CS 4758/6758: Robot Learning

Spring 2010: Lecture 6

Ashutosh Saxena

Previous Lecture

Probability distributions for modeling the sensor data.

Maximum Likelihood approach for estimation.

(First homework problem in HW2 explores this.)

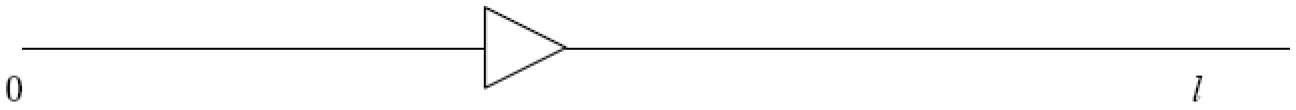
Robot Localization

- Data from sensors is affected by measurement errors.

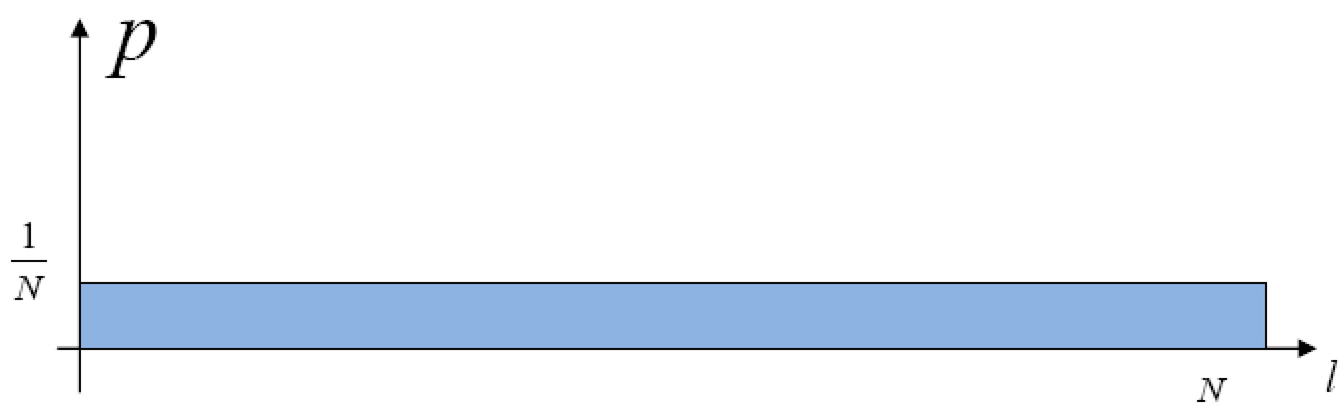
- We can only compute the probability that the robot is in a given configuration.

Localization

- Robot is placed along a line, but it does not know where.

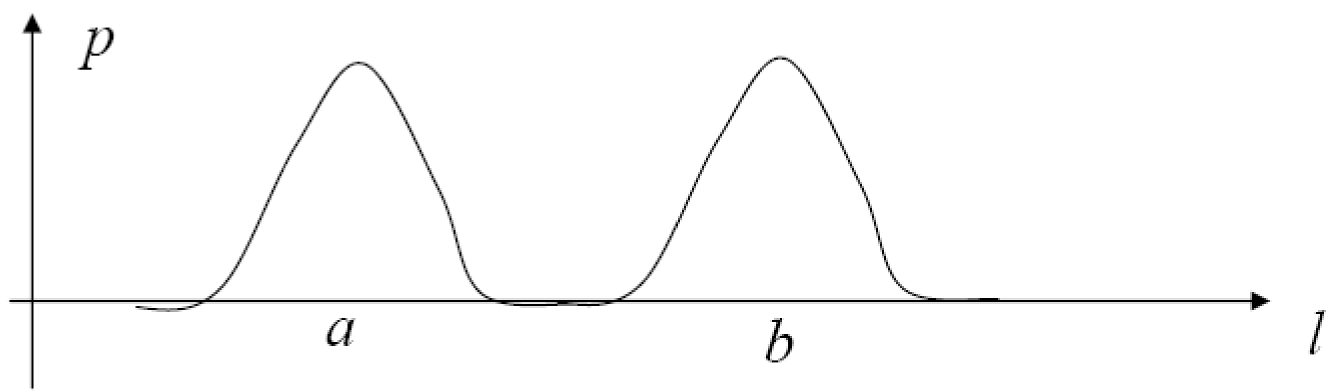


- What is p(l) ?



Multimodal Distribution

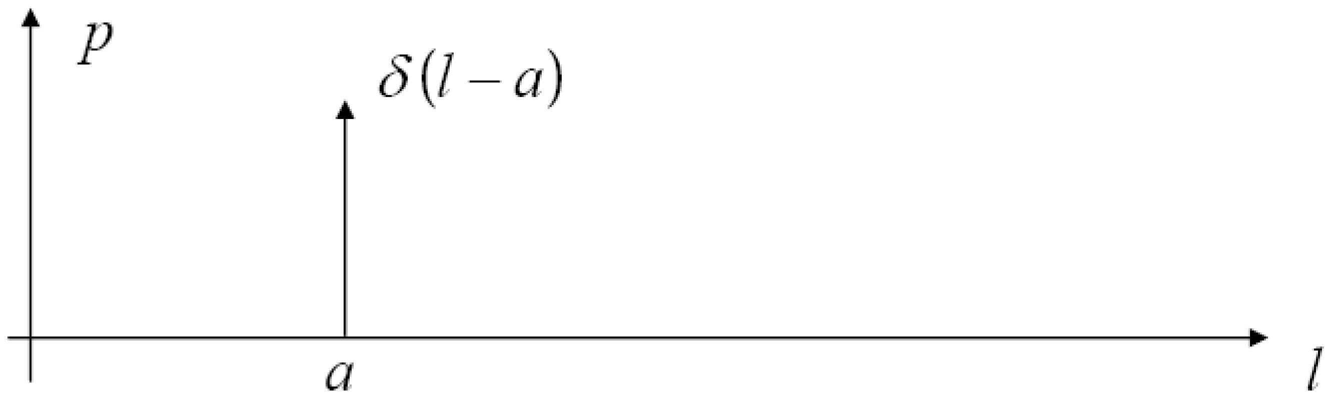
Robot near 'a' or 'b'



Which distribution to use to model this?

Dirac Distribution

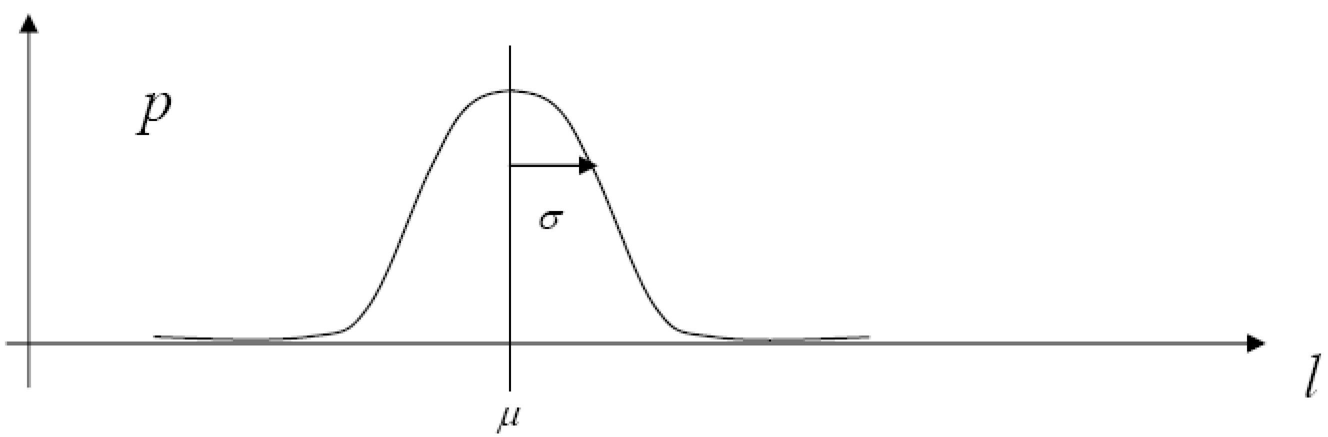
Robot is at ‘a’ with probability 1.0



$$\delta(l) = \begin{cases} \infty & l = 0 \\ 0 & otherwise \end{cases} \quad \text{with} \int_{-\infty}^{+\infty} \delta(l)\,dl = 1$$

Gaussian Distribution

- Used in Kalman filters.

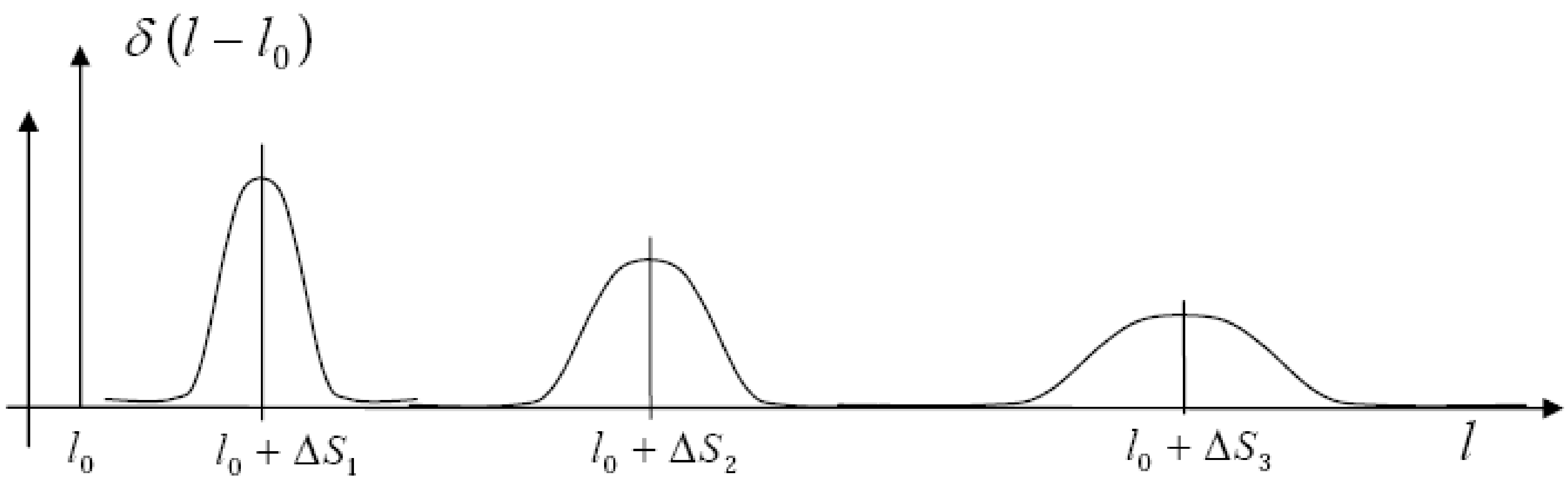


Probabilistic approach for Robot Localization

- Initial probability distribution p(l)
- Statistical model characterizing the error of each sensor p(s|l)
- Data from the sensor $s=s_i$
- Map of the environment.

