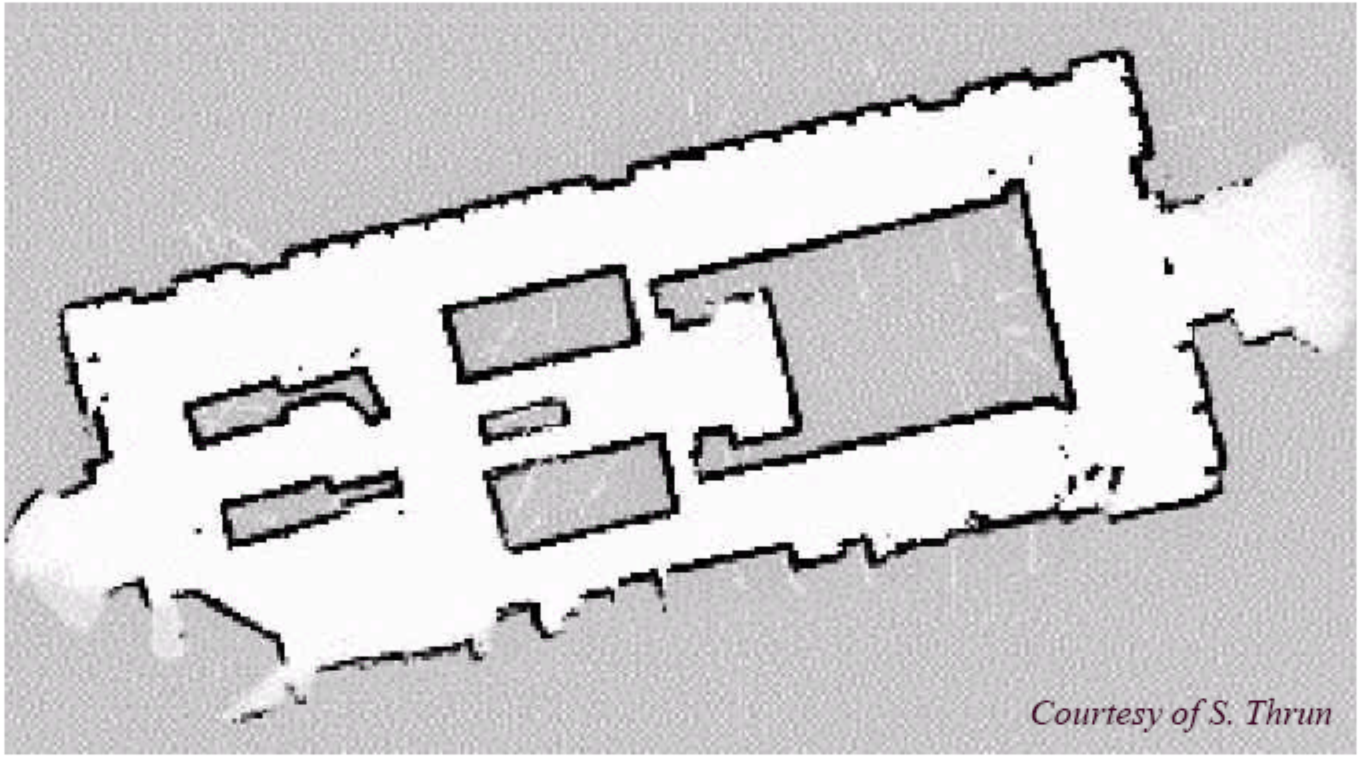
- Fixed cell decomposition





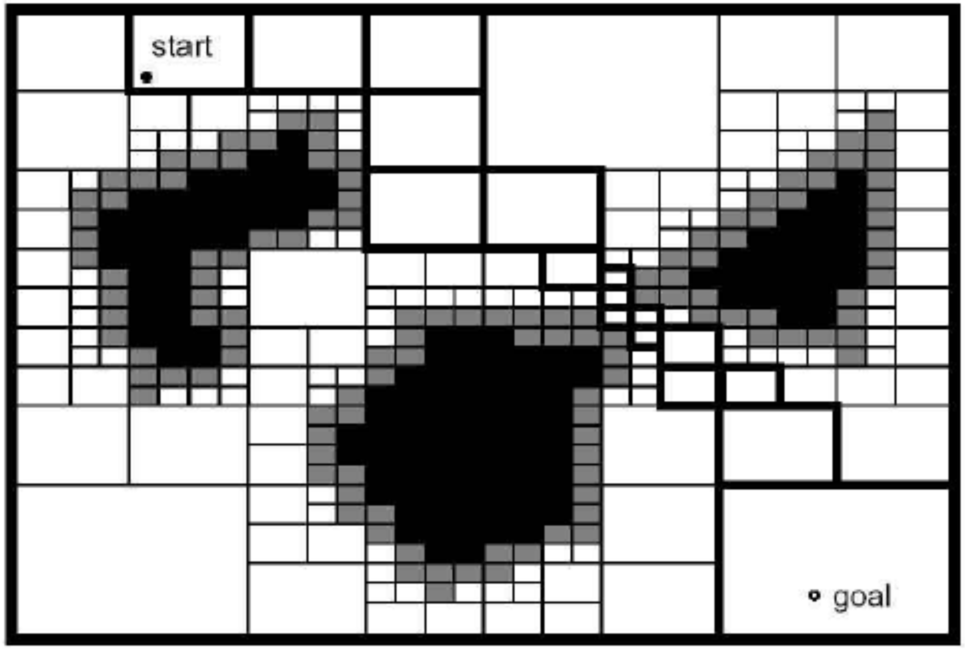
Map Representation

- Fixed cell decomposition



Map Representation

- Adaptive cell decomposition

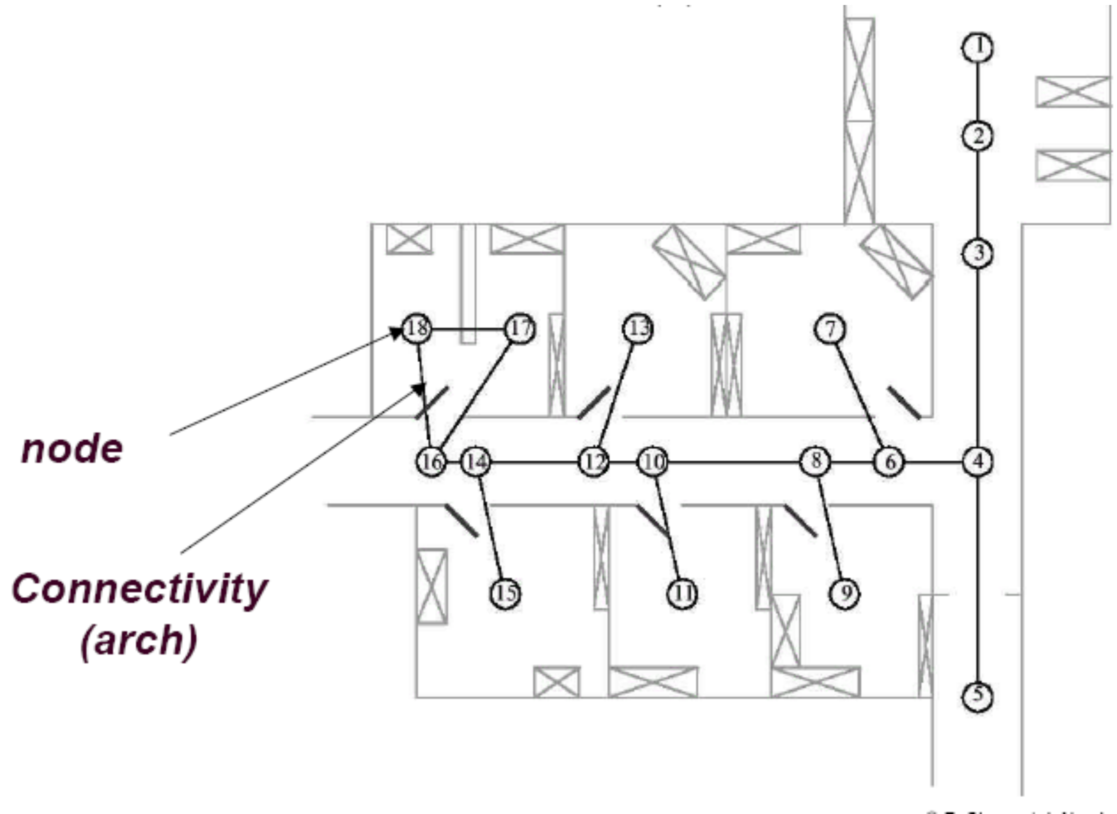


Map Representation

- Topological decomposition

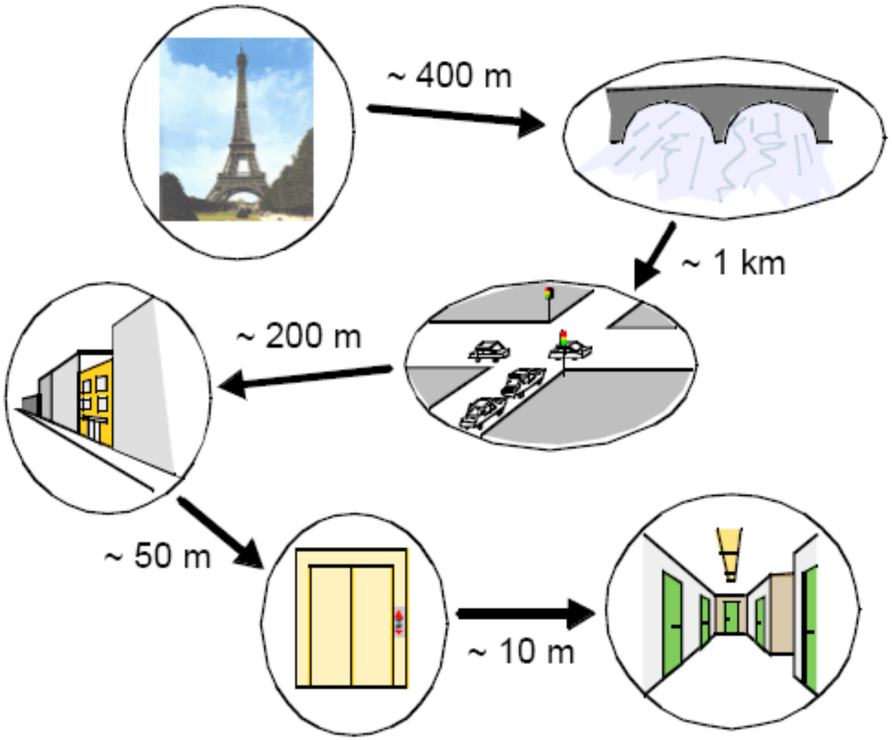
Map Representation

- Topological decomposition



Map Representation

- Topological decomposition



Outline - Localization

1. Localization Tools
2. Overview of Algorithms
   - Typical Methods
   - Basic Structure
3. Markov Localization

Methods

- Mapping Problem
  - Determine the state of the environment given a known robot state.
- Localization Problem
  - Determine the state of a robot given a known environment state.

Methods

- Strategy:
  - It might start to move from a known location, and keep track of its position using odometry.
  - However, the more it moves the greater the uncertainty in its position.
  - Therefore, it will update its position estimate using observation of its environment

Methods

- Method:
  - Fuse the odometric position estimate with the observation estimate to get best possible update of actual position

