# Markov Localization and Environment Mapping

Lecture 7 – Thursday November 17, 2016

# Objectives

When you have finished this lecture you should be able to:

- Understand **Markov localization**.
- Understand **Occupancy grid framework** and how to use it in sonar-based **environment mapping**.

# Outline

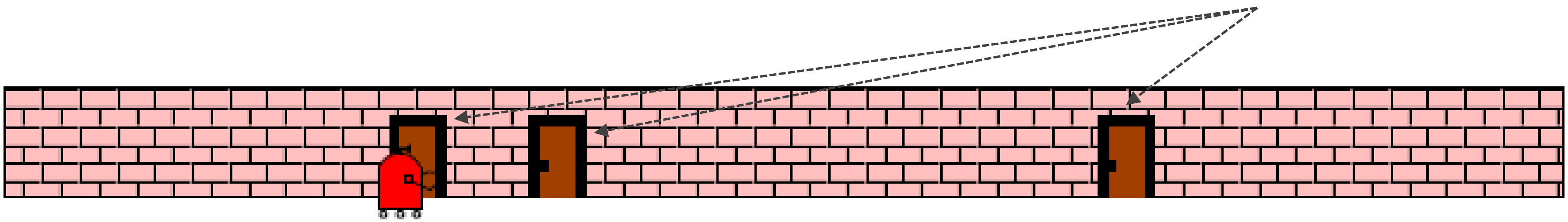
- Markov Localization
- World Modeling
- Metric Mapping
- Occupancy Grid Framework
- Occupancy Grid-based Mapping
- Case Study
- Summary

# Outline

- <u>**Markov Localization**</u>
- World Modeling
- Metric Mapping
- Occupancy Grid Framework
- Occupancy Grid-based Mapping
- Case Study
- Summary

# Markov Localization

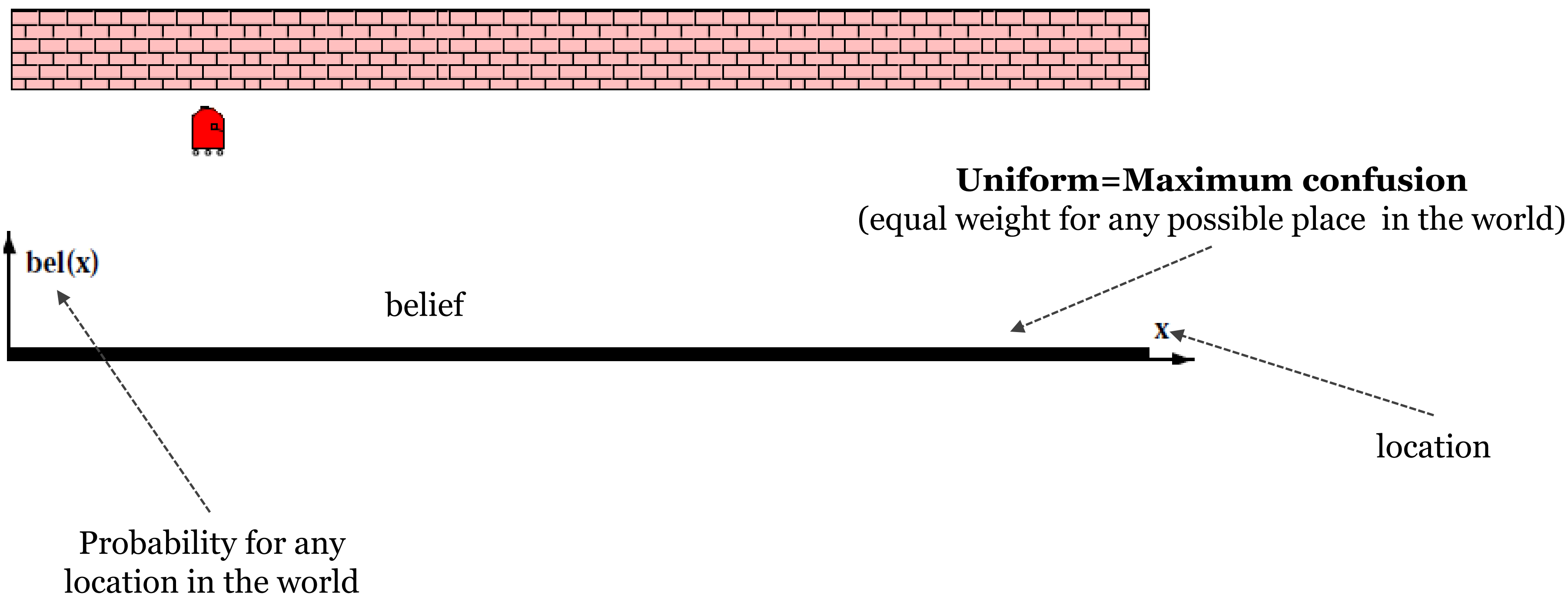
- **Global localization:** a robot is placed somewhere in the environment and has to localize itself from scratch using artificial/natural landmarks (like doors).



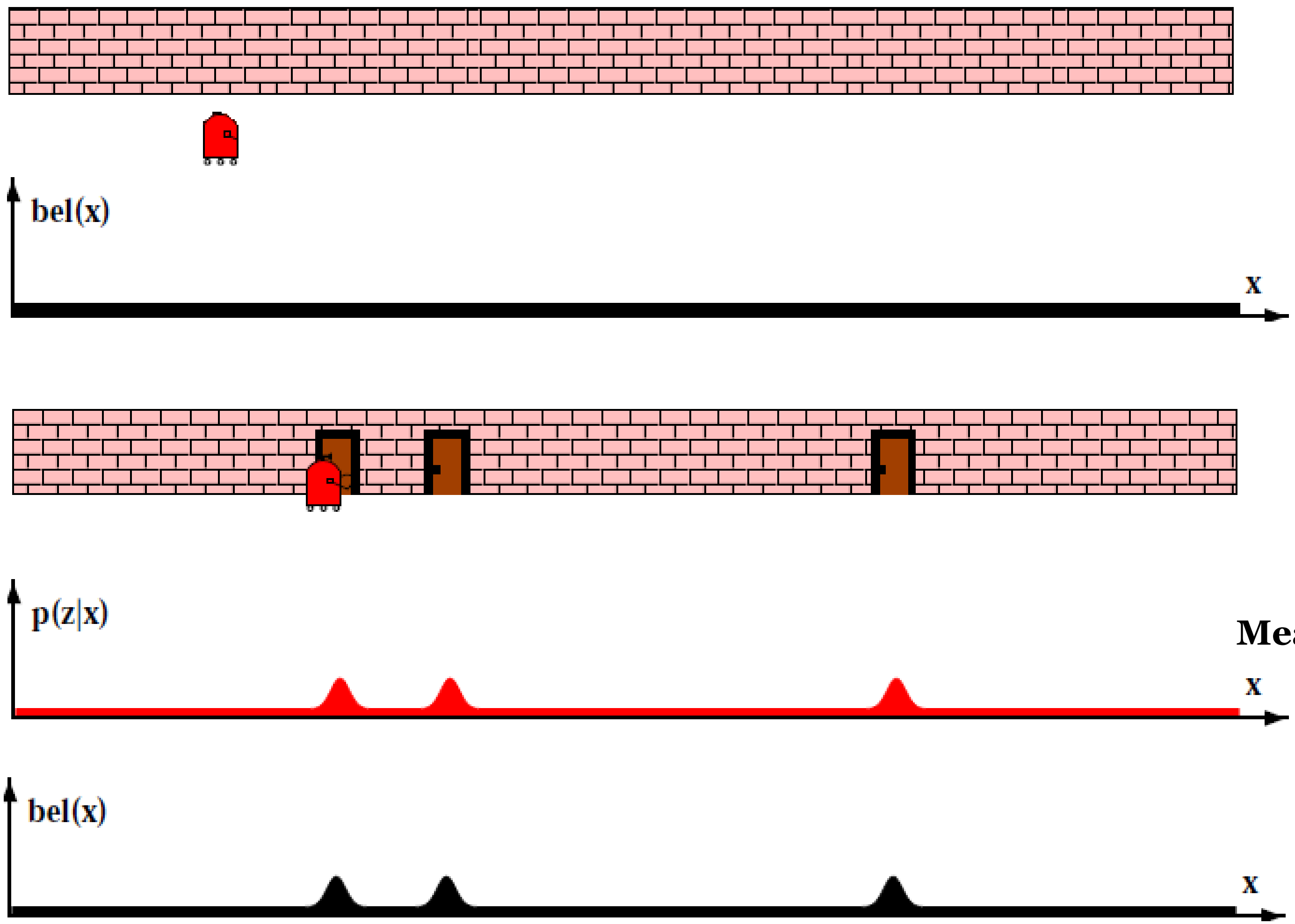

A mobile robot during global localization

# Markov Localization

- In the probabilistic paradigm, the **robot's momentary estimate** (also called **belief**) is represented by a **probability density function** over the space of all locations.