Two phases in Robot Operation

- **Action**
  - Take a known action
  - E.g., move the robot by ΔS
  - Often movement is not perfect.

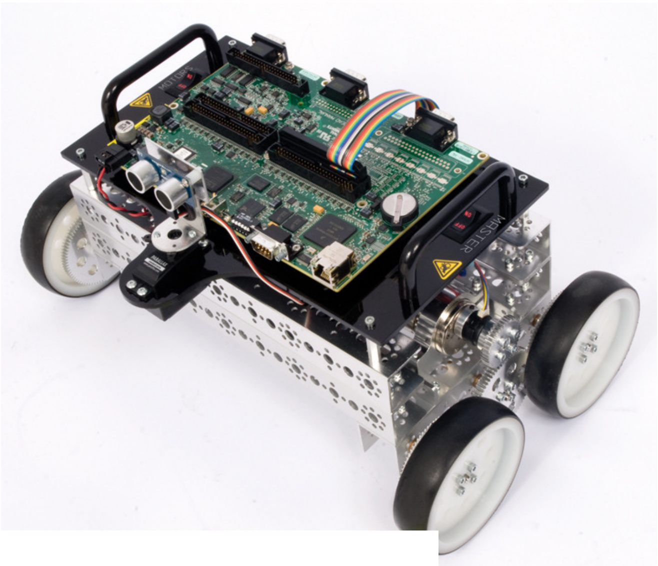


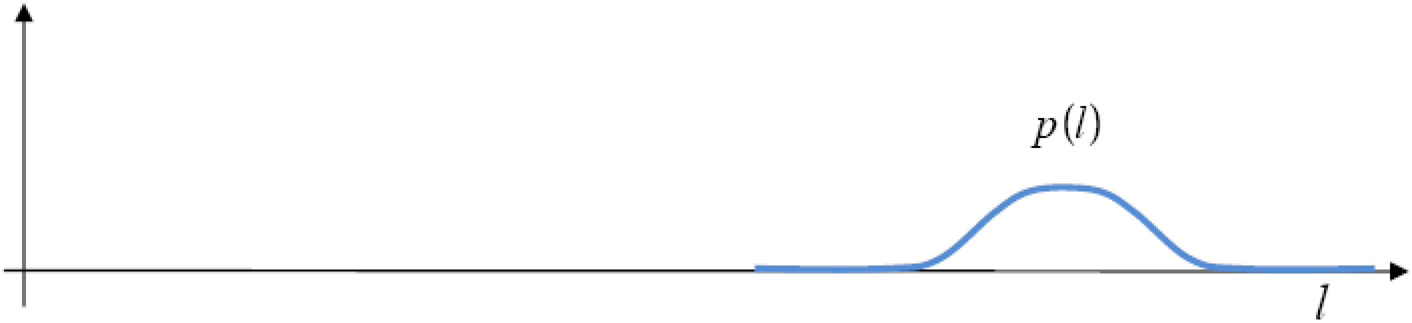
Two phases in Robot Operation

- **Perception**
  - Use sensors to figure out where it is.
  - Sensors are not perfect.
  - Use probability distributions.

Example

- Initial estimate p(l)

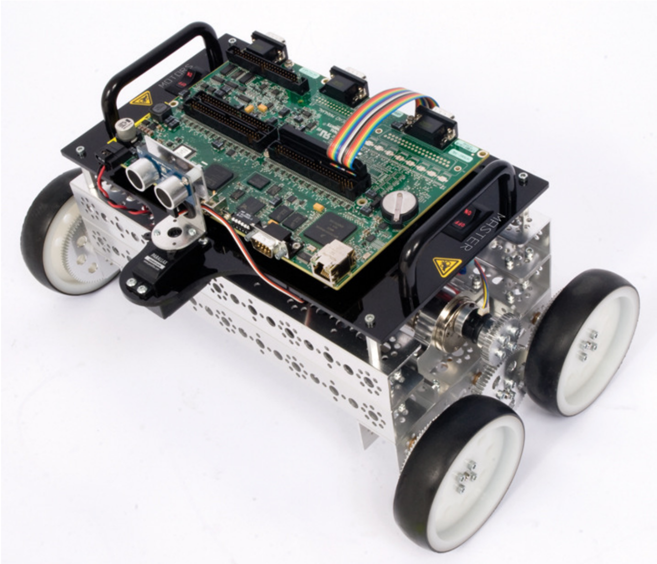


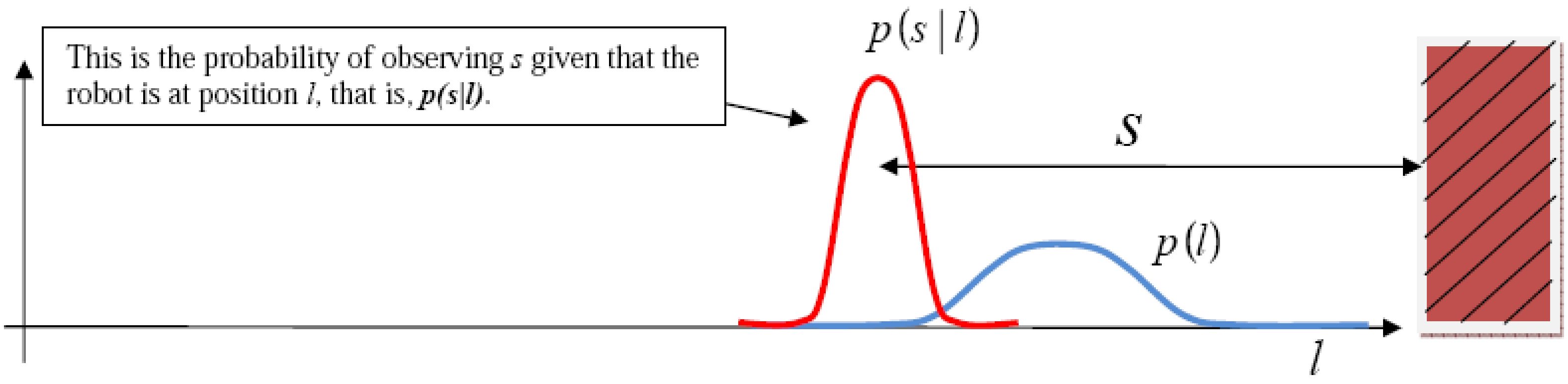


Example

- Robot uses sonar.
- Sonar statistical error:

  p(s | l)

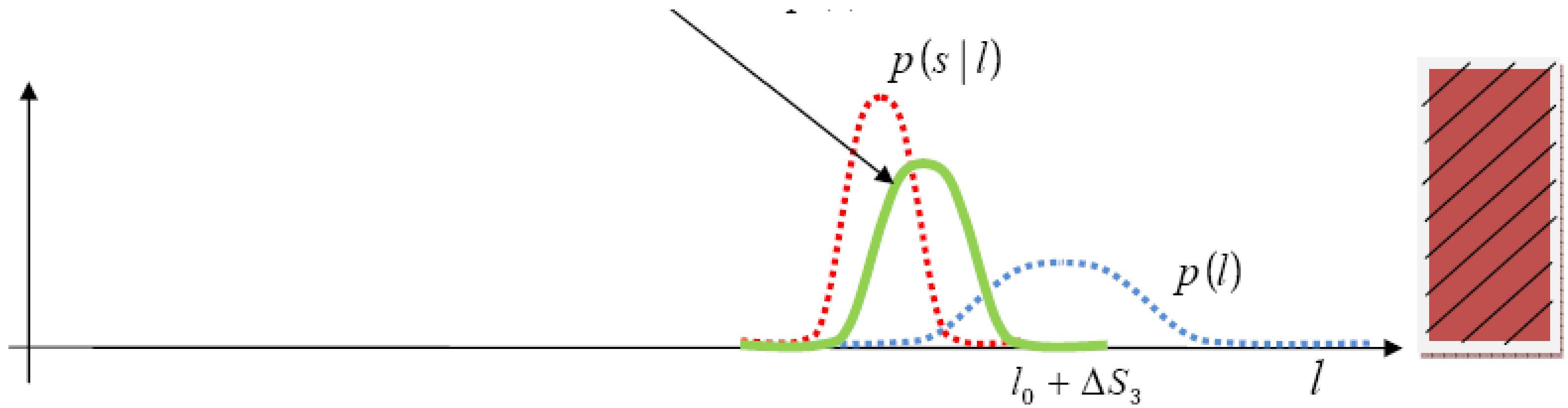




What is p(l | s) ?