- This can be implemented with two main functions:

1. Act
2. See

Methods

- Action Update (Prediction)
  - Define function to predict position estimate based on previous state x_{t-1} and encoder measurement o_t or control inputs u_t

$$x'_t = Act\,(o_t\,,\,x_{t-1})$$

  - Increases uncertainty

Methods

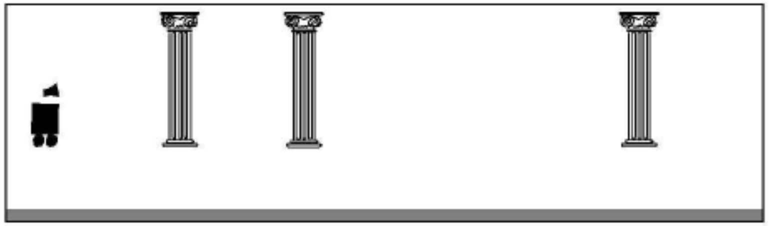
- Perception Update (Correction)
  - Define function to correct position estimate x'_t using exteroceptive sensor inputs z_t

$$x_t = See\,(z_t\,,\,x'_t)$$

  - Decreases uncertainty

Methods

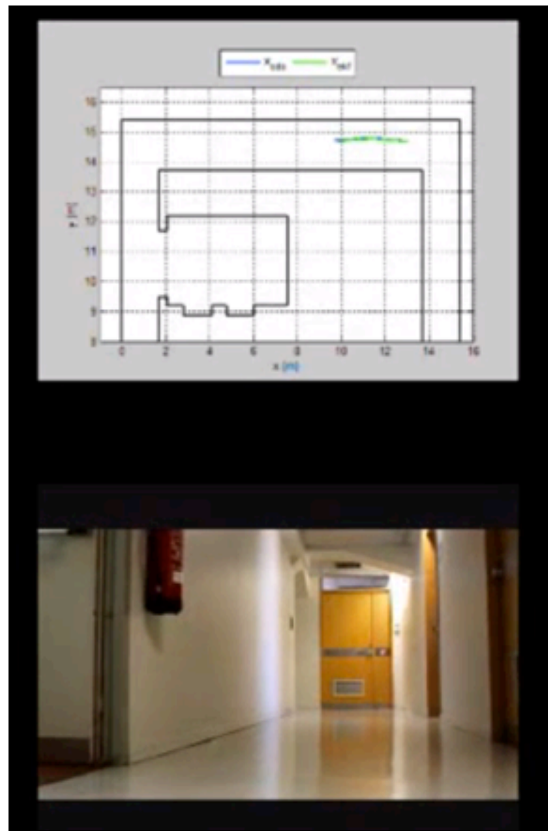
- Motion generally improves the position estimate.



Kalman Filtering vs. Markov

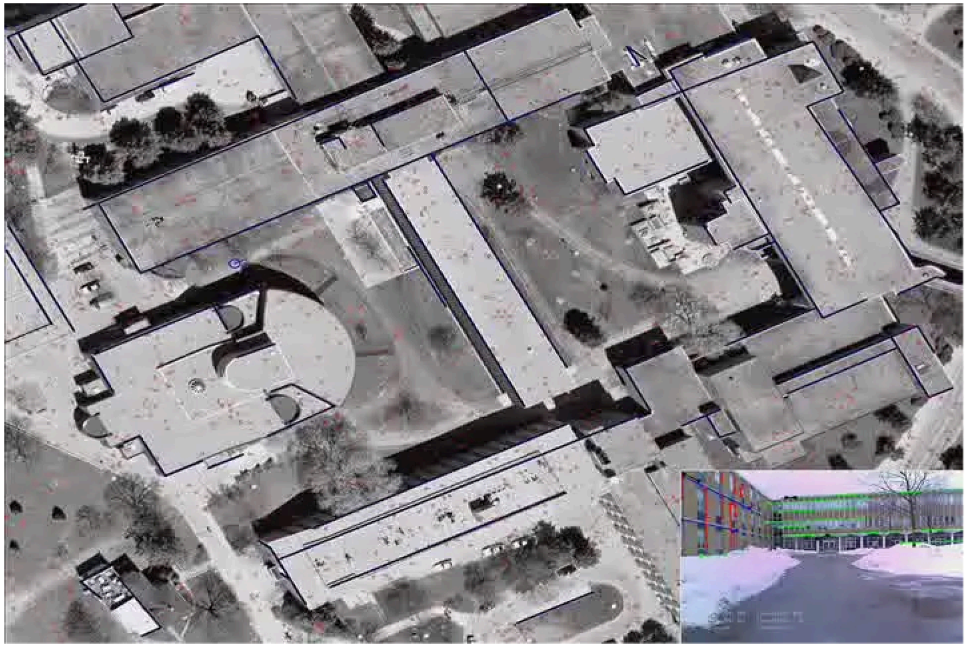
- Markov Localization
  - Can localize from **any unknown** position in map
  - **Recovers** from ambiguous situation
  - However, to update the probability of all positions within the whole state space requires discrete representation of space. This can require large amounts of **memory** and **processing** power.
- Kalman Filter Localization
  - Tracks the robot and is inherently **precise** and **efficient**
  - However, if uncertainty grows too large, the KF will fail and the robot will get **lost**.