# Markov Localization



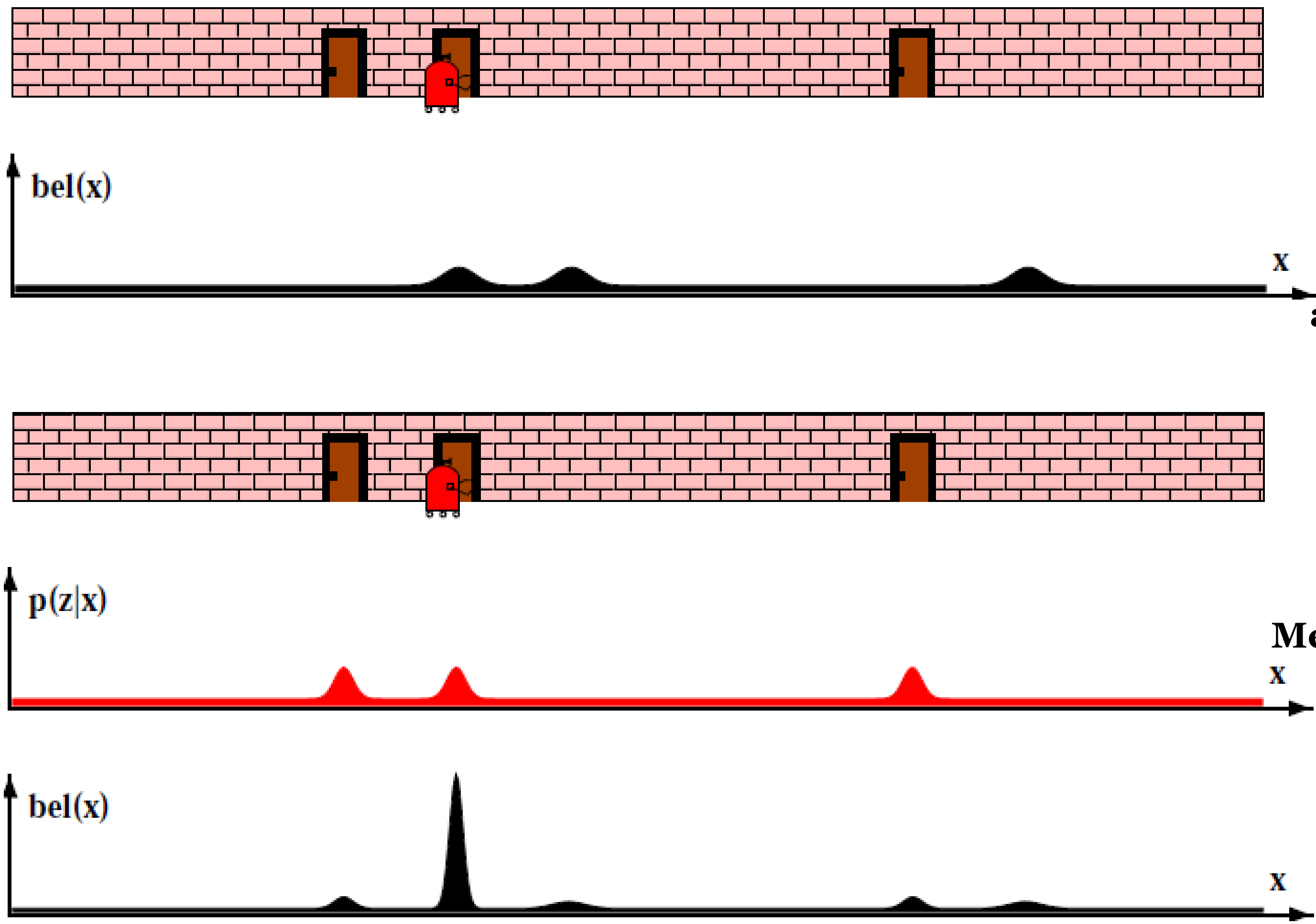

**Prior Belief**

**Robot detects landmarks (doors)**

**Measurement Probability (Likelihood)**

**Posterior Belief**
after the robot's sense measurement has been taken

# Markov Localization



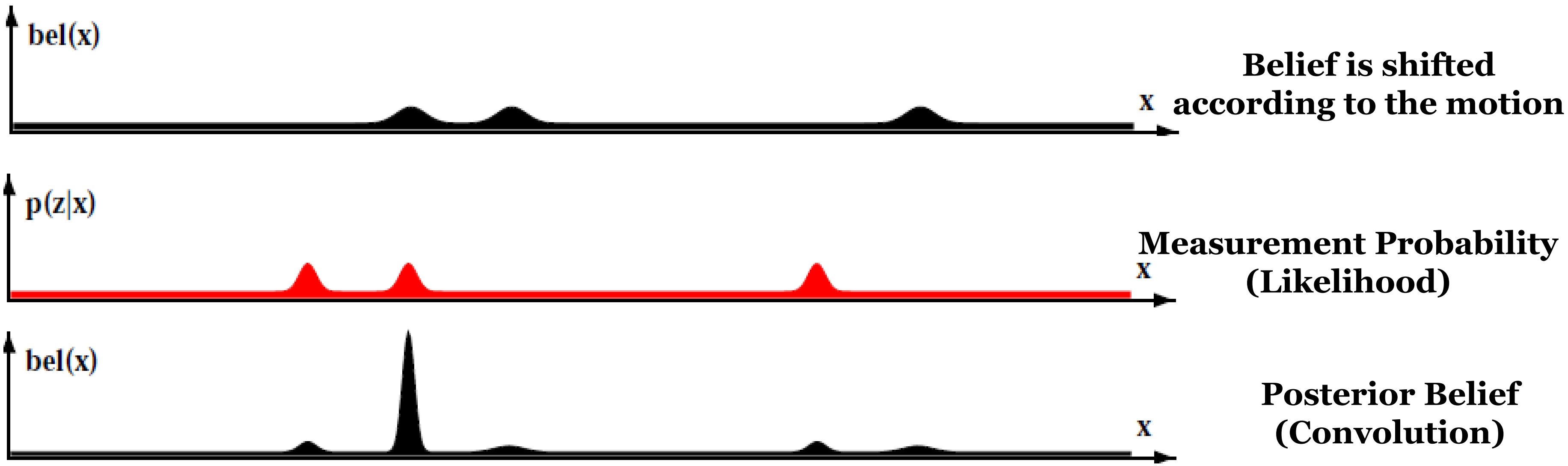

**Robot moves to the right a certain distance**

**Belief is shifted according to the motion**

**Measurement Probability (Likelihood)**

**Posterior Belief (Convolution)**

# Markov Localization



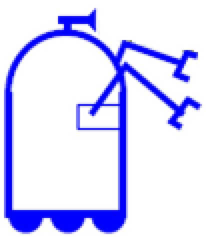

**Convolution** (similar to cross-correlation) is a mathematical operation that takes two functions and measures their overlap. It measures the amount of overlap as you slide one function over another. For example, if two functions have zero overlap, the value of their convolution will be equal to zero. If they overlap completely, their convolution will be equal to one.

More and animation: http://en.wikipedia.org/wiki/Convolution

# Markov Localization

- **Example**

| Green | Red | Red | Green | Green | |
|---|---|---|---|---|---|
| $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | **Cells** |



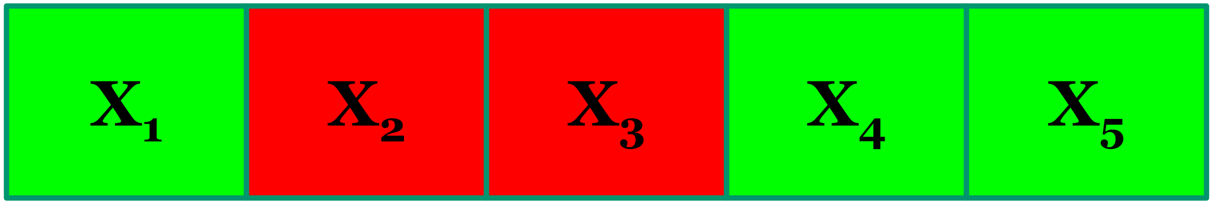
**Robot**

$$p(X_i) = 0.2 \quad \forall i = 1,2,3,4,5$$

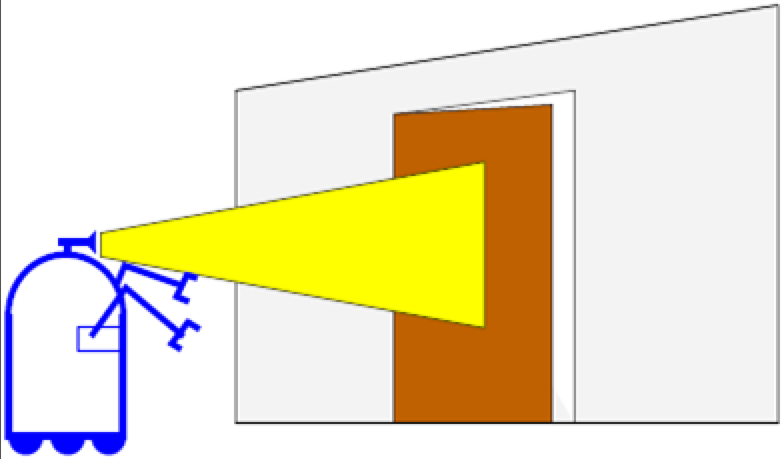
Robot's initial belief about its location in the world

# Markov Localization

- **Example**




**Cells**



The robot is equipped with an uncertain **sensor** so it can sense itself in a red/green cell.