Bayes Rule

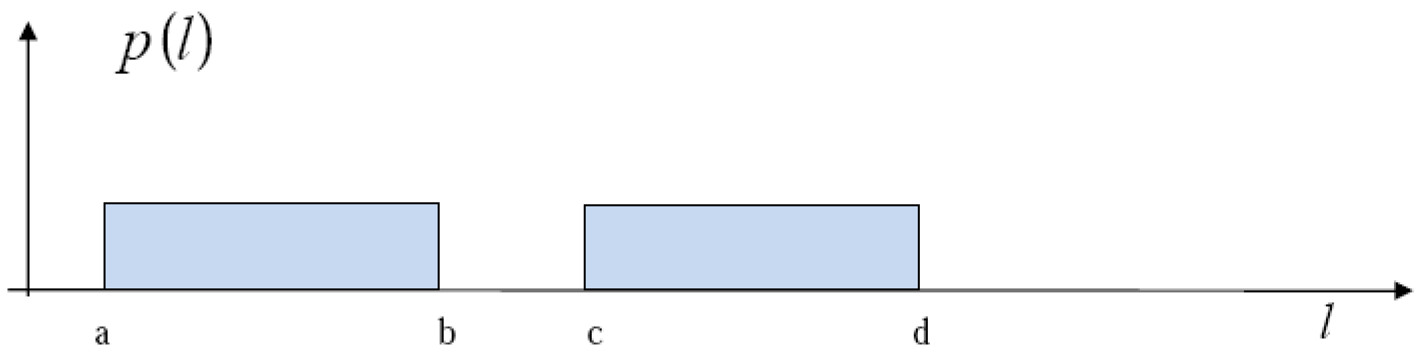
$$P(l \mid s) = \frac{p(s \mid l)\,p(l)}{p(s)}$$



Example 2

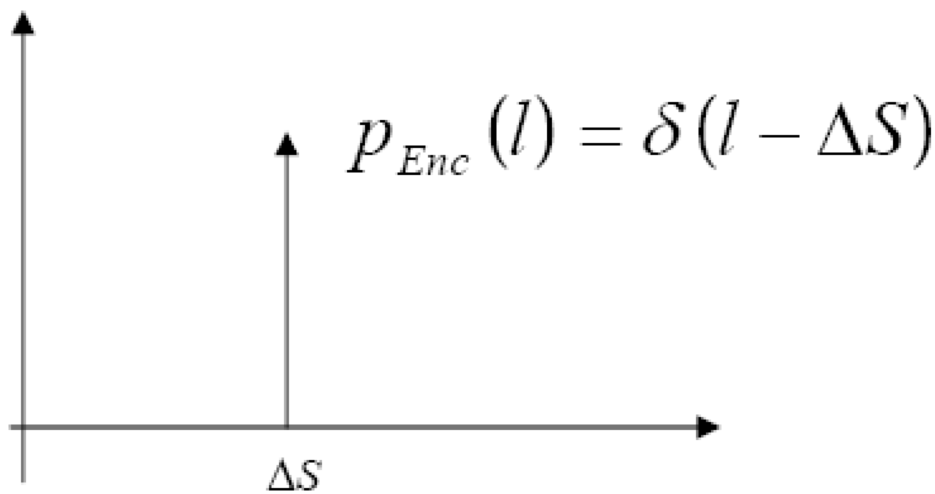
- Prob. Distribution of robot position = *belief*

Initial belief:



Example 2

- The robot moves ΔS, and assume the action is "perfect".



- What is the new robot belief?

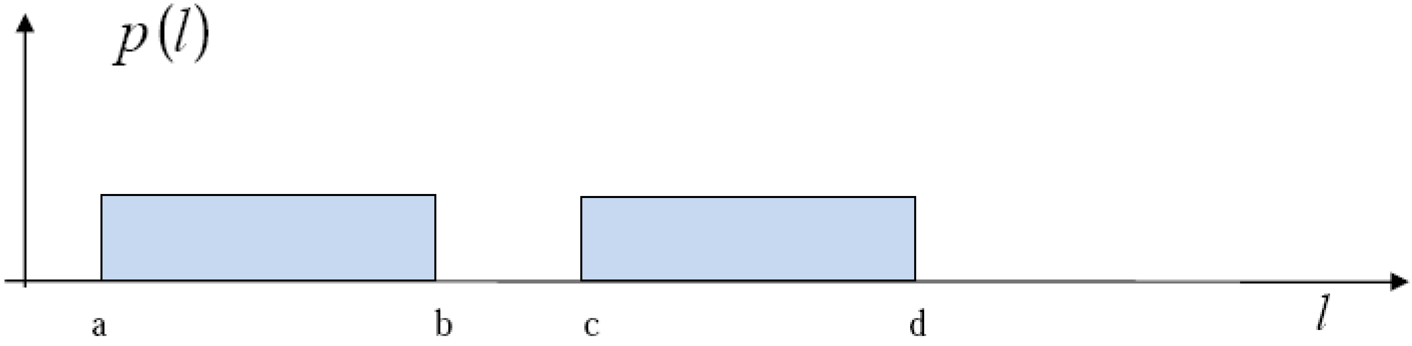
Involves a convolution operation

$$P_{new}(l) = p(l) * p_{enc}(l)$$

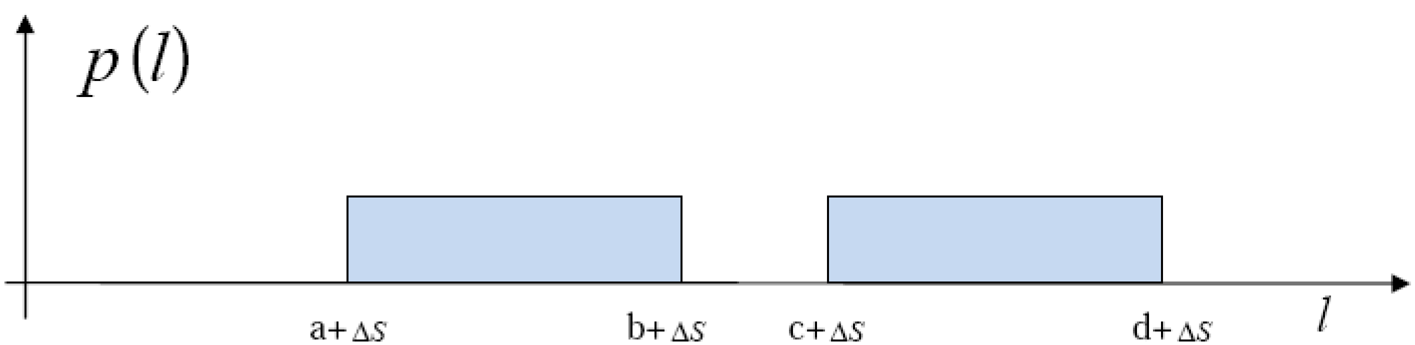
$$= \sum_{l'} p(l')\, p_{enc}(l-l')$$

New Robot Belief

- Before:

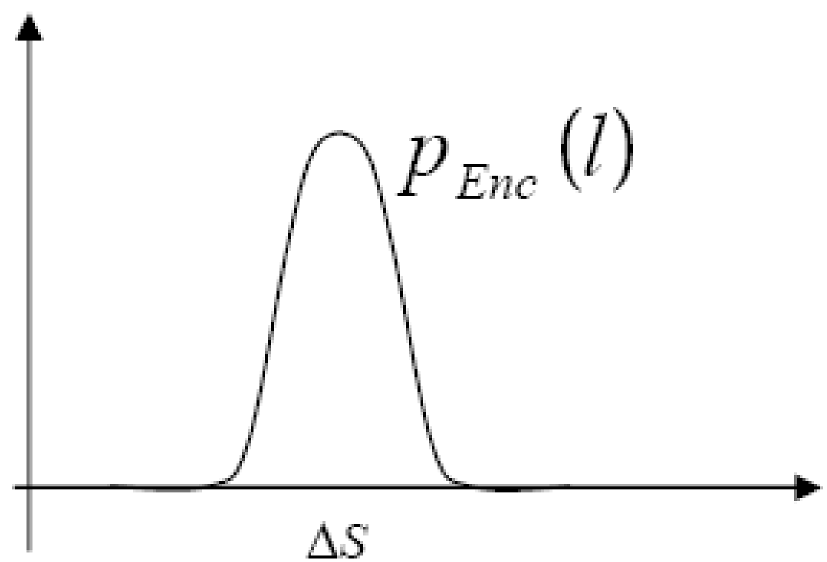


- After taking the action:



Error in Action

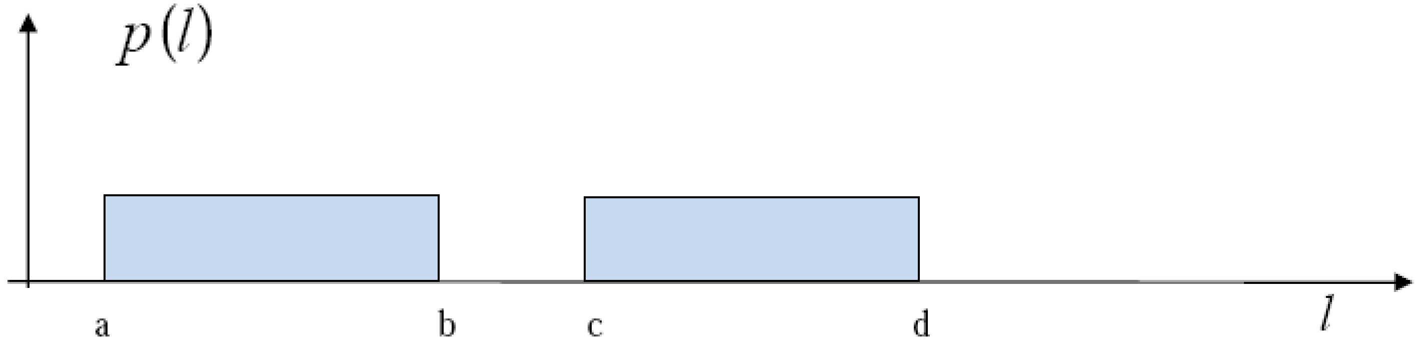
- Suppose the action is not perfect, but rather approximately moves the robot.