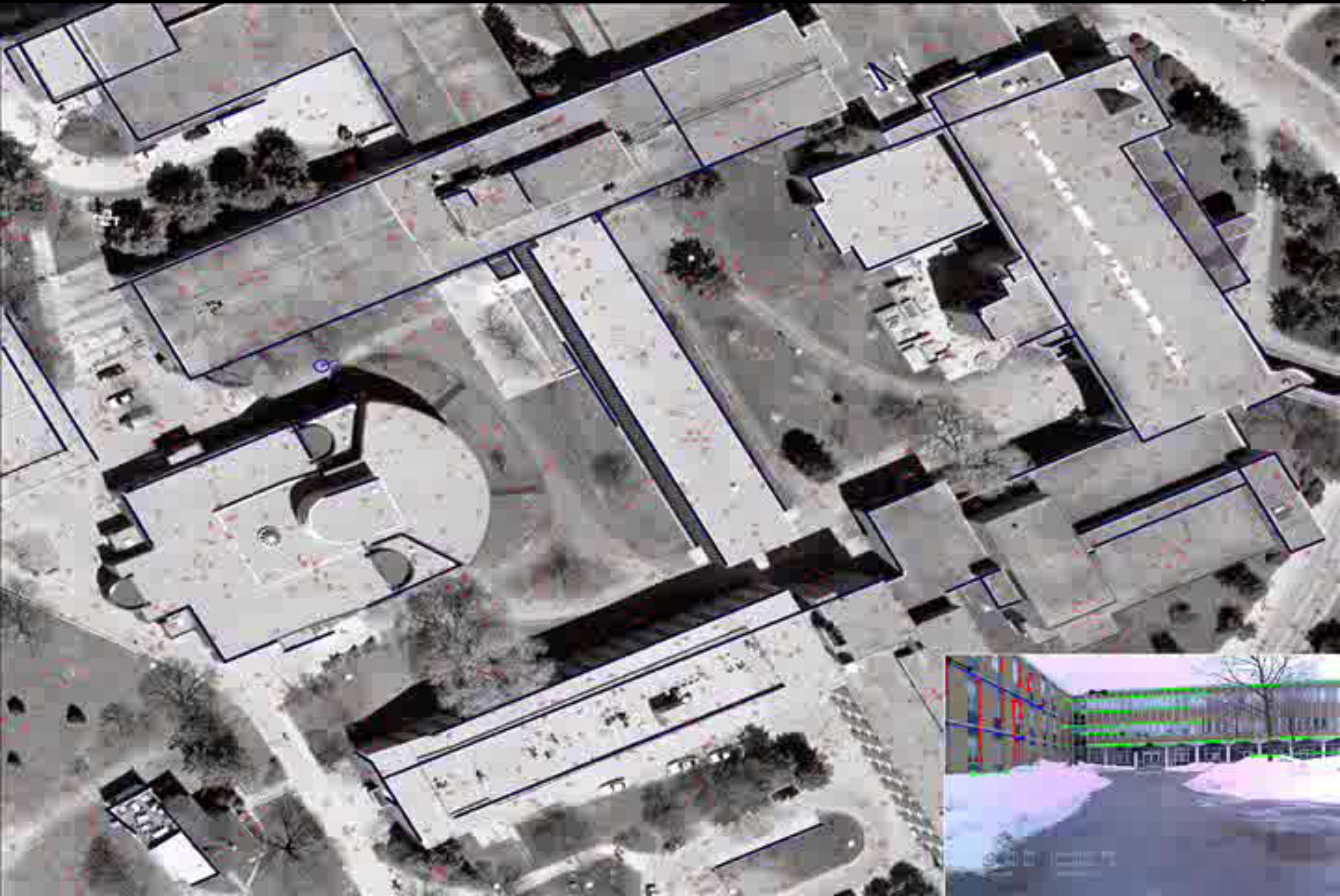
Kalman Filtering



http://www.youtube.com/watch?v=AXGXfD1GMY4

Particle Filter Localization





Outline - Localization

1. Localization Tools
2. Overview of Algorithms
3. Markov Localization
   - Overview
   - Prediction Step
   - Correction Step
   - ML Example

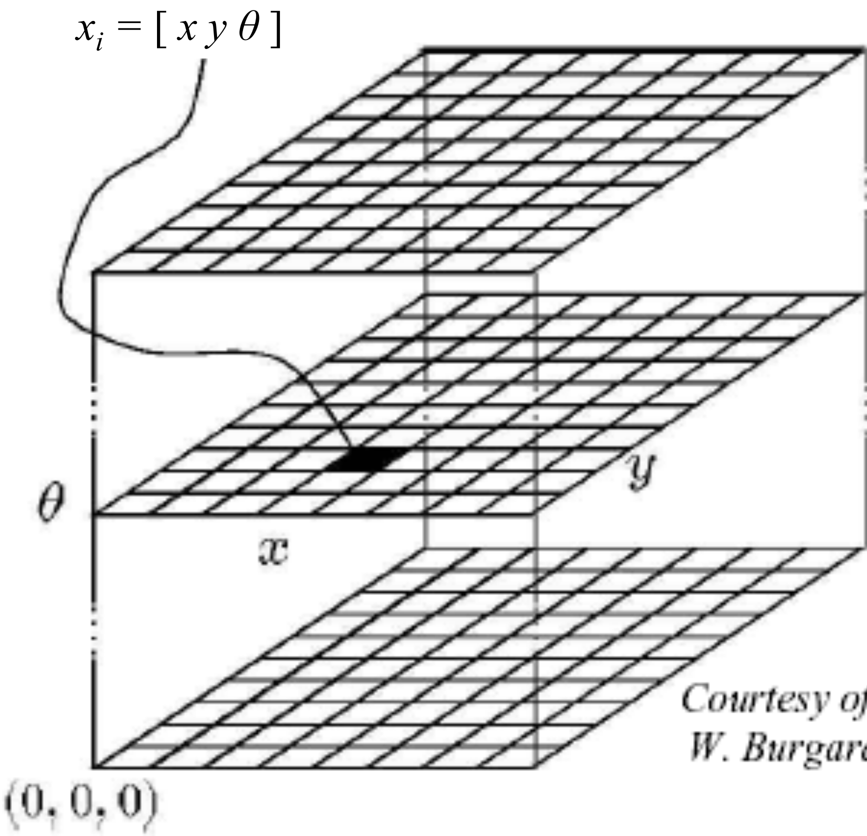
Markov Localization

- Markov localization uses an explicit, **discrete** representation for the probability of all positions in the state space.
- Usually represent the environment by a finite number of (states) positions:
  - Grid
  - Topological Map

Markov Localization
Grid Based Example

- Use a fixed decomposition grid by discretizing each dof:

  (x, y, θ)
  - For each location $x_i = [\, x\ y\ \theta \,]$ in the configuration space:
  - Determine probability $P(x_i)$ of robot being in that state.