Assume that robot senses itself in a **red cell**.

Assume that the following rule represents the **probability** that the robot is in a red or a green cell, based on the robot's measurement of "red":

red cells * 0.6
green cells * 0.2

# Markov Localization

- **Example**

| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | Cells |
|---|---|---|---|---|---|---|
| $p(X_i)$ | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | |
| $p(X_i \mid Z)$ | 0.04 | 0.12 | 0.12 | 0.04 | 0.04 | |

*After measurement*

*Note:*
red cells * 0.6
green cells * 0.2

$$\sum = 0.36 \neq 1$$

These are not our probabilities, because they **do not add up to one**. Since we know we are in some cell, the individual probabilities must add up to one. What we have now is called an **unnormalized probability distribution**.

# Markov Localization

- **Example**

| | $X_1$ | $X_2$ | $X_3$ | $X_4$ | $X_5$ | Cells |
|---|---|---|---|---|---|---|
| $p(X_i)$ | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | |
| $p(X_i \mid Z)$ | 0.04 | 0.12 | 0.12 | 0.04 | 0.04 | |

**Normalized probability distribution:**

$$\sum_{i=1}^{5} p(X_i \mid Z) = 0.04 + 0.12 + 0.12 + 0.04 + 0.04 = 0.36$$

| | | | | | | |
|---|---|---|---|---|---|---|
| $\dfrac{p(X_i \mid Z)}{\sum_{i=1}^{5} p(X_i \mid Z)}$ | $\frac{1}{9}$ | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{9}$ | $\frac{1}{9}$ | $\Rightarrow \sum = 1$ |

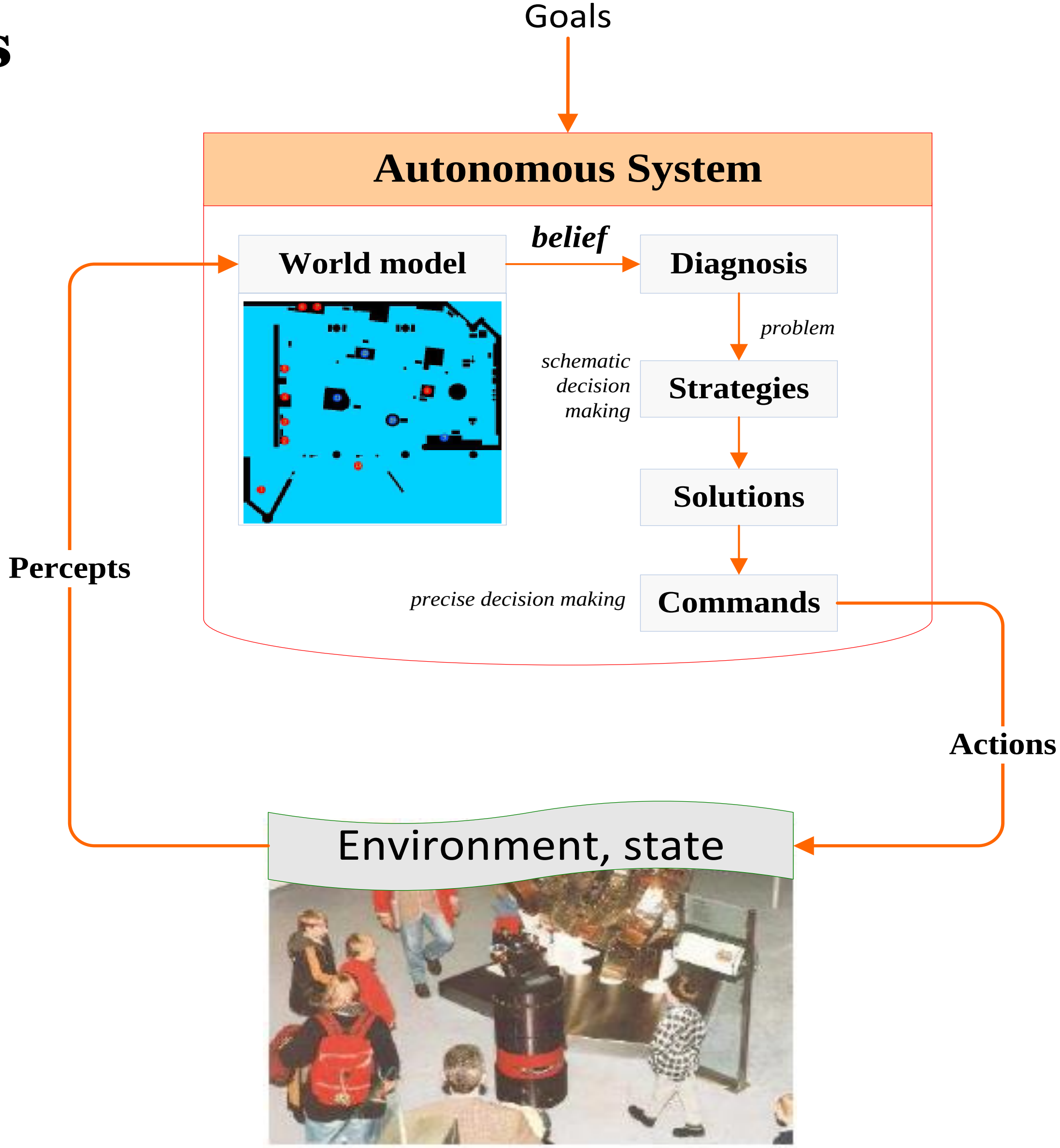
# Outline

- Markov Localization
- **World Modeling**
- Metric Mapping
- Occupancy Grid Framework
- Occupancy Grid-based Mapping
- Case Study
- Summary

# World Modeling

The construction of **models of the environment** is crucial to the development of several applications of mobile robot systems.

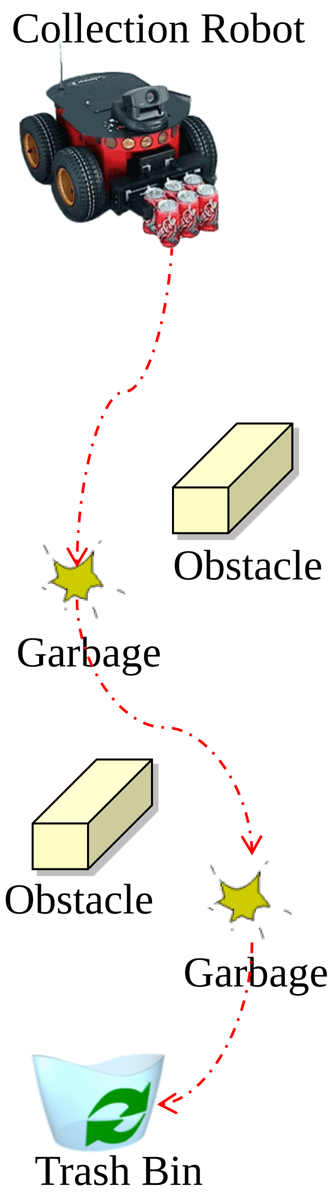
It is through these environment models that the robot can **adapt its decisions to the current state of the world**.