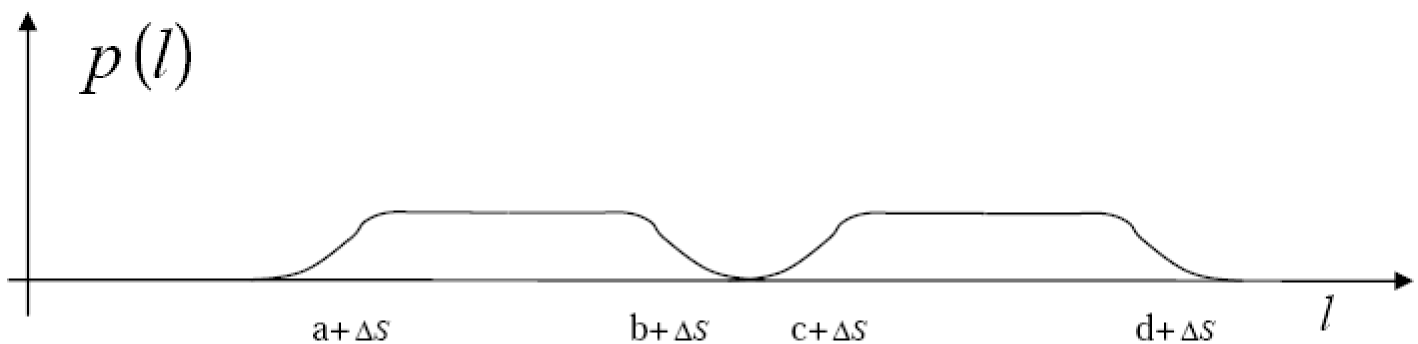
- What is the new robot belief?

New Robot Belief

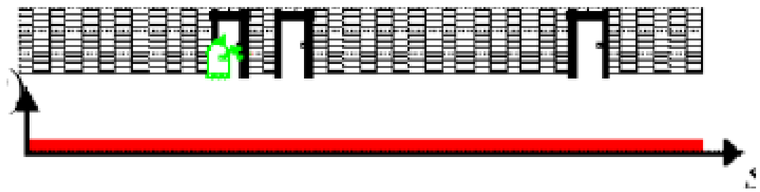
- Before



- After taking the action:



Robot movement



Initially, the robot is put somewhere, but not told its location. P(l) is uniform

Perception, robot queries its sensor, but does not know which one.

$p(l|s_1) = p(s_1|l)\ p(l)\ /\ p(s_1)$

Action, robot moves one meter forward. Error in motion makes p(l|s) smoother.

$p(l) = p(l) * p_{enc}(l)$

Perception, robot queries again

$p(l|s_2) = p(s_2|l)\ p(l)\ /\ p(s_2)$

Where is the robot?

$l^* = \arg\max_l \; p(\, l \mid s_1, s_2)$

Maximum likelihood estimate.

Can also do:

$l^* = \arg\max_l \; \log p(\, l \mid s_1, s_2)$

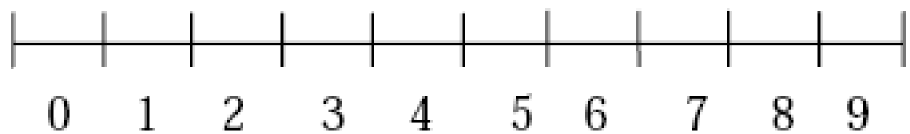
Summary

- Decide state space, i.e., how to represent 'l'.
- Choose a statistical model for sensors $P(s \mid l)$
- Choose a statistical model for action $P_{enc}(l)$
  - Only needed when the robot is moving.
- Use *Bayes rule / conditional independence* properties to compute the probability $p(l \mid s_1, s_2, \ldots)$
  - Compute $\arg\max p(l \mid s_1, s_2, \ldots)$

Two approaches

Grid-based localization

- Discretize the state into many cells.
  - E.g., for 1D problem:



  - For 3D problem: need 100x100x100 = 1000000 cells.

Kalman filter based localization

- Only use Gaussian distribution to model robot motion and sensors.
- Benefit: Only need μ and ∑ (fewer numbers)
- Cons: ?

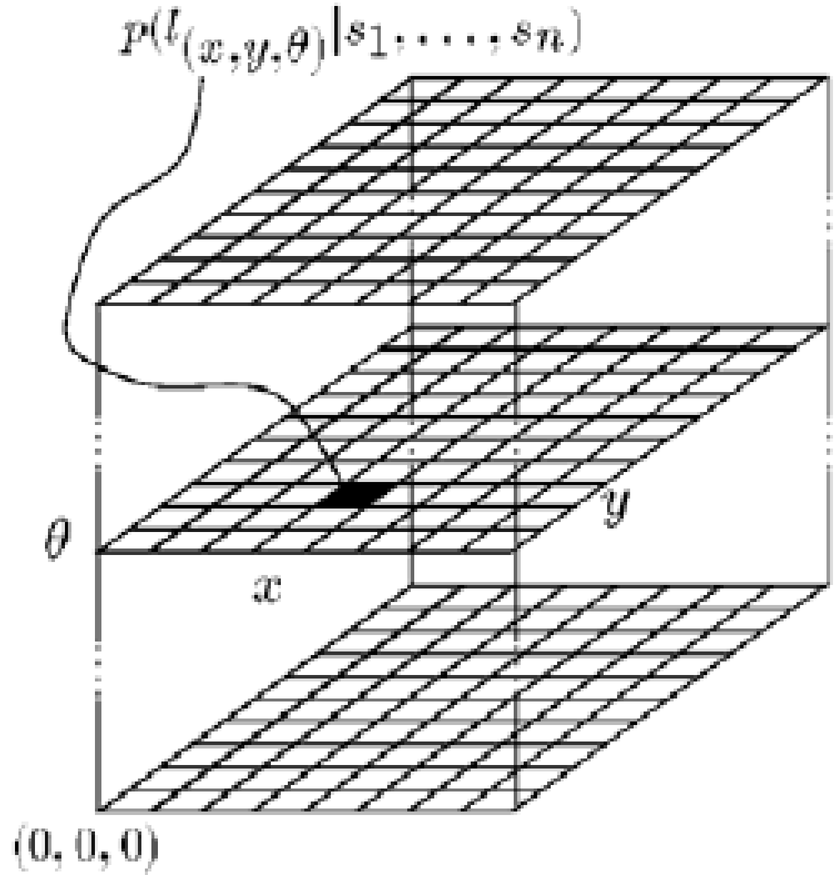
Grid-Based (Markov localization)

For a 2D robot (e.g., a car): (x,y,θ)

- We need 3D grid.

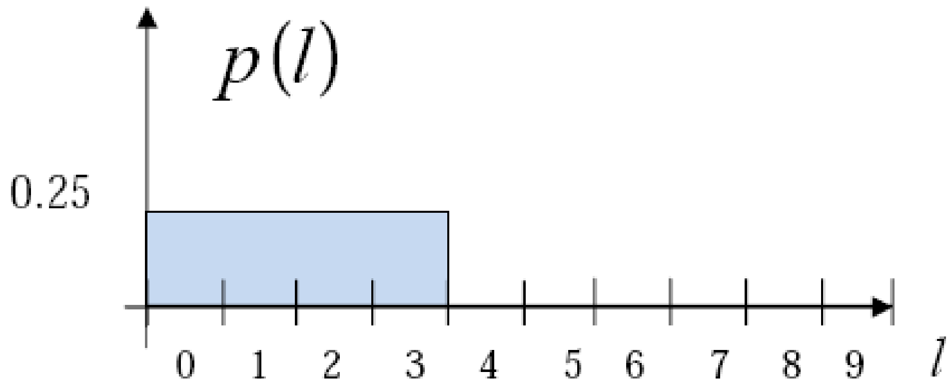
For a 3D robot (e.g., a helicopter)

- We need 6D grid



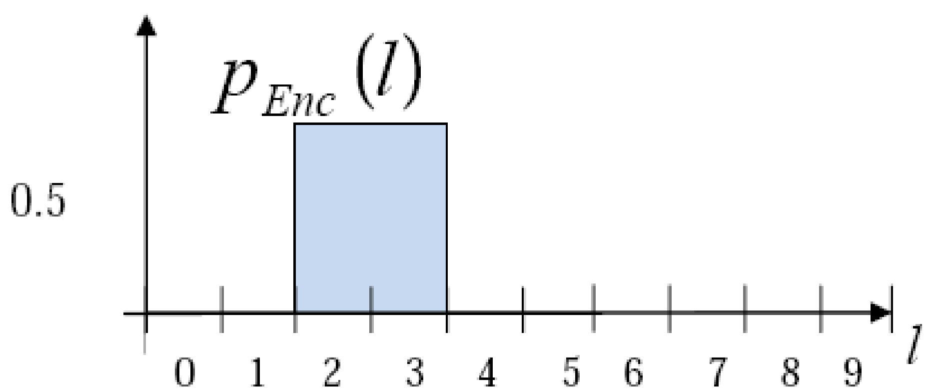
Example

Let us assume that the robot probability distribution at t=0 is the one below:



9.1 Action Phase:

Let us assume that the robot moves forward with the following statistical model:

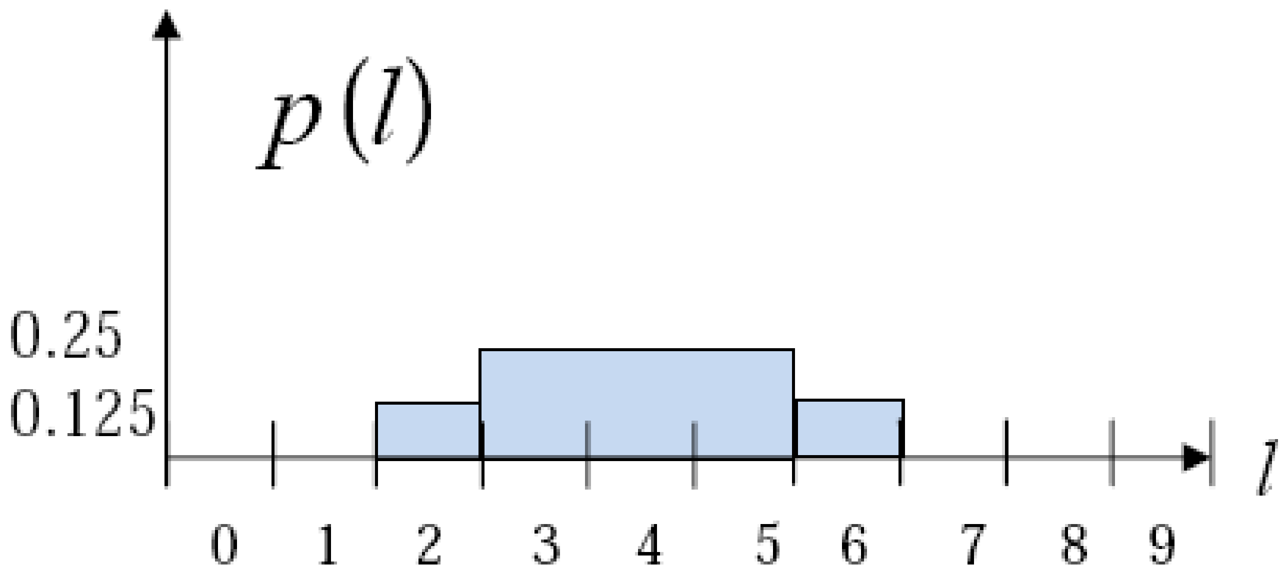


After Action phase

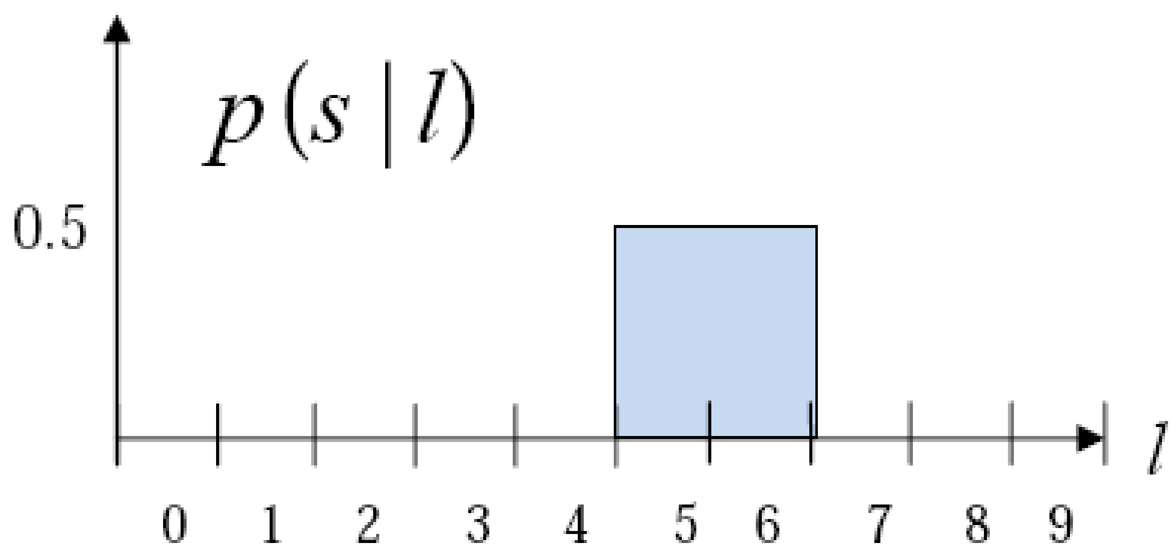
- How would p(l) look?

$$p(l) * p_{Enc}(l) = \sum_{m=0}^{m=9} p[m] \cdot p_{Enc}[l - m]$$

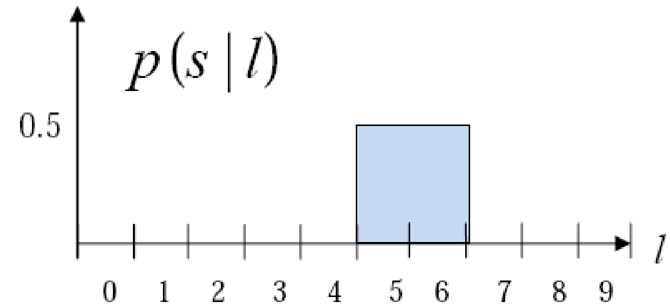
That is:



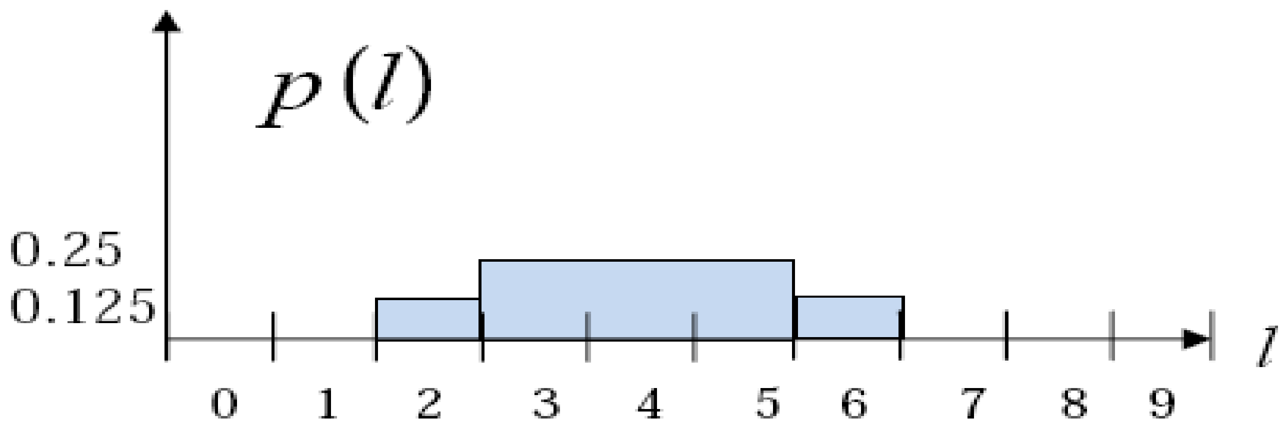
Perception phase



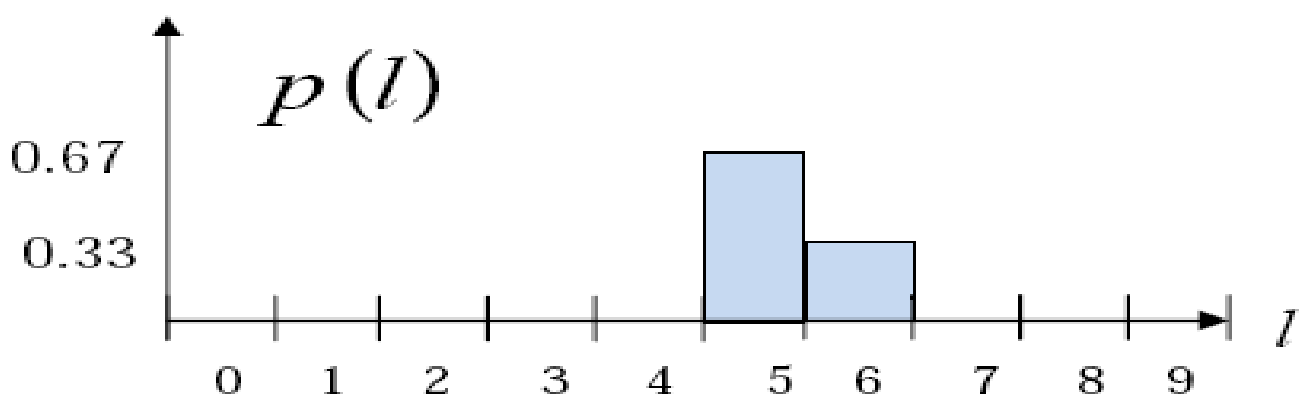
Robot's final belief



Before perception:

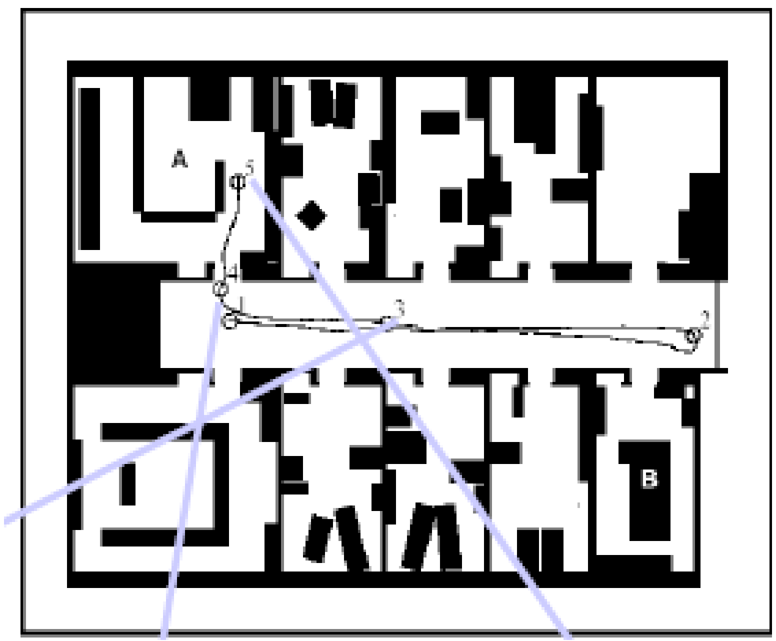


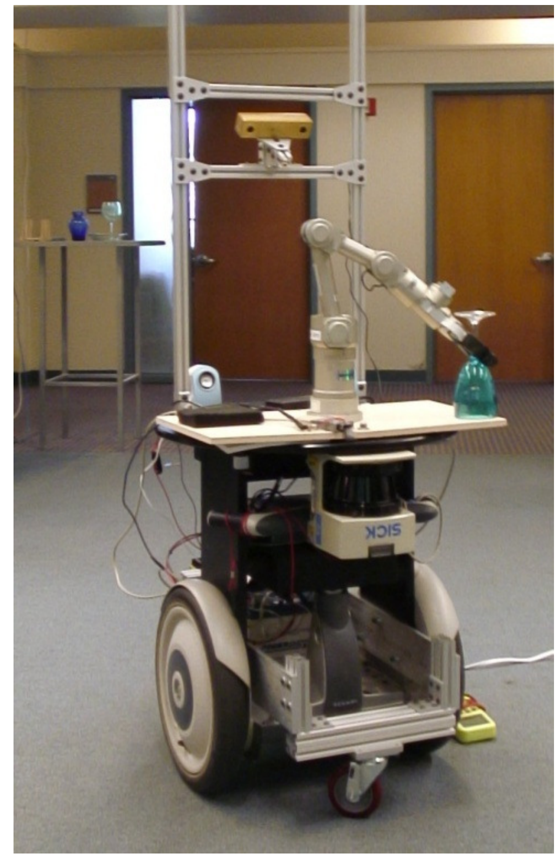
After perception



2D localization

- Sensors (laser)
- 2D grid



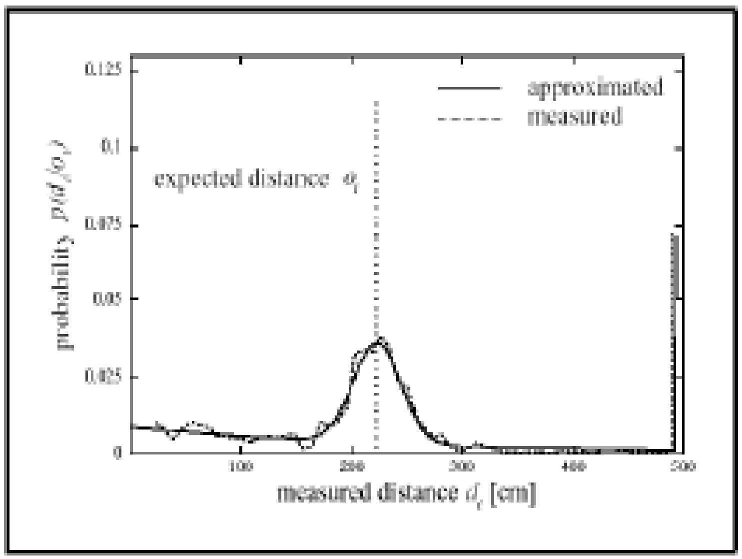


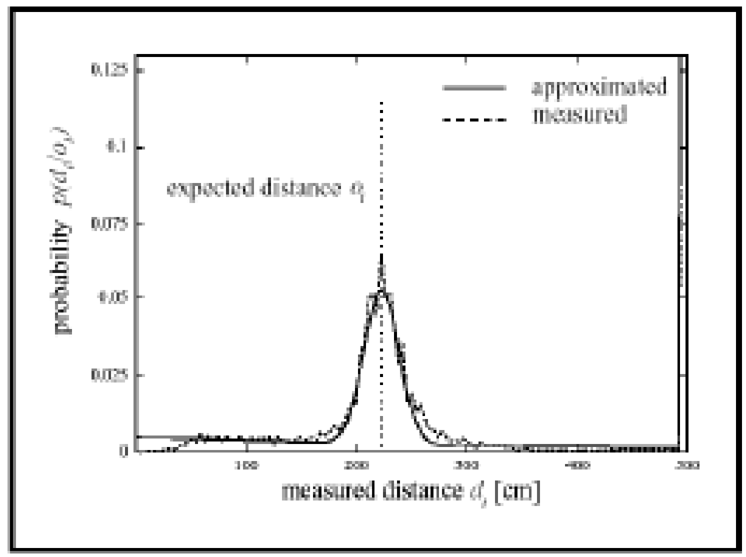
Sensor Modeling

How to model p(s|l) ?

Ultrasonic (left)

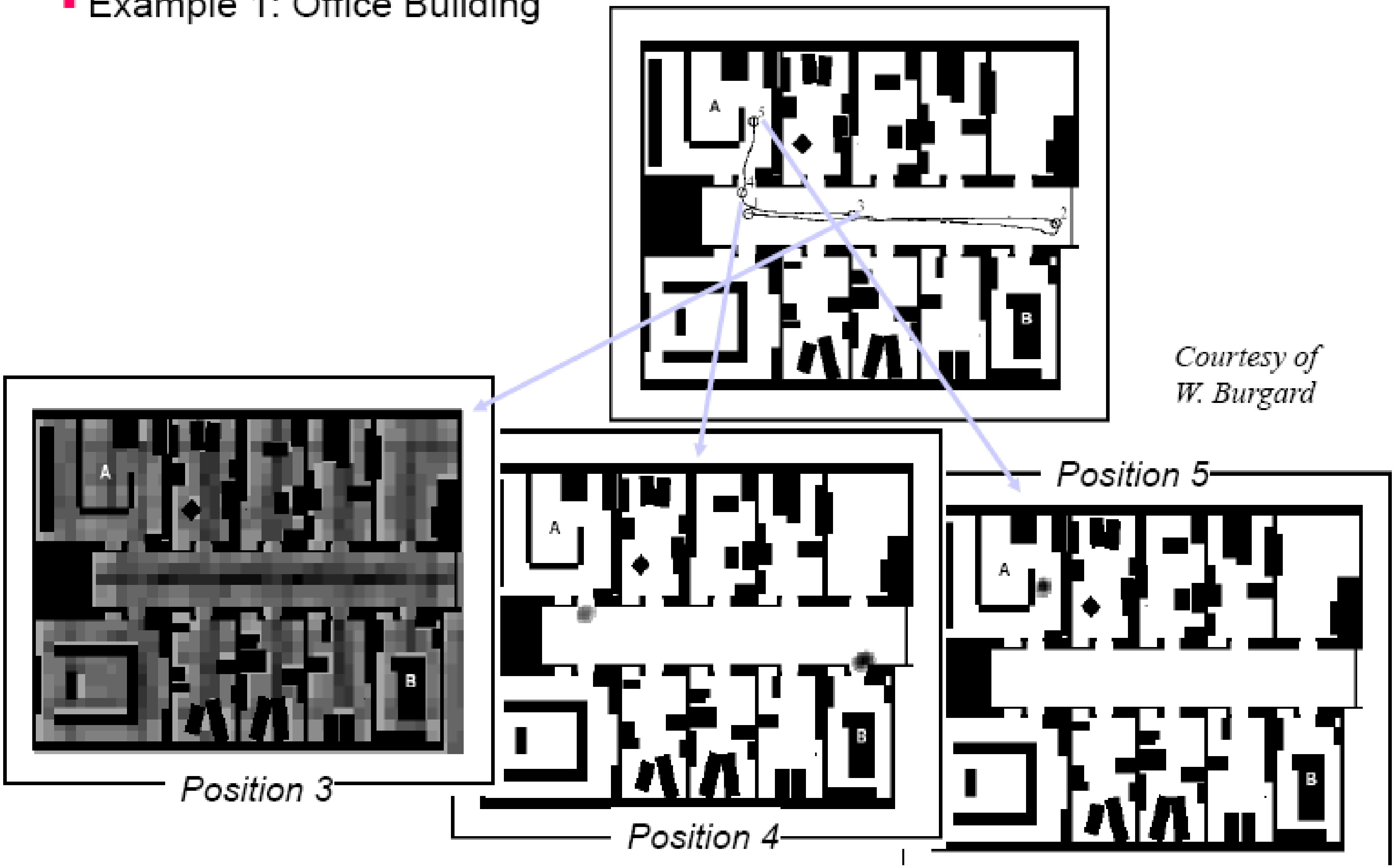
Laser (right)