Courtesy of W. Burgard

Markov Localization

- We assume in localization the Markov Property holds true…

- Markov Property
  - A stochastic Process satisfies the Markov Property if it is conditional **only on the present state of the system, and its future and past are independent**
  - The robot state x_t only depends on previous state x_{t-1} and most recent actions u_t and measurements z_t

Markov Localization

- Algorithm PseudoCode to update all *n* states

for $i = 1{:}n$
    $P(x_i) = 1/n$

while (true)
    o = getOdometryMeasurements
    z = getRangeMeasurements
    for $i = 1{:}n$
        $P(x_i')$ = predictionStep($P(x_j)$, o)
    for $i = 1{:}n$
        $P(x_i)$ = correctionStep($P(x_i')$, z)

Markov Localization
Applying Probability Theory

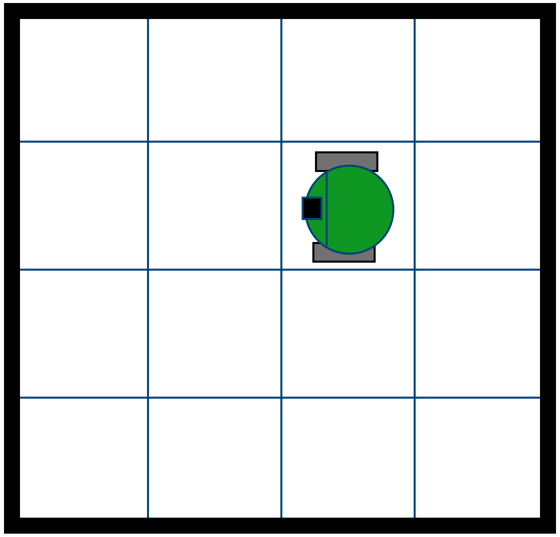
1. PREDICTION Step: Updating the belief state

$$P(x_{i,t}') = P(x_{i,t} \mid o_t)$$
$$= \sum_{j=1}^{n} P(x_{i,t} \mid x_{j,t-1}, o_t)\, P(x_{j,t-1})$$

- Map from a belief state $P(x_{j,t-1})$ and action o_t to a new predicted belief state $P(x_{i,t}')$
- Sum over all possible ways (i.e. from all states $x_{j,t-1}$) in which the robot may have reached $x_{i,t}'$
- This assumes that update only depends on previous state and most recent actions/perception

Markov Localization
Grid Based Example

- Example Problem:
  - *Consider a robot equipped with encoders and a perfect compass moving in a square room that is discretized into a map of 16 cells:*



Markov Localization
Grid Based Example

- Example Problem:
  - *What is the probability of being in position (2,3) given odometry $o_t = (\Delta x, \Delta y) = (-1.0 \text{ cells}, 0.0 \text{ cells})$, and starting from the following distribution?*

.02	.05	.05	.05
.02	.05	.18	.05
.05	.05	.18	.05
.05	.05	.05	.05

Markov Localization
Grid Based Example

- Example Solution:
  - *We must have a model of how well our odometry works. For example, we could use a model for* $o_t = (\Delta x, \Delta y) = (-1.0, 0.0)$ *that looks like:*

.00	.00	.00
.00	.00	1.0
.00	.00	.00

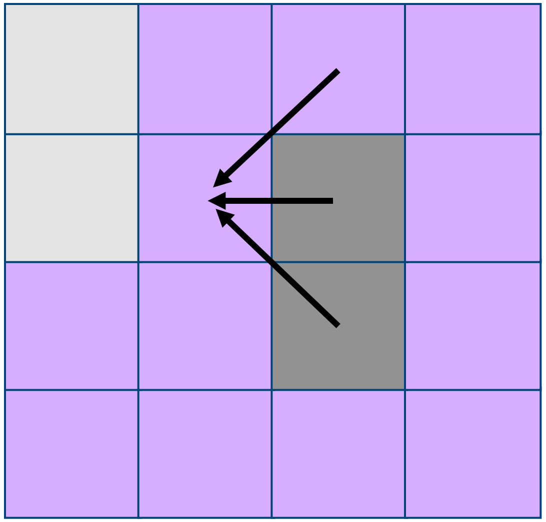
$(\Delta x, \Delta y)$ →

.00	.20	.00
.00	.50	.10
.00	.20	.00

Markov Localization
Grid Based Example

- Example Solution:
  - *Now apply this model to the initial state. We must consider the following possible scenarios for getting to position (2,3):*

$$(3,3) \rightarrow (2,3)$$
$$(2,3) \rightarrow (2,3)$$
$$(3,2) \rightarrow (2,3)$$
$$(3,4) \rightarrow (2,3)$$



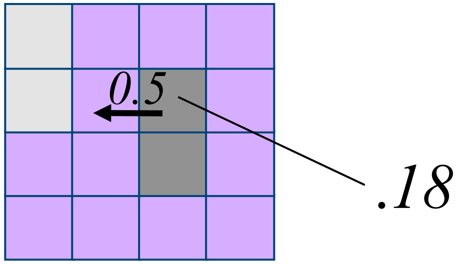
Markov Localization
Grid Based Example

- Example Solution:
  - *Consider the first possibility:*

    $(3,3) \rightarrow (2,3)$

  - *We can calculate the probability of this happening*

$$P(x_{i,t} \mid x_{j,t-1}, o_t)\, P(x_{j,t-1})$$
$$= P(x_t=(2,3) \mid x_{t-1}=(3,3), o_t=(-1,0))\;\; P(x_{t-1}=(3,3))$$
$$= (0.5)\,(0.18)$$
$$= 0.09$$



Markov Localization
Grid Based Example

- Example Solution:
  - *Similarly, we can calculate the probability of all other possible ways to get to (2,3).*

$P(x_t=(2,3) \mid x_{t-1}=(2,3) , o_t=(-1,0))\ P(x_{t-1}= (2,3))$

$= 0.005$

$P(x_t=(2,3) \mid x_{t-1}=(3,2) , o_t=(-1,0))\ P(x_{t-1}= (3,2))$

$= 0.036$

$P(x_t=(2,3) \mid x_{t-1}=(3,4) , o_t=(-1,0))\ P(x_{t-1}= (3,4))$

$= 0.01$

Markov Localization
Grid Based Example

- Example Solution:
  - *So the probability of being at position (2,3) given the odometry is the total probability of moving there from each possible position:*

$$P(x_{it}=(2,3)|\ o_t=(-1,0)) \quad = \sum P(\ x_t=(2,3)|\ x_{j,t-1}\ ,\ o_t=(-1,0))\ P(x_{j,t-1}\ )$$

$$= 0.09 + 0.005 + 0.036 + 0.01$$

$$= 0.141$$

Markov Localization
Applying Probability Theory

2. CORRECTION Step: refine the belief state

$$P(x_{i,t} \mid z_t) = \frac{P(z_t \mid x_{i,t}') \, P(x_{i,t}')}{P(z_t)}$$

- $P(x'_{i,t})$: the belief state before the perceptual update i.e. $P(x_{i,t} \mid o_t)$
- $P(z_t \mid x_{i,t}')$: the probability of getting measurement z_t from state $x_{i,t}'$
- $P(z_t)$: the probability of a sensor measurement z_t. Calculated so that the sum over all states $x_{i,t}$ from equals 1.

Markov Localization

- Critical challenge is calculation of $P(z \mid x)$
  - The number of possible sensor readings and geometric contexts is extremely large
  - $P(z \mid x)$ is computed using a model of the robot's sensor behavior, its position x, and the local environment metric map around x.
  - Assumptions
    - Measurement error can be described by a distribution with a mean
    - Non-zero chance for any measurement
    - Sensor is located at center of robot

Markov Localization
Grid Based Example

- Example Problem:
  - *What is the probability of being in state* $x = (2,3)$ *given we have range measurement* $z = 0.8m$ *?*

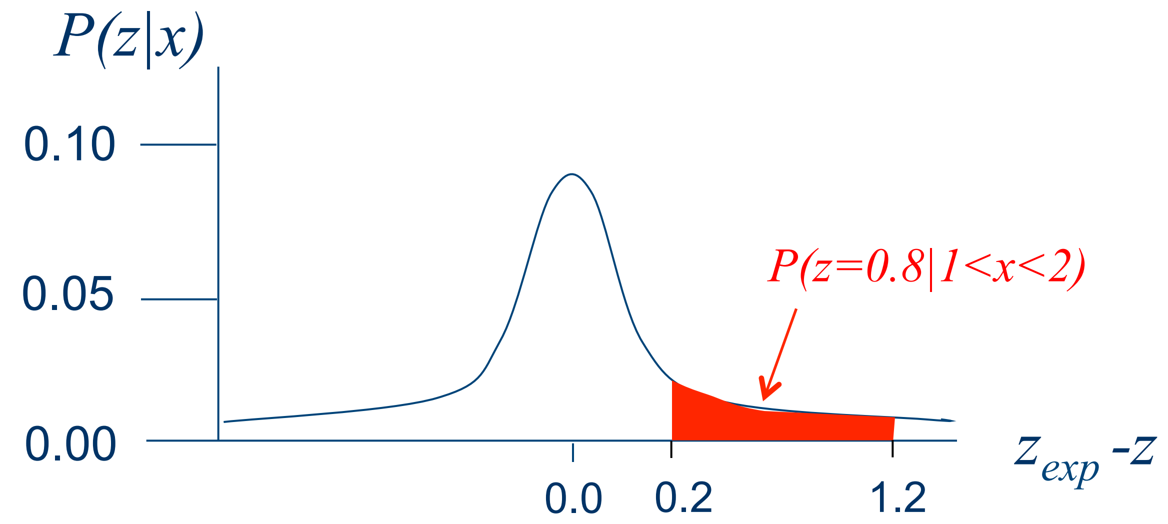
$$P(x_t = (2,3) \mid z_t = 0.8) = \frac{P(z_t = 0.8 \mid x_t' = (2,3)) \; P(x_t' = (2,3))}{P(z_t = 0.8)}$$