# World Modeling

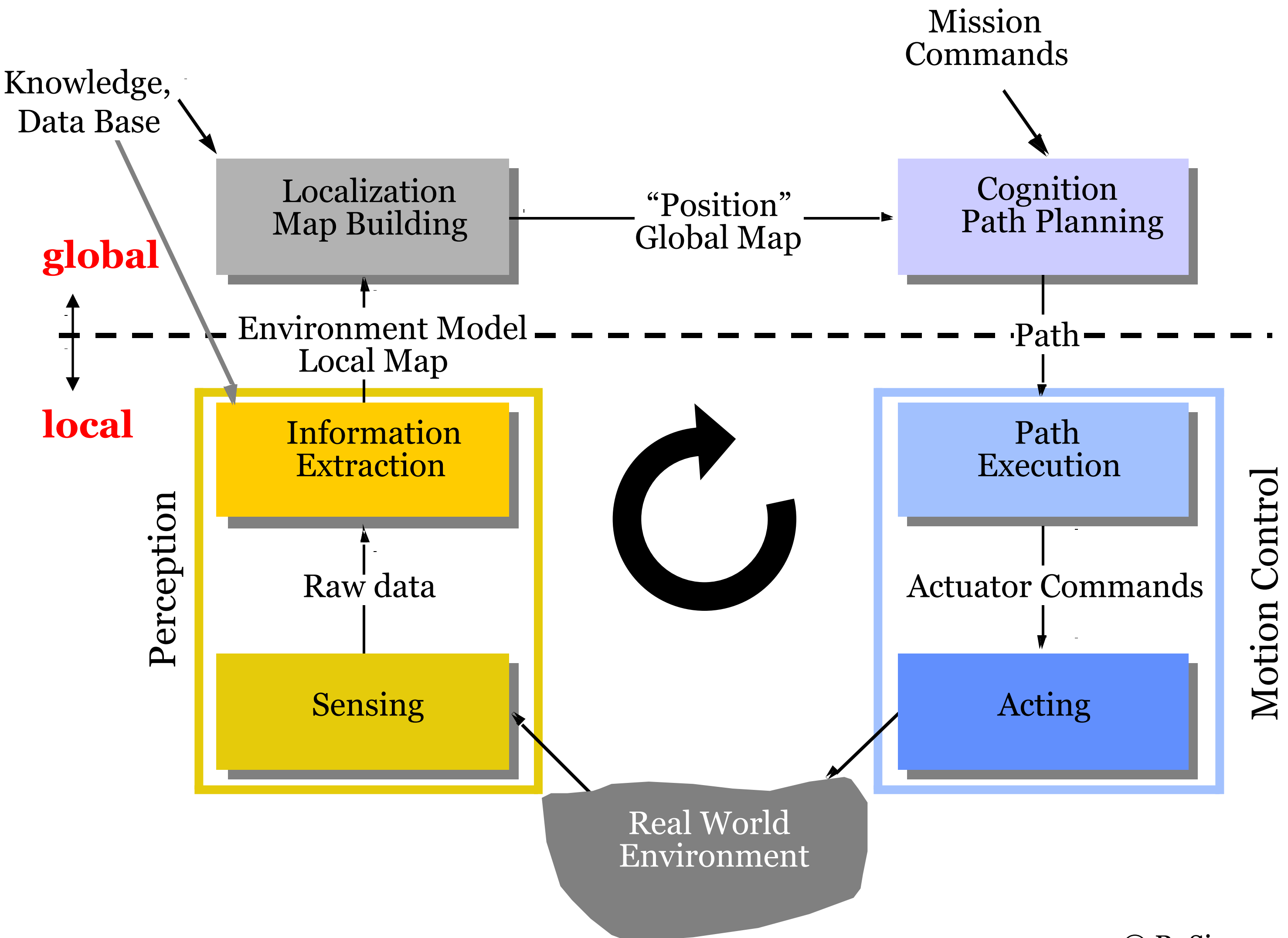
- **Mapping in Robot Architecture:**

| Process | Tasks |
|---|---|
| Measurement | ◇ Input user commands<br>◇ Odometry<br>◇ Sense the environment<br>◇ Detect objects |
| Modeling | ◇ Map the environment<br>◇ Extract features<br>◇ Model objects<br>◇ Map Paths |
| Comprehension | ◇ Find paths<br>◇ Detect collision situations<br>◇ Learn the map |
| Planning | ◇ Decompose task into subgoals<br>◇ Select a path<br>◇ Choose alternatives when path is blocked |
| Action | ◇ Navigate<br>◇ Traverse path and avoid collisions<br>◇ Control based on kinematic and dynamic models of robot |



# World Modeling

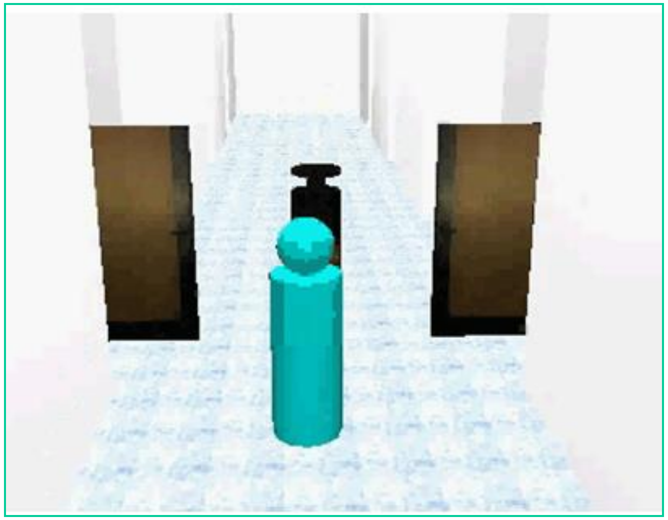
- **Mapping in Robot Architecture:**




© R. Siegwart, I. Nourbakhsh

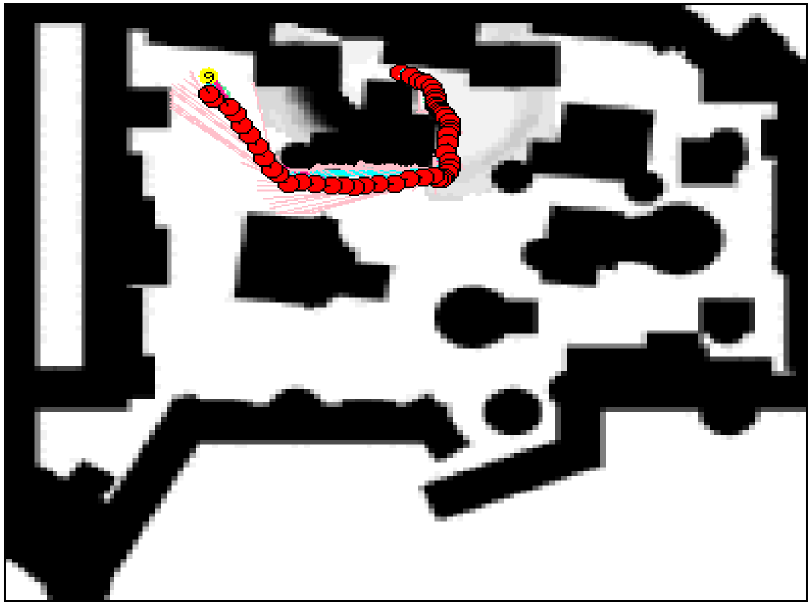
# World Modeling

- Generally, the **map isn't static** while mapping:
  - ◇ people walking by,
  - ◇ moving objects.
- Often the world changes over time:
  - ◇ doors open or close,
  - ◇ objects are moved around,
  - ◇ plants grow…
- Typical resulting problems:
  - ◇ Bad alignments (localization),
  - ◇ spurious objects (mapping)



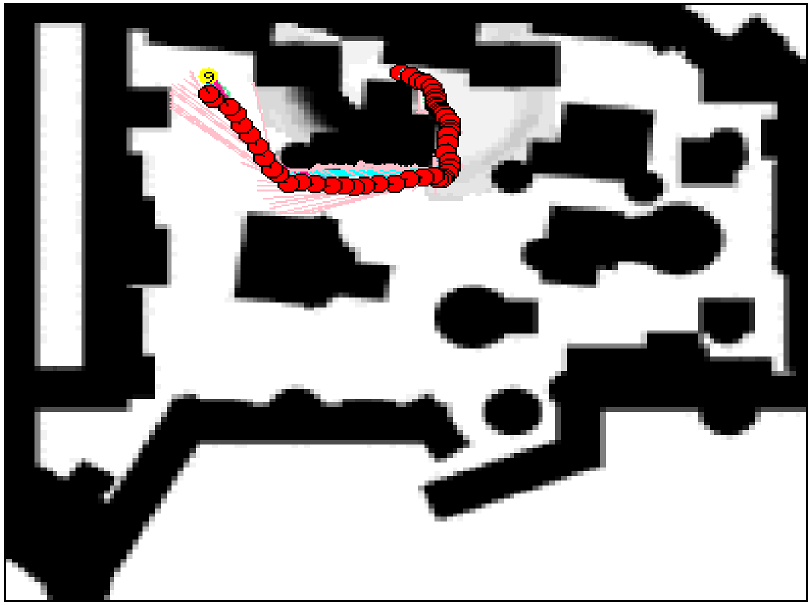
# World Modeling

- Mapping provides an answer for the question “**What does the world look like?**”
- Mapping is the problem of **generating maps from sensor measurements**. The information provided by the sensors has to be interpreted in a way that is sufficiently a structured knowledge of the space that surrounds the robot is obtained by the robot.



# World Modeling

- Structuring this external information is accomplished through the construction of a **model or a map** of the environment.
- This model is used to **make decisions** and to fuse the new information that the robot receives from its sensors, either from the same sensor in several time instants or from several sensors of the same or different type.