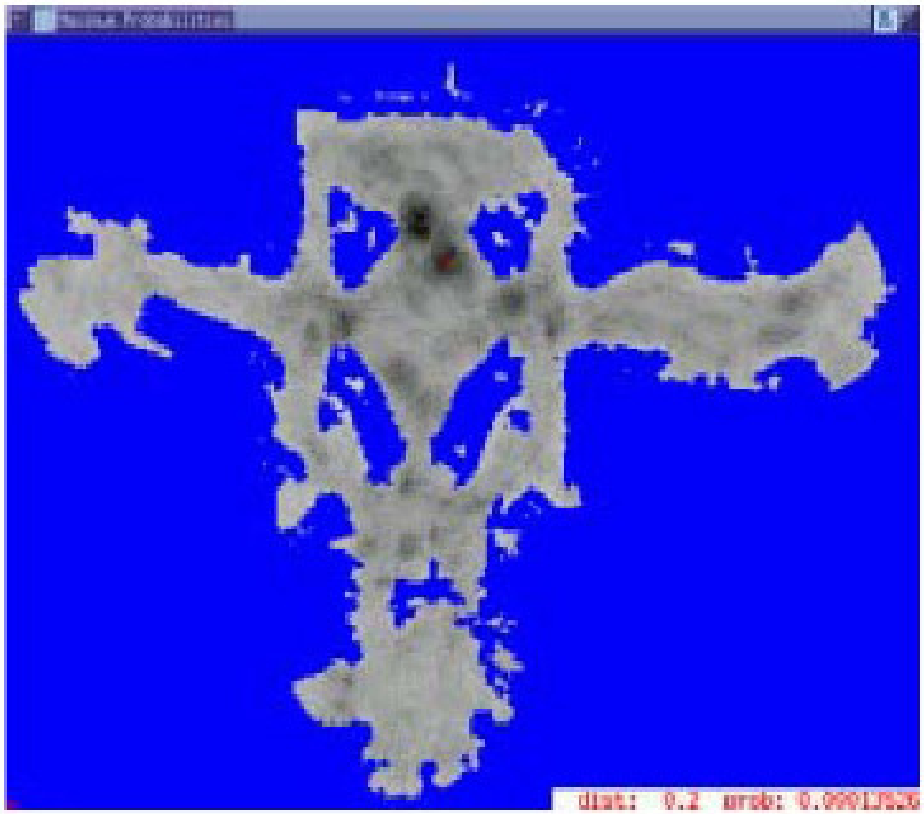


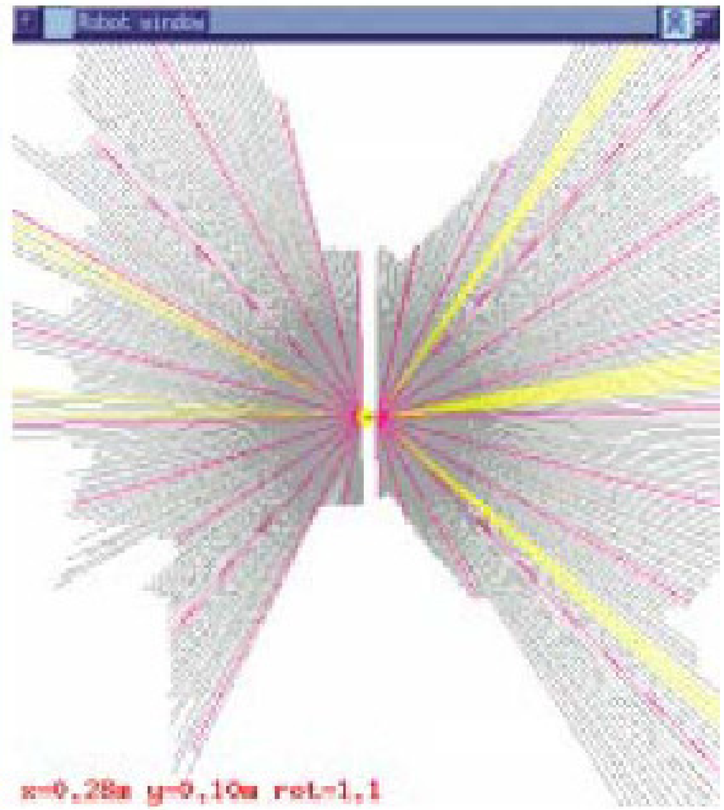
2D grid: beliefs at different times

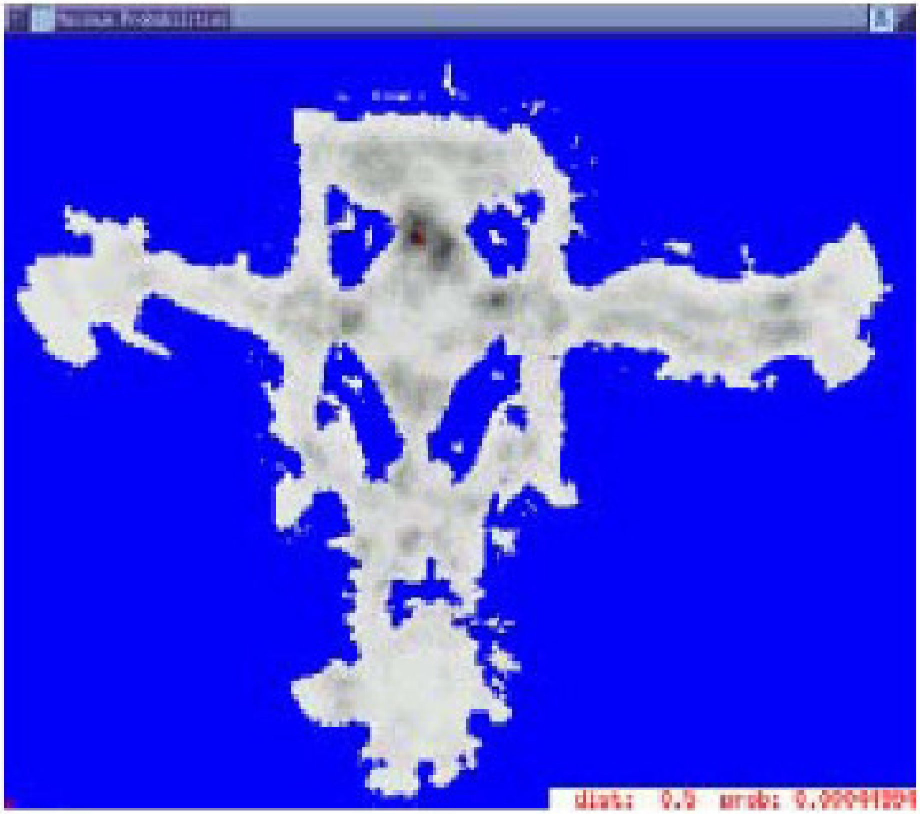
- Example 1: Office Building

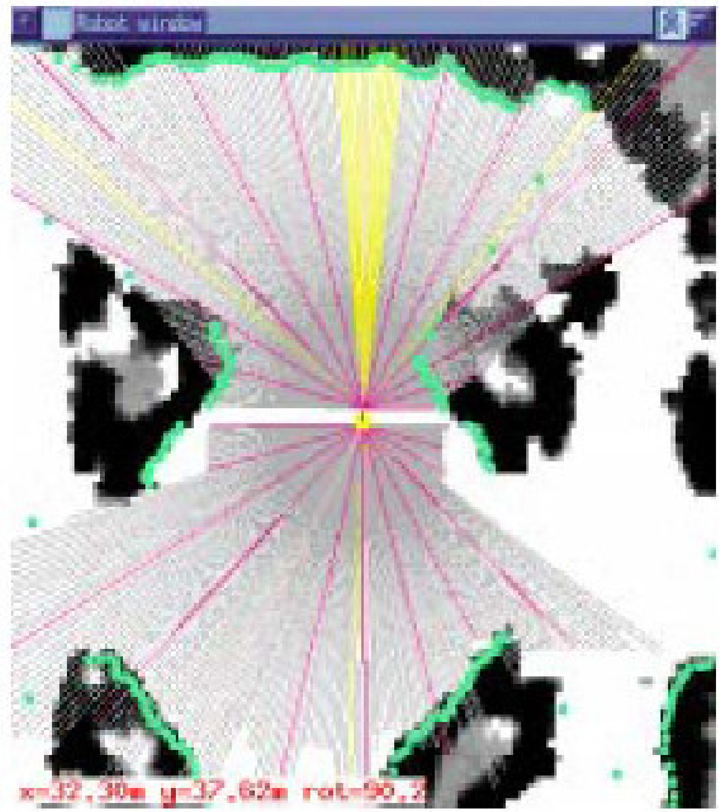


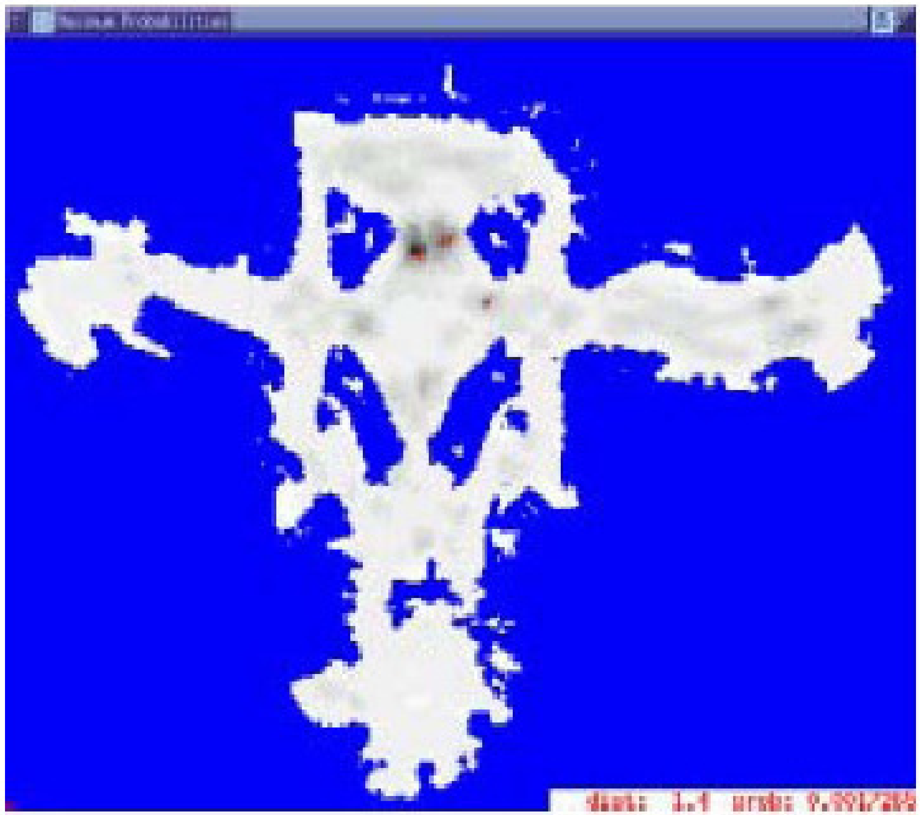
Real Example 2