Markov Localization
Grid Based Example

- Example Solution:
  - *We can use the probability* $P(x_t'=(2,3)) = 0.141$ *from the previous example.*
  - *The interesting term is* $P(z_t=0.8 \mid x_t'=(2,3))$*.*
    - *Using the map, we can calculate the expected value of the range sensor measurement.*
    - *If the robot is at* $(2,3)$ *and facing to the left, it should get a range measurement between* $1m$ *and* $2m$*.*
    - *Recall that we can use the probability density function representing the sensor characteristics, and that the expected value is between 1 and 2.*

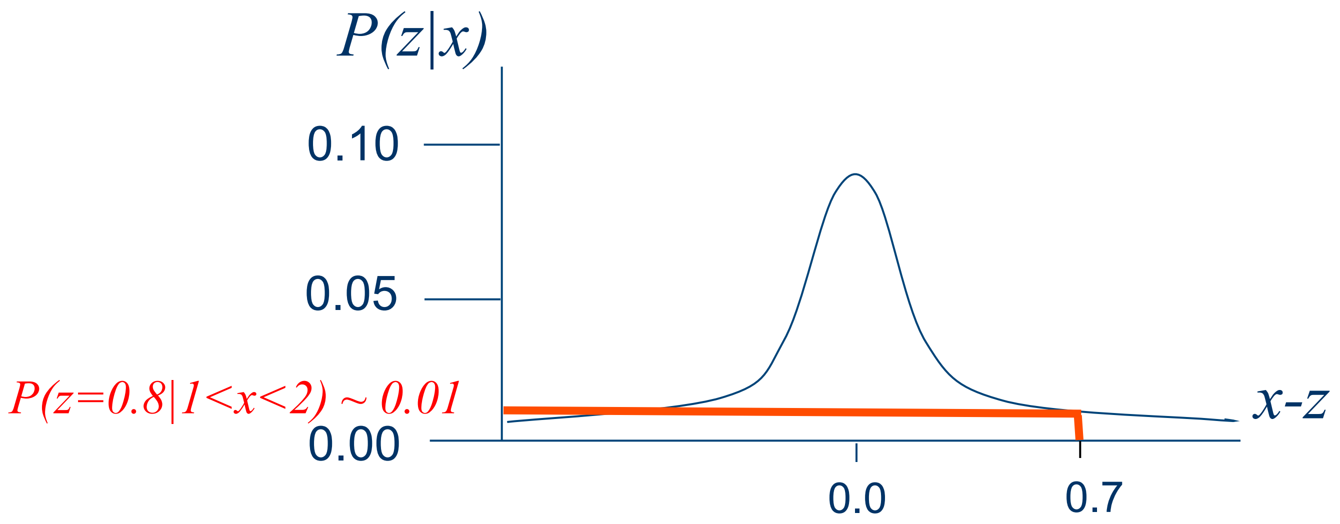
Markov Localization
Grid Based Example

- Example Solution:
  - For Ultrasound, $P(z|x)$ can be taken from the following distribution:



Markov Localization Grid Based Example

- Example Solution:
  - Often, we approximate



Markov Localization
Grid Based Example

- Example Solution:
  - *Now we can calculate the numerator for*

$$p\,(x_t = (2,3)) \mid z_t = 0.8)$$

$$= \frac{p\,(z_t = 0.8 \mid x_t' = (2,3))\;\; p\,(x_t' = (2,3))}{p\,(z_t = 0.8)}$$

$$= \frac{(0.01)\;(0.141)}{p\,(z_t = 0.8)}$$

Markov Localization
Grid Based Example

- Example Solution:
  - *Finally, we can calculate the denominator by ensuring the sum of all probabilities is 1.*

$$1 = \sum_{i=1}^{n} P(x_{i,t} \mid z_t = 0.8)$$

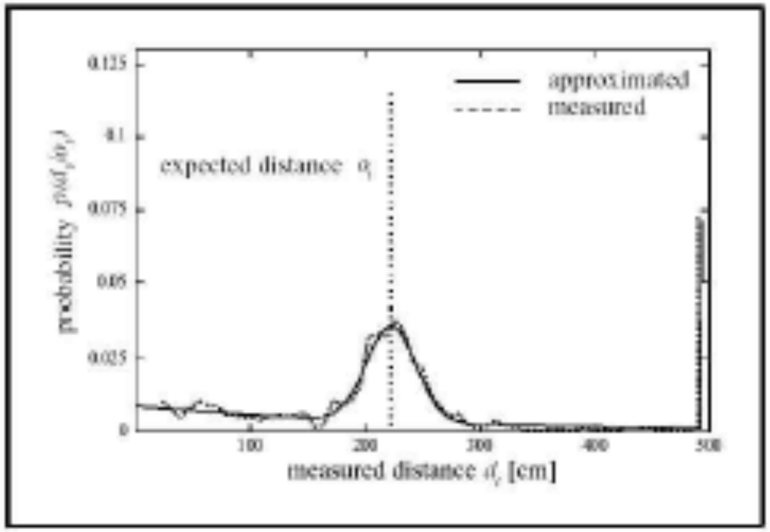
$$= \frac{\sum P(z_t = 0.8 \mid x_{i,t}') \, P(x_{i,t}')}{P(z_t = 0.8)}$$

Therefore:

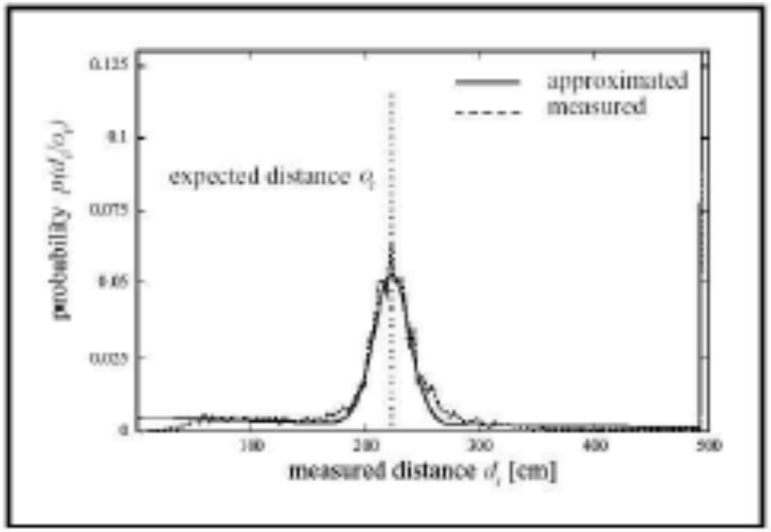
$$P(z_t = 0.8) = \sum P(z_t = 0.8 \mid x_{i,t}') \; P(x_{i,t}')$$

Markov Localization
Grid Based Example

- Here are some typical sensor distributions:



Ultrasound.



Laser range-finder.