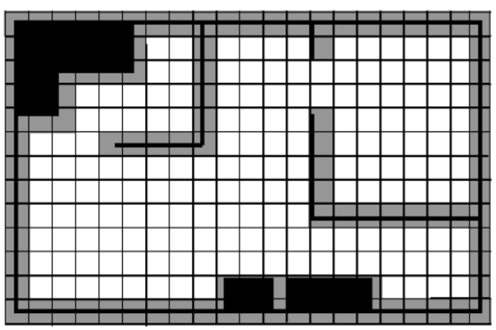


# World Modeling

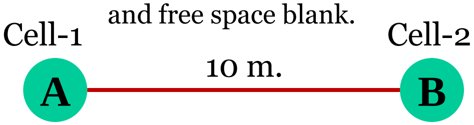
## Maps

### Metric (grid-based) Map

Metric maps capture the geometric properties of the environment.



Map divided into evenly size square with obstacle shaded black (occupied) and free space blank.

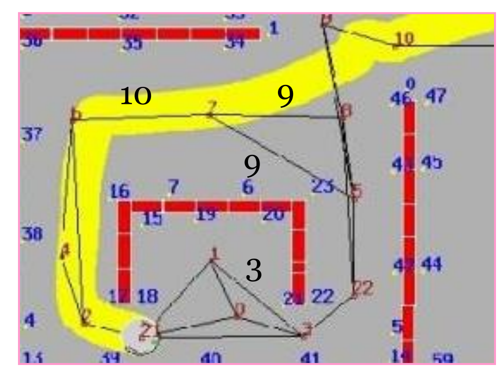


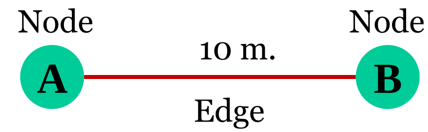

### Topological Map

Topological maps describe the connectivity of different places by means of nodes-and-arcs graphs


### Metric Topological Map

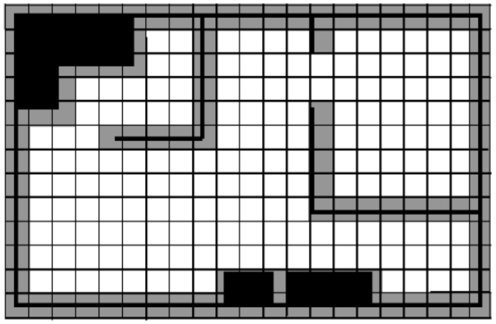
Metric Topological maps posses geometry relation between path





# World Modeling

- **Metric (grid based) Map**



◇ **Characteristics**

- cell-based structure
- store information of obstacle and spatial relationship

◇ **Pros**

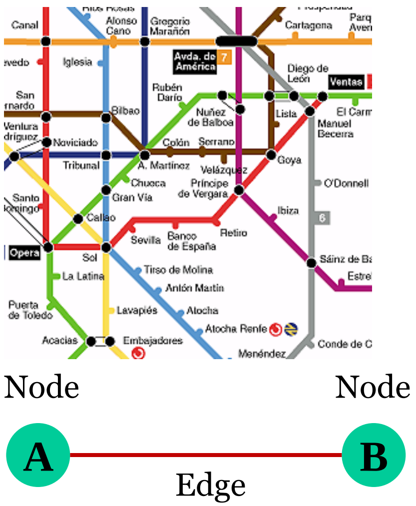
- easy to construct.
- useful in map matching.
- can dissimilar identical places or objects.
- enable estimation of robot's and obstacle's pose.
- sensitive to noise.

◇ **Cons**

- require huge storage.
- large computation time.
- path planning may not be efficient, but the path chose may be shorter than that of Topological Map.

# World Modeling

- **Topological Map**
  - ◇ **Characteristics**
    - ▪ graph-based structure
    - ▪ no geometry relation between path.




◇ **Pros**

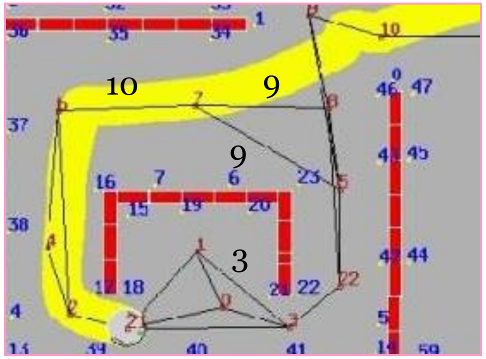
- ▪ require less storage
- ▪ less computation time
- ▪ faster path planning using Dijkstra Algorithm for example (next lecture), but path may not the shortest.

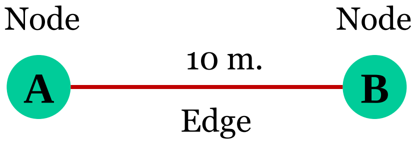
◇ **Cons**

- ▪ harder to construct
- ▪ not valid for map matching
- ▪ perceptual aliasing in recognizing identical place.
- ▪ cannot estimate the position of robot and obstacle.

# World Modeling

- **Metric Topological Map**
  - ◇ **Characteristics**
    - graph-based structure
    - possesses geometry relation between path.






◇ **Pros**
- require less storage
- less computation time
- path planning algorithms more optimal compare to Topological Map.

◇ **Cons**
- harder to construct
- not valid for map matching
- perceptual aliasing in recognizing identical place.
- cannot estimate the position of robot and obstacle.

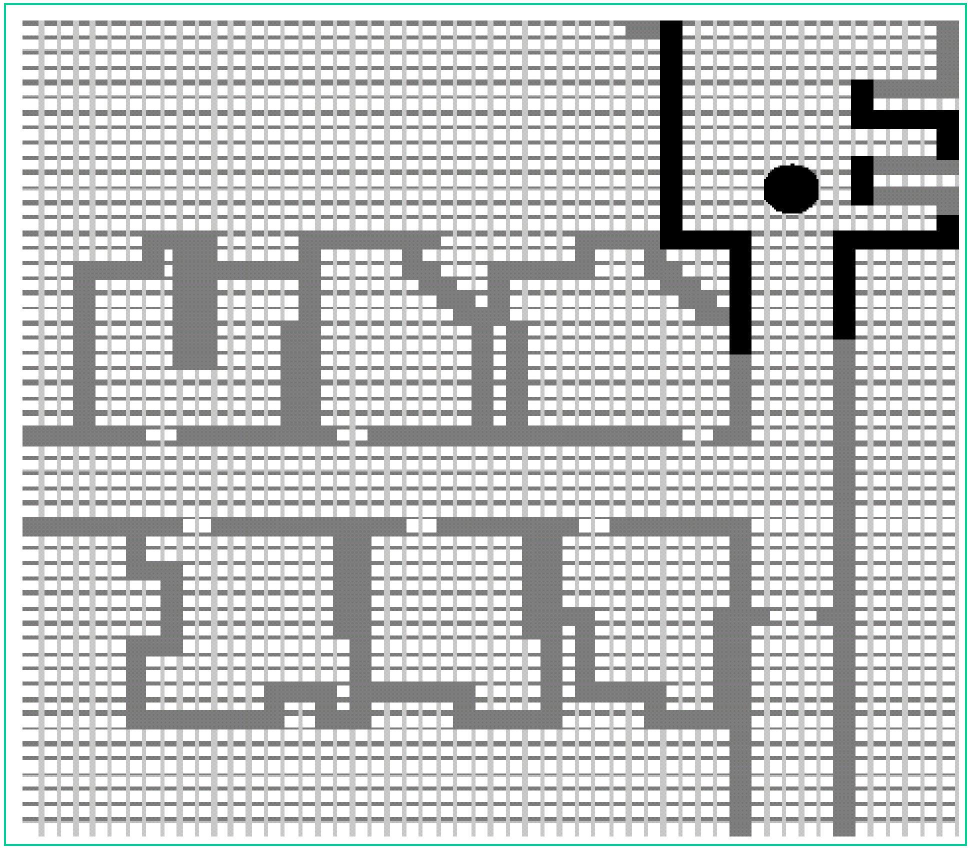
# Outline

- Markov Localization
- World Modeling
- **Metric Mapping**
- Occupancy Grid Framework
- Occupancy Grid-based Mapping
- Case Study
- Summary

# Metric Mapping

- Occupancy grid maps, which were introduced in the 1980s by **Moravec and Elfes***, are a popular, probabilistic metric mapping approach to represent the environment.

- They are an **approximative technique** in which we calculate for each cell of a discrete grid the posterior probability that the corresponding area in the environment is occupied by an obstacle.



*H.P. Moravec, A.E. Elfes: High Resolution Maps from Wide Angle Sonar, Proc. IEEE Int. Conf. Robot. Autom. (ICRA), 1985.