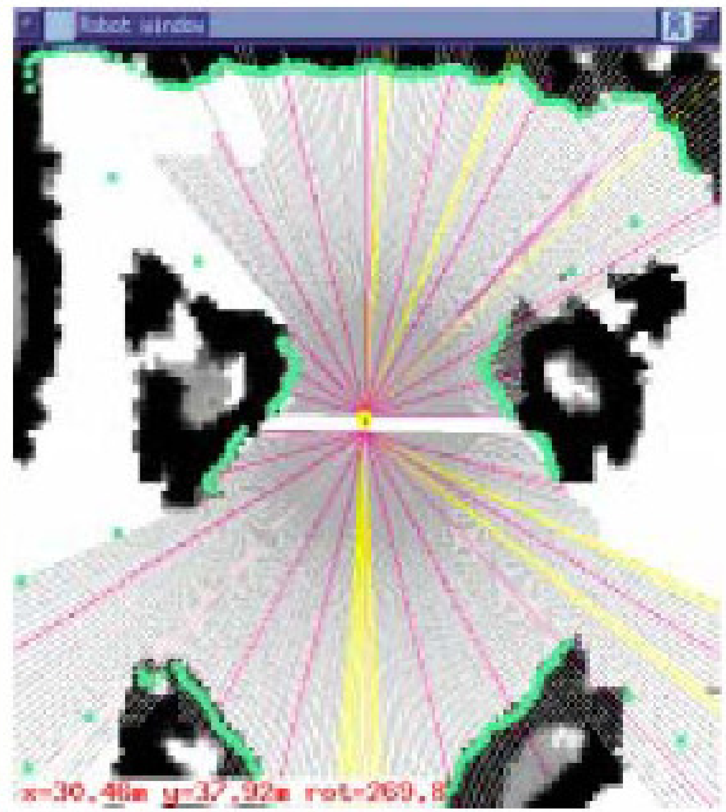


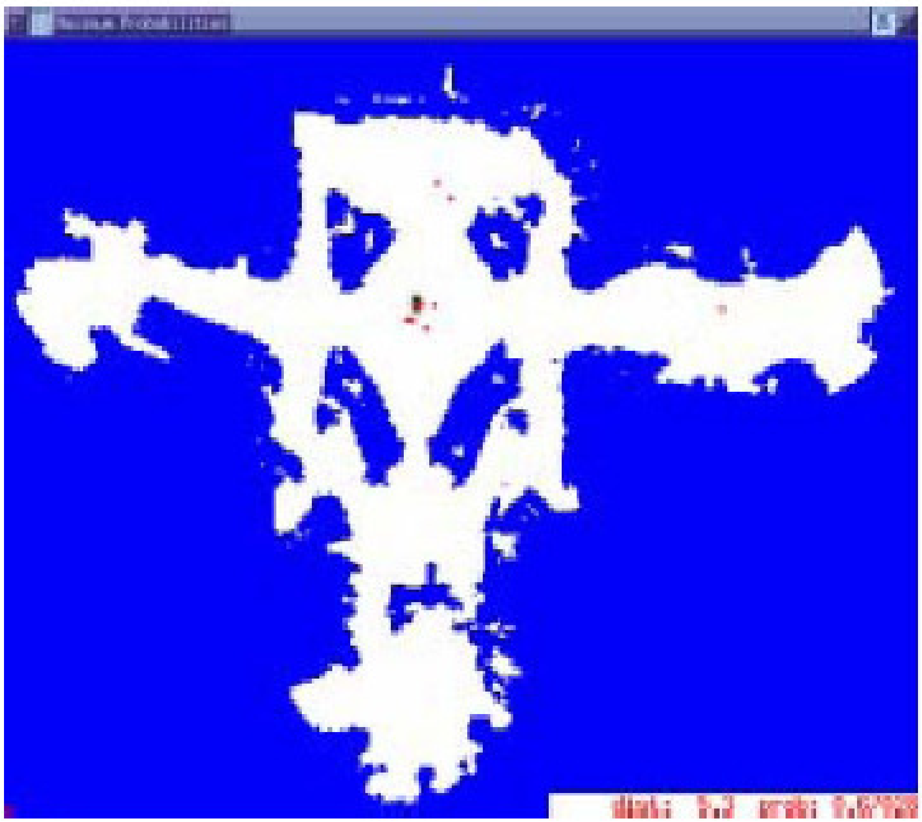


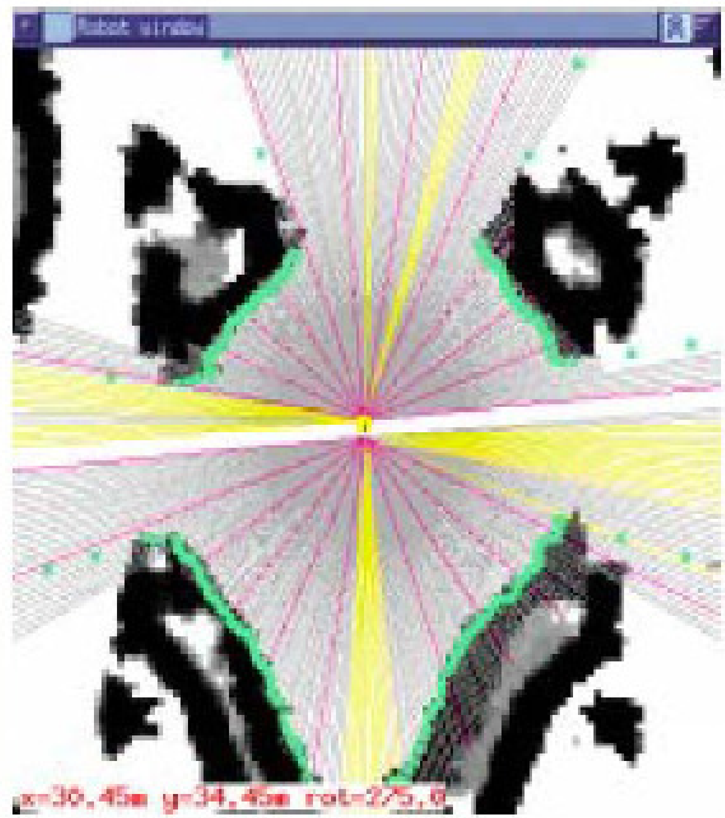




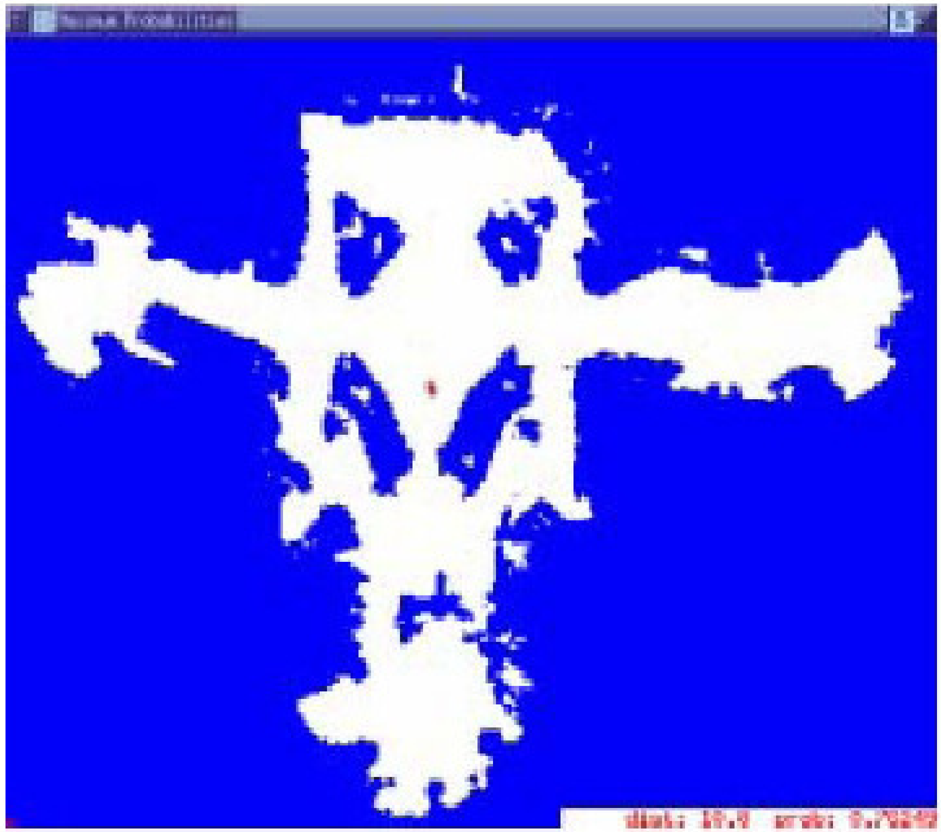


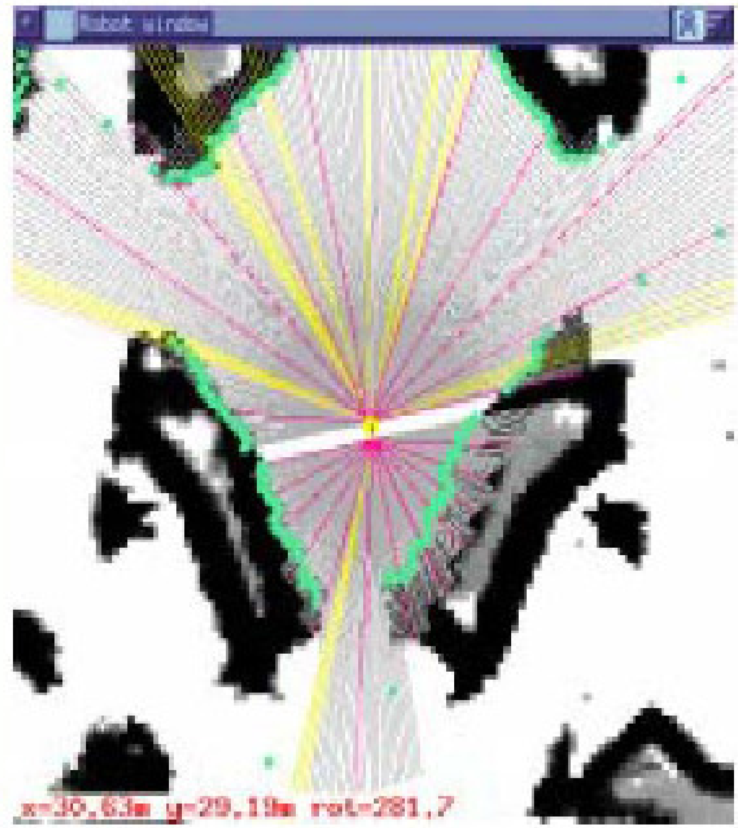






x=30.45m y=34.45m rot=275.0




Robot window
x=30.63m y=29.19m rot=281.7