Outline - Localization

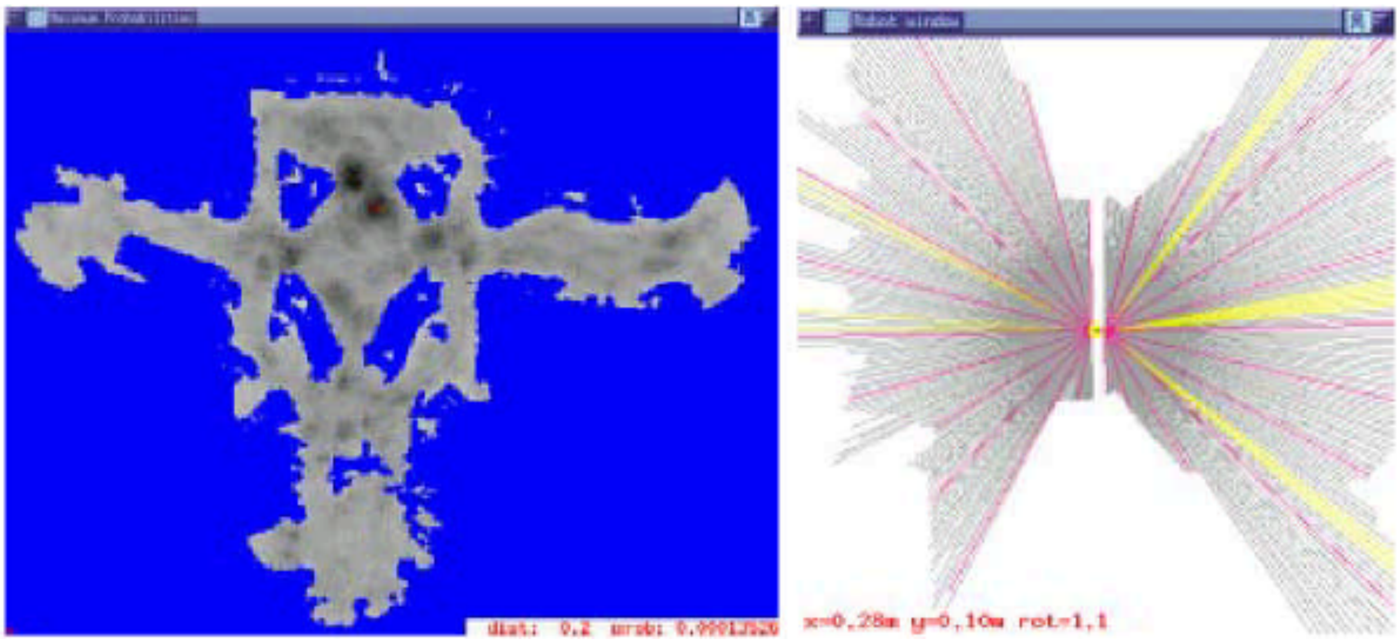
1. Localization Tools
2. Overview of Algorithms
3. Markov Localization
   - Overview
   - Prediction Step
   - Correction Step
   - ML Example

ML Example 1

- Smithsonian Navigation
  - Time steps taken from ML example of the robot Minerva navigating around the Smithsonian.
  - In the following figures:
    - Left side shows belief state. Darker means higher probability.
    - Right side shows actual robot position and sensor measurements.