# Metric Mapping

- **Advantage of occupancy grid maps:**
  - ◇ Do not rely on any predefined features.
  - ◇ Additionally, they offer a constant-time access to grid cells and provide the ability to **represent unknown (unobserved) areas**, which can be important, for example, in exploration tasks.

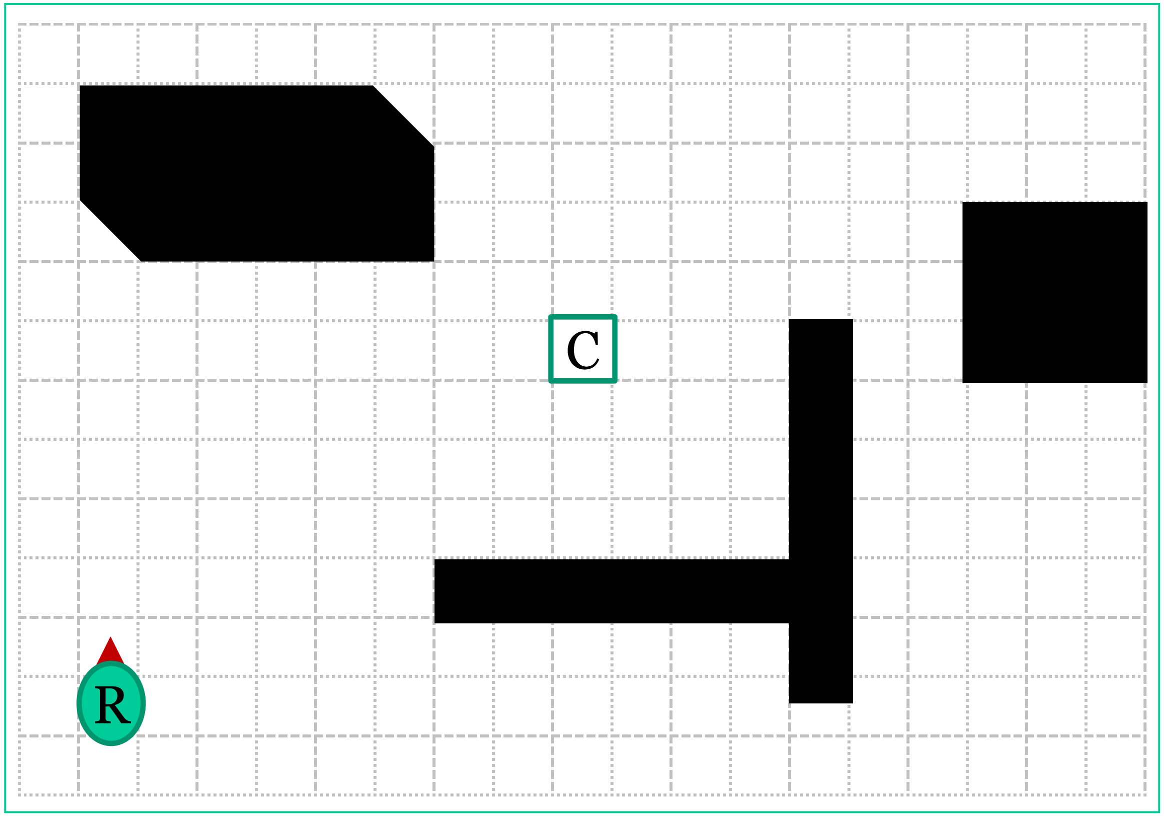
- **Disadvantage of occupancy grid maps:**
  - ◇ Potential discretization errors and
  - ◇ High memory requirements.

# Outline

- Markov Localization
- World Modeling
- Metric Mapping
- **Occupancy Grid Framework**
- Occupancy Grid-based Mapping
- Case Study
- Summary

# Occupancy Grid Framework

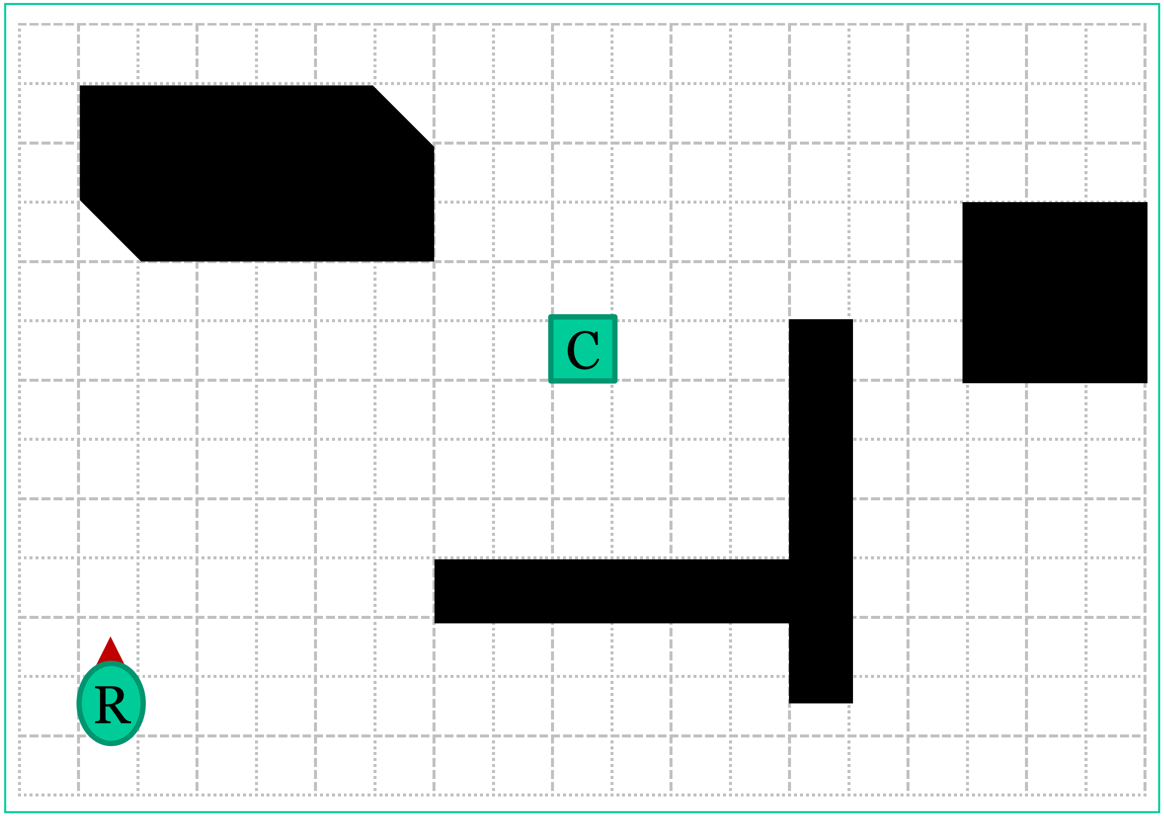
- **Occupancy Grid Representation:**
  - ◇ The occupancy grid representation employs a multidimensional (typically 2D or 3D) tesselation of space into cell, where each cell stores a probabilistic estimate of its state.




  - ◇ R: a robot equipped with sensors,
  - ◇ C: a cell in the occupancy grid.

# Occupancy Grid Framework

- **Occupancy Grid Representation:**
  - ◇ The state variable **s(C)** associated with a cell C of the occupancy grid is defined as a discrete random variable with two states, occupied (OCC) and empty (EMP).




- ◇ p[s(C)=OCC]

  +

  p[s(C)=EMP]=1

# Occupancy Grid Framework

- **Estimating the Occupancy Grid:**
  - ◇ Since a robot can only obtain information about its environment indirectly, through its sensors, the recovery of a spatial world model from sensor data is best modeled as estimation theory problem.