ML Example 1

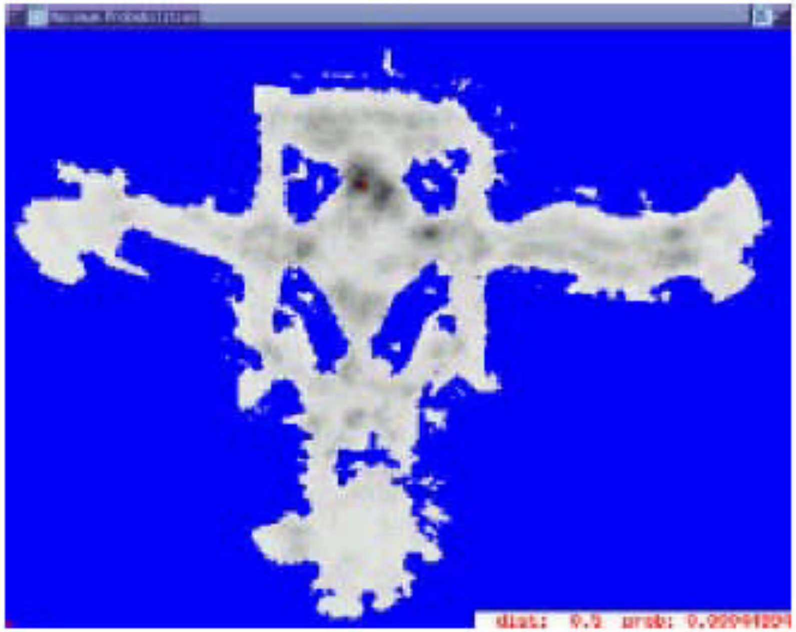
- Laser Scan 1 of Museum

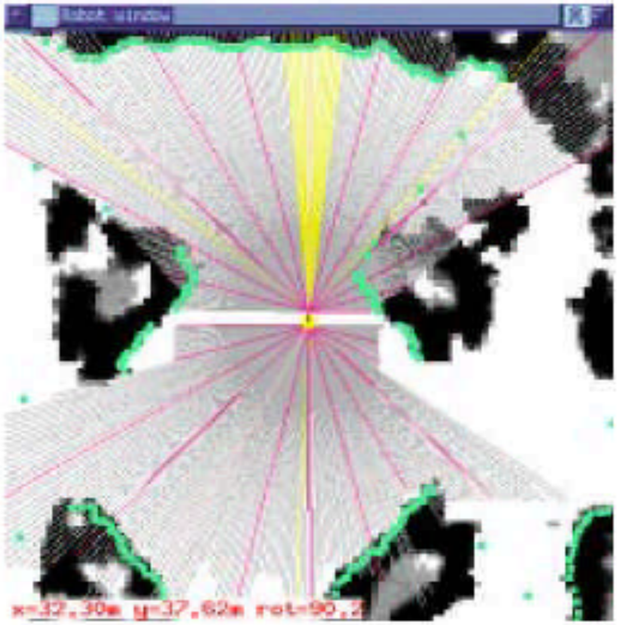


Figures courtesy of W. Burgard

ML Example 1

- Laser Scan 2 of Museum

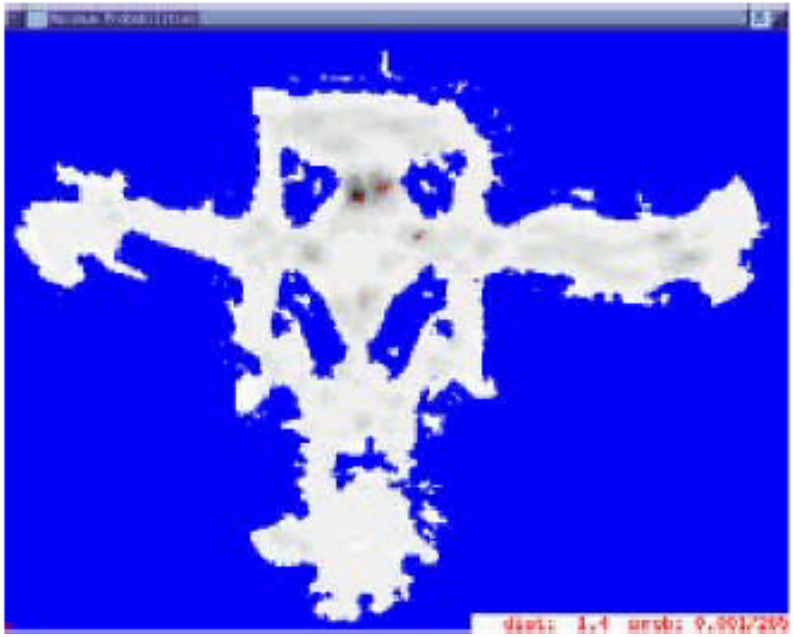


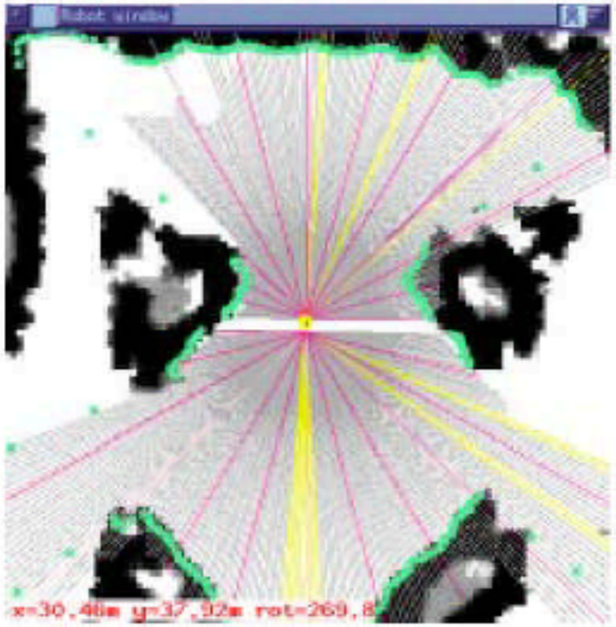


Figures courtesy of W. Burgard

ML Example 1

- Laser Scan 3 of Museum

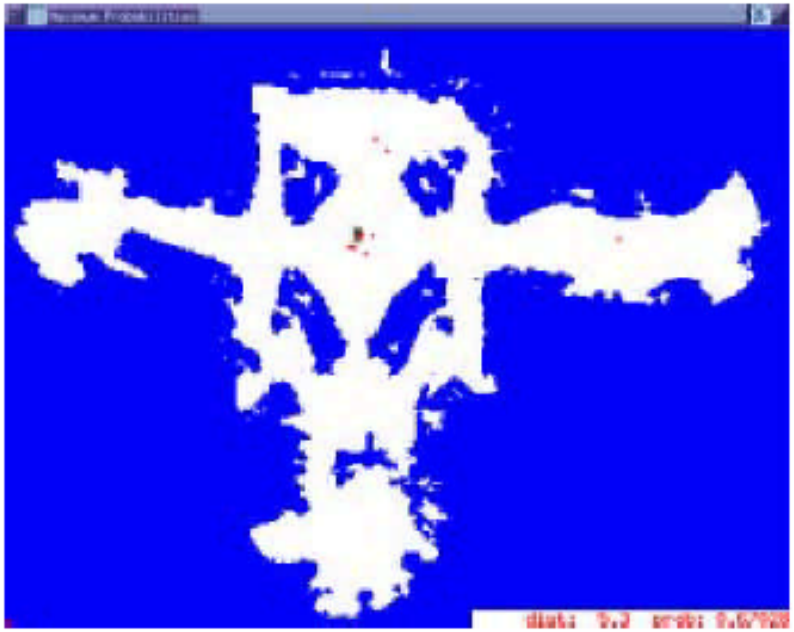


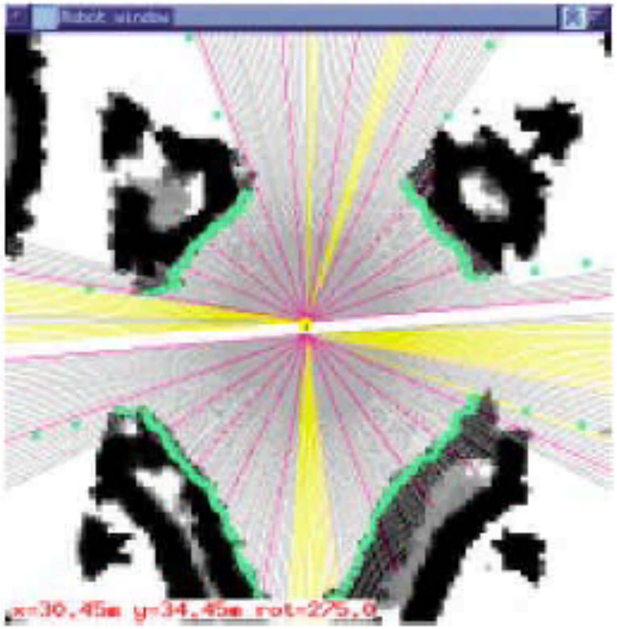


Figures courtesy of W. Burgard

ML Example 1

- Laser Scan 13 of Museum

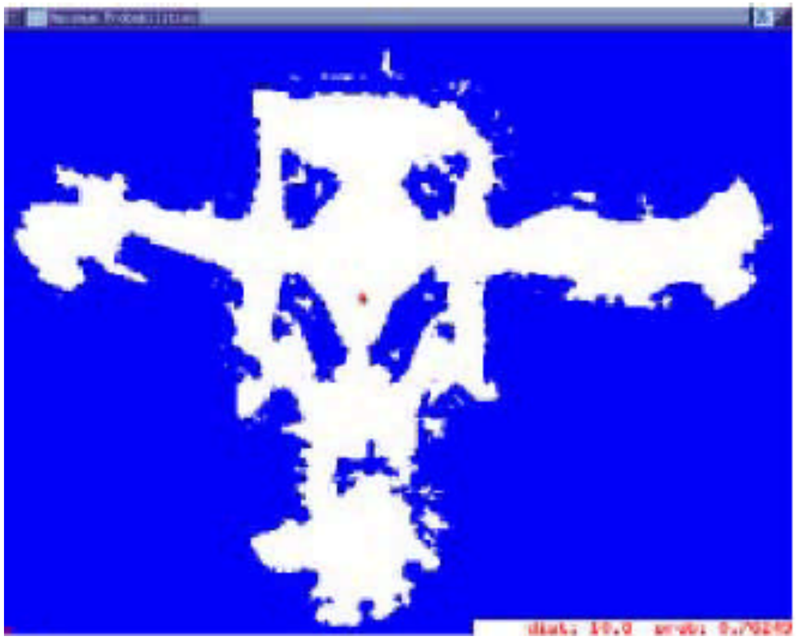


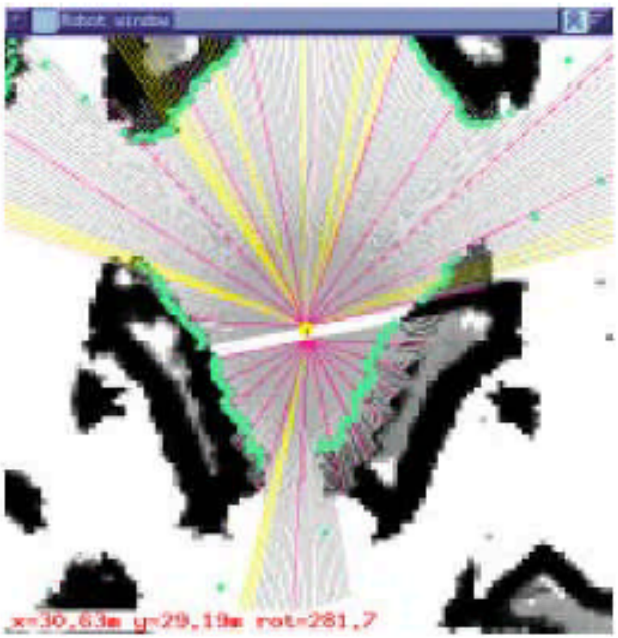


Figures courtesy of W. Burgard

ML Example 1

- Laser Scan 21 of Museum





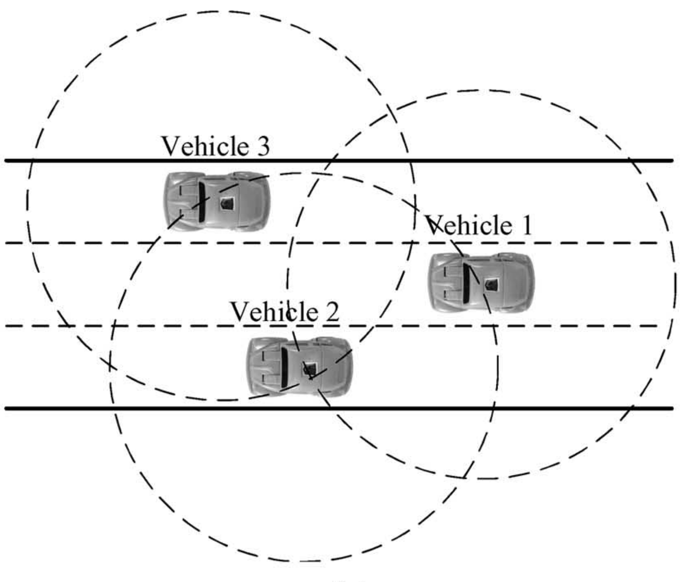
Figures courtesy of W. Burgard

ML Example 2

- Lane State Estimation
  - (Semi) Autonomous Highway Systems will benefit from lane position optimization
  - Vehicles must need to know what lane they are in.