# Occupancy Grid-based Mapping

- **Estimating the Occupancy Grid: Basic Idea**


B21r in the corridor

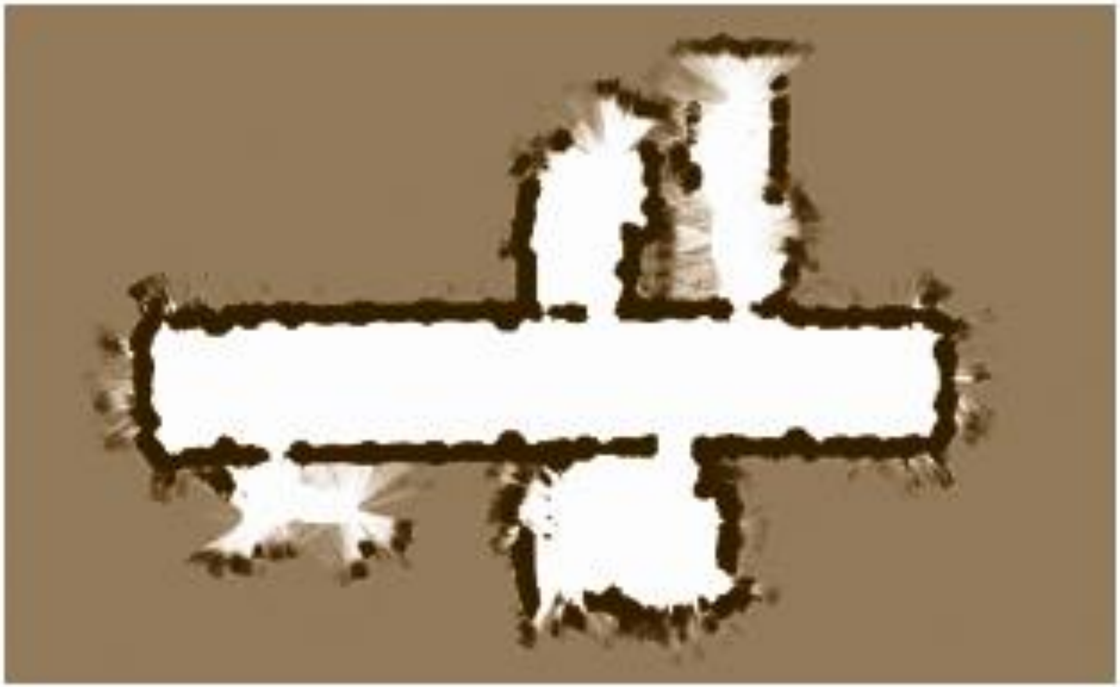
# Occupancy Grid-based Mapping

- **Estimating the Occupancy Grid: Incremental Mapping**


- ◇ A map is built from a sequence of previous ultrasound scans.
- ◇ Afterwards the robot perceived a series of 18 ultrasound scans, each consisting of 24 measurements.
- ◇ The occupancy probabilities for these 18 scans are depicted in rows 2–7.
- ◇ The occupancy probability grid is obtained by integrating the individual observations into the map [converges to corridor structure ].

# Occupancy Grid-based Mapping

- **Estimating the Occupancy Grid: Incremental Mapping**

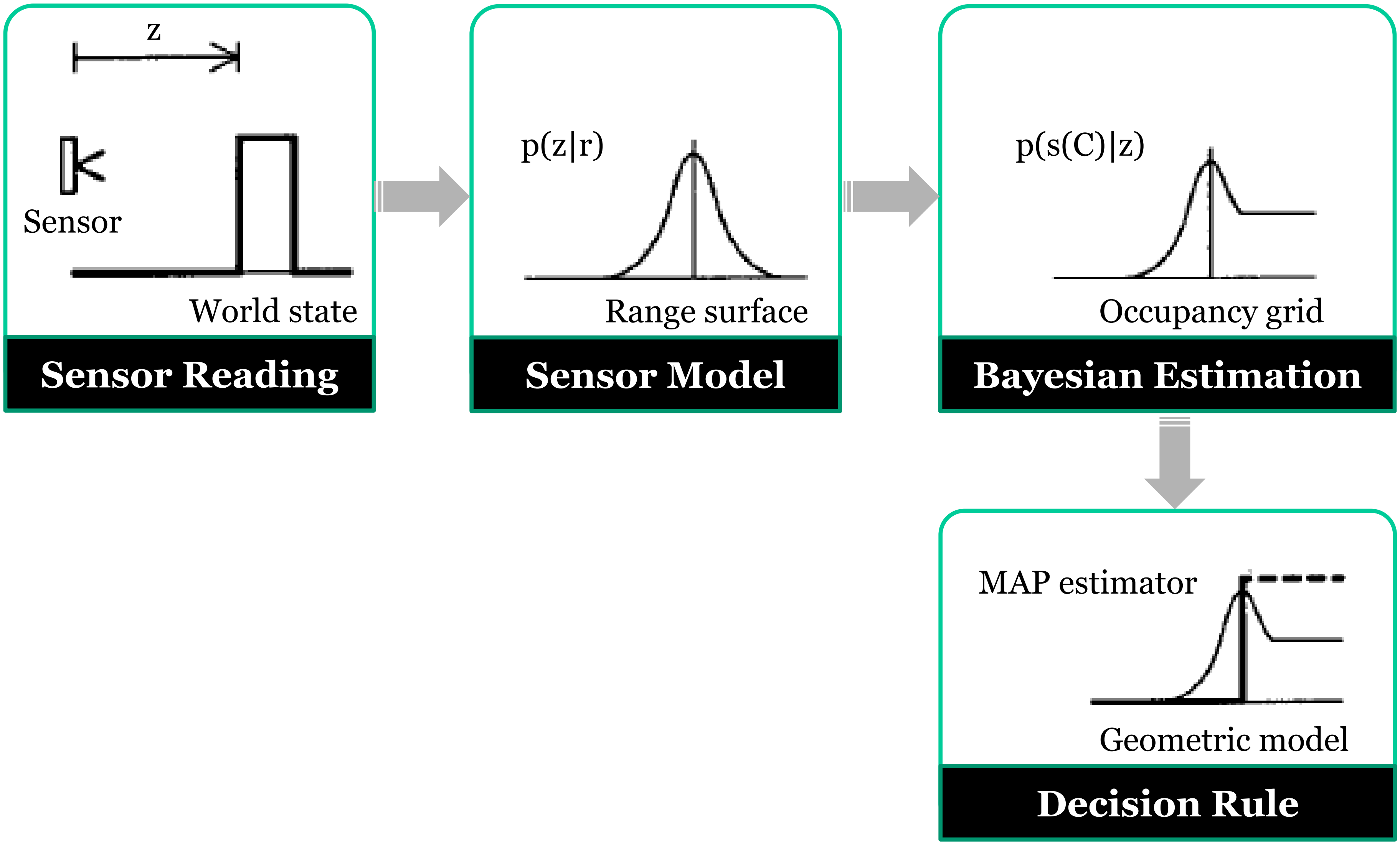


Occupancy grid map obtained from ultrasound data

B21r in the corridor

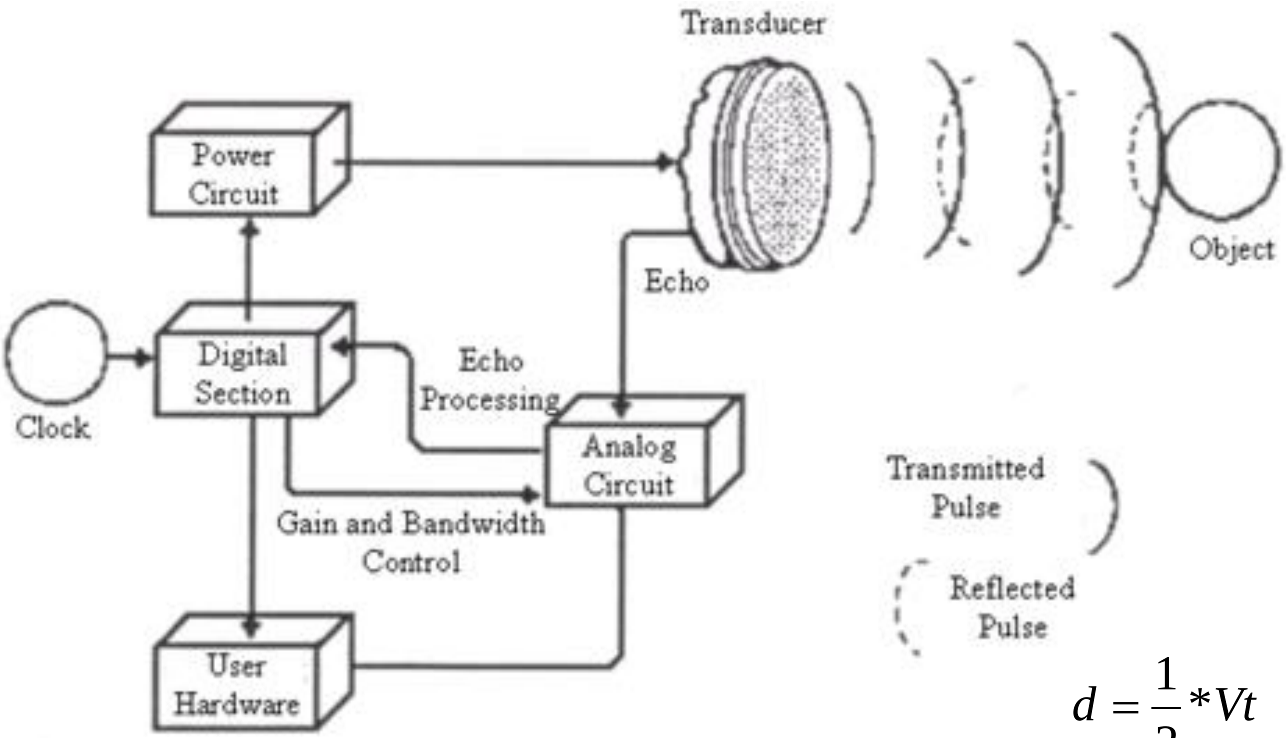
# Occupancy Grid Framework

- **Estimating the Occupancy Grid:**



# Occupancy Grid Framework

- **Sensor Model:**
  - ◇ Sonar sensors are used in this method.