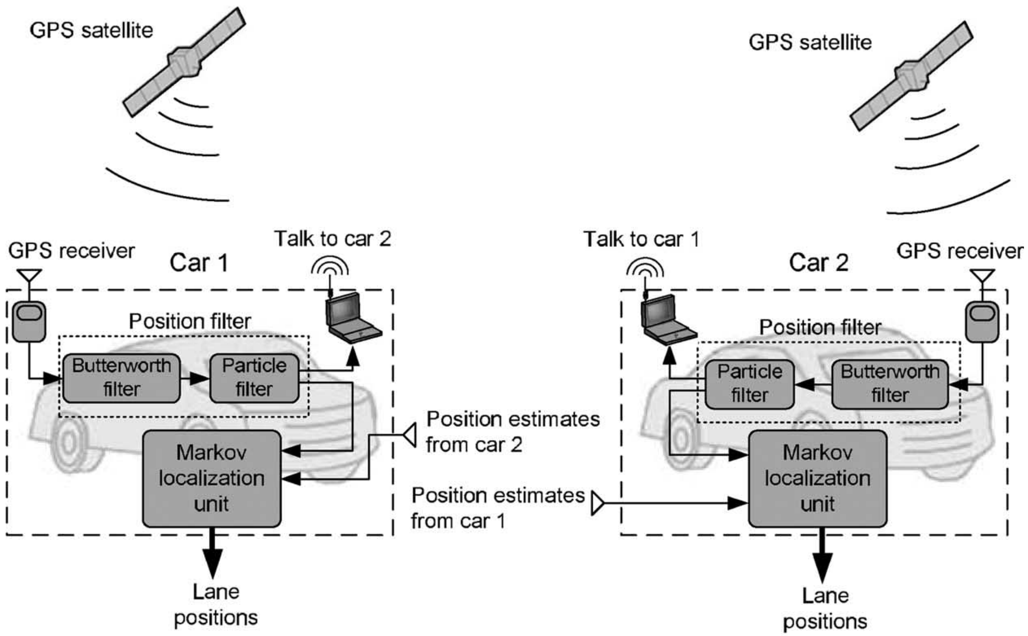
ML Example 2

- Multiple vehicles driving down a highway.
  - Can we estimate what lane they are in?



ML Example 2

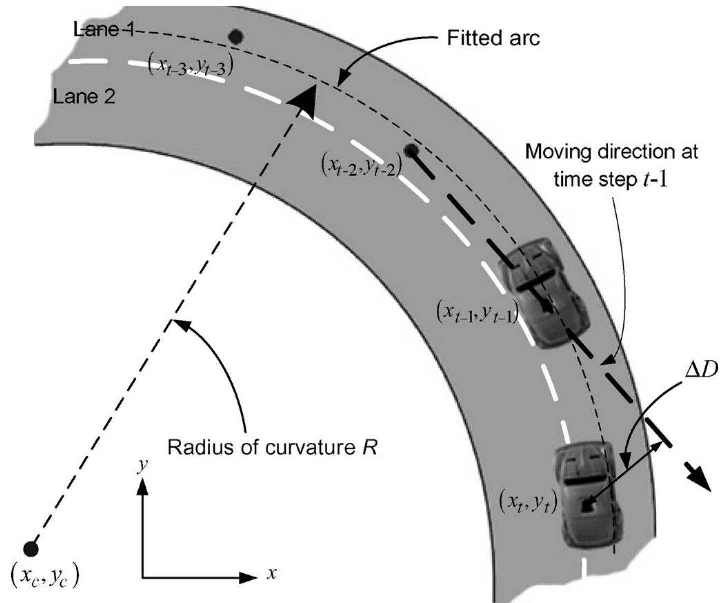
- Assume vehicles have
  - Inter-Vehicle Communication (IVC)
  - GPS



ML Example 2

- Baye's Filter - Prediction Step

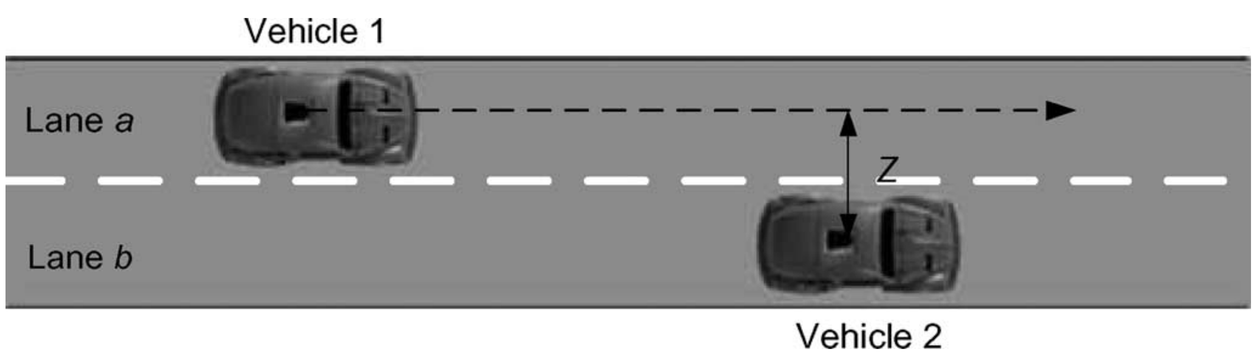
$$P(v_{i,t}=l_a) \leftarrow \sum_{j=1}^{2} P(v_{i,t}=l_a|v_{i,t-1}=l_j)P(v_{i,t-1}=l_j)$$



ML Example 2

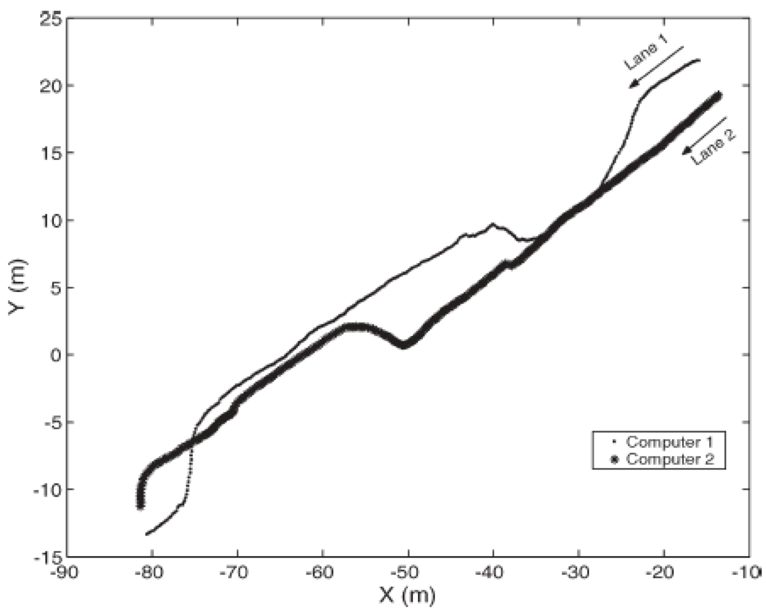
- Baye's Filter - Correction Step

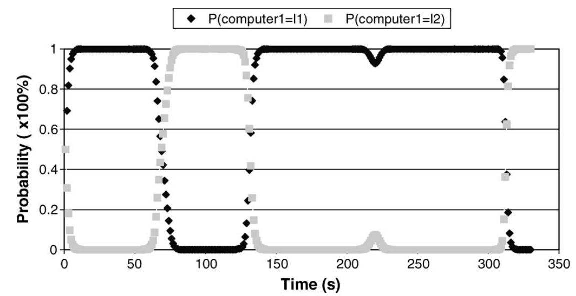
$$P(v_{1,t} = l_a, v_{2,t} = l_b | z_t) \leftarrow \frac{P(z_t | v_{1,t} = l_a, v_{2,t} = l_b) P(v_{1,t} = l_a, v_{2,t} = l_b)}{P(z_t)}$$



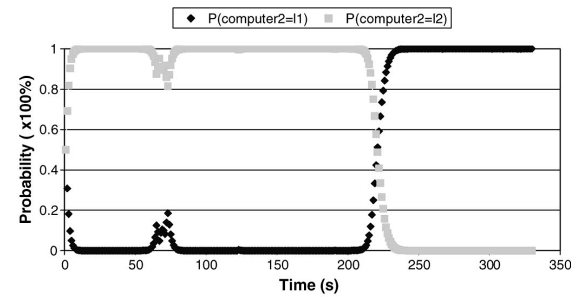
ML Example 2

- Results





(a)



(b)

ML Example 2

- Results