$$d = \frac{1}{2} * Vt$$

V = speed of sound = 331 + 0.6T m/s

T is the temperature in $^{o}C$

$t$ is the time of travel

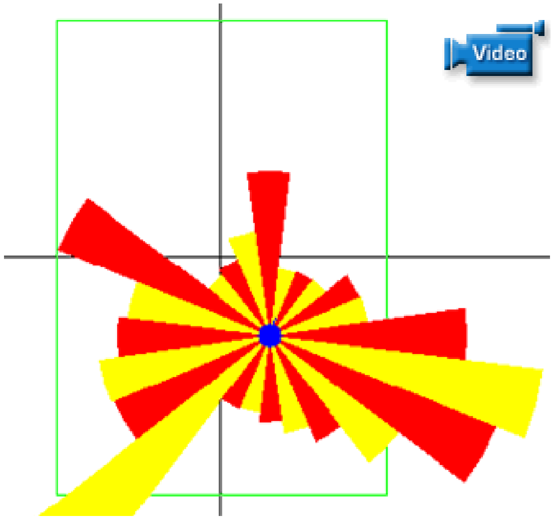
# Occupancy Grid Framework

- **Sensor Model:**
  - ◇ Sonar sensors

| **Pros** | **Cons** |
|---|---|
| ◇ relatively low cost<br>◇ fast computational time due to less or no process of determine obstacle position<br>◇ able volumetric sensing. | ◇ inaccurate and noisy, as roughness surface causes scattering reflections or angle of reflection is too large that acoustic pulse reflected is away from receiver<br>◇ specular reflections give rise to erroneous readings<br>◇ arrays of sonar sensors can experience crosstalk, which one sensor detects the reflected beam of another sensor<br>◇ unable to determine the exact position of objects |

# Occupancy Grid Framework

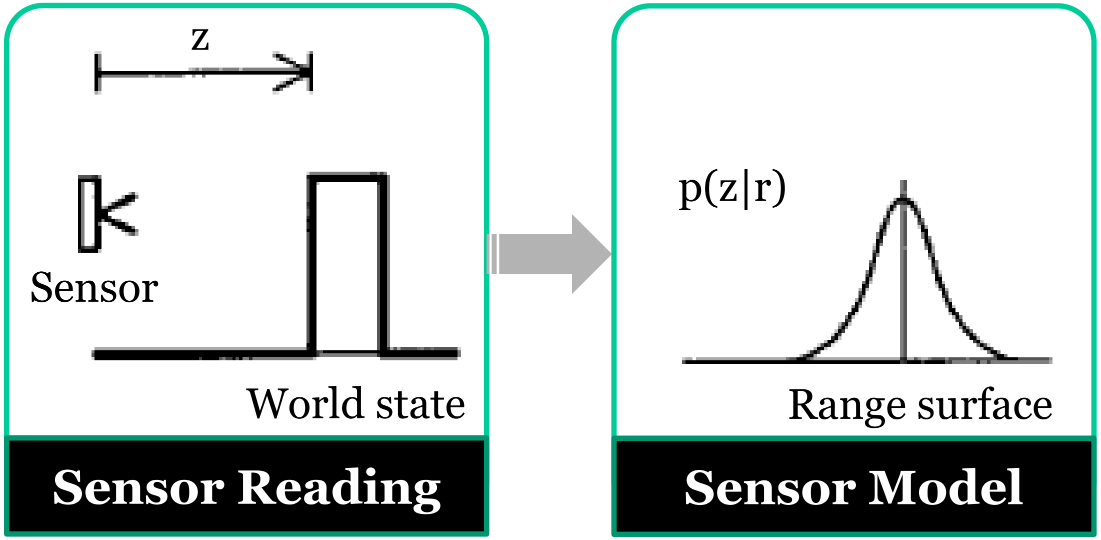
- **Sensor Model:**
  - ◇ Sensors can detect in range 27 cm – 10.5 m
  - ◇ Accuracy ±3 cm.
  - ◇ 24 transducers, configured in a ring, spaced 15° apart.
  - ◇ To avoid interference sensors must be fired sequentially (~200ms/firing).




B21

# Occupancy Grid Framework

- **Sensor Model:**

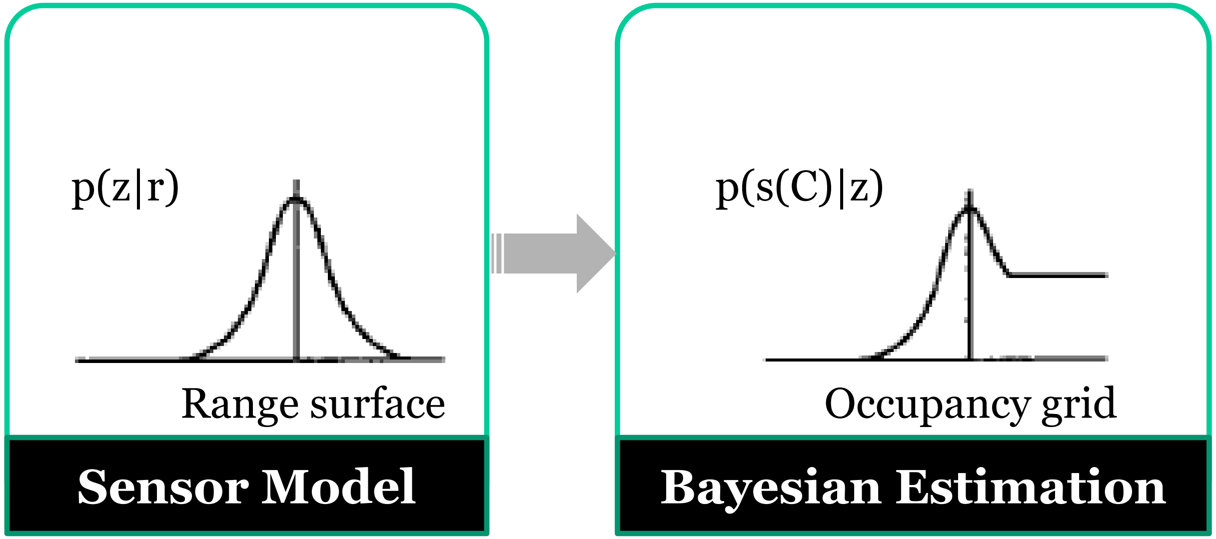



◇ To interpret the range data obtained from a given sensing device, **stochastic sensor model** is used. This model is defined by a probability density function (pdf).

◇ This pdf is of the form **p(z|r)** and relates reading/observation of measurement **z** with the true parameter range value **r**.

# Occupancy Grid Framework

- **Bayesian Estimation Process:**




◇ **Bayesian estimation** procedure is used to determine the **occupancy grid cell state probabilities**.

◇ To allow incremental composition of sensory information, **sequential updating formulation of Bayes' theorem** is used.

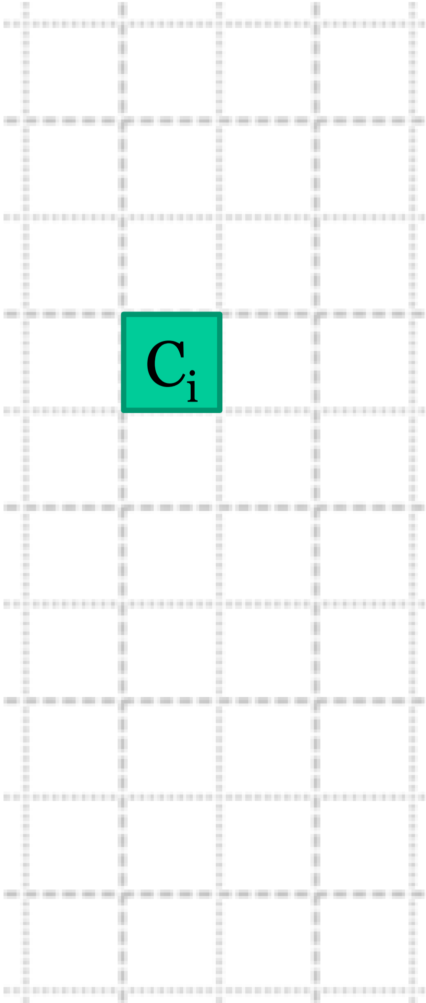
# Occupancy Grid Framework

- **Bayesian Estimation Process:**
  - ◇ Given a current estimate of the state of the cell $C_i$

    $p[s(C_i)=OCC|\{z\}_t]$

    based on observation $\{z\}_t=\{z_1,....,z_t\}$

  - ◇ For a new observation $\{z\}_{t+1}$, the improved estimate is give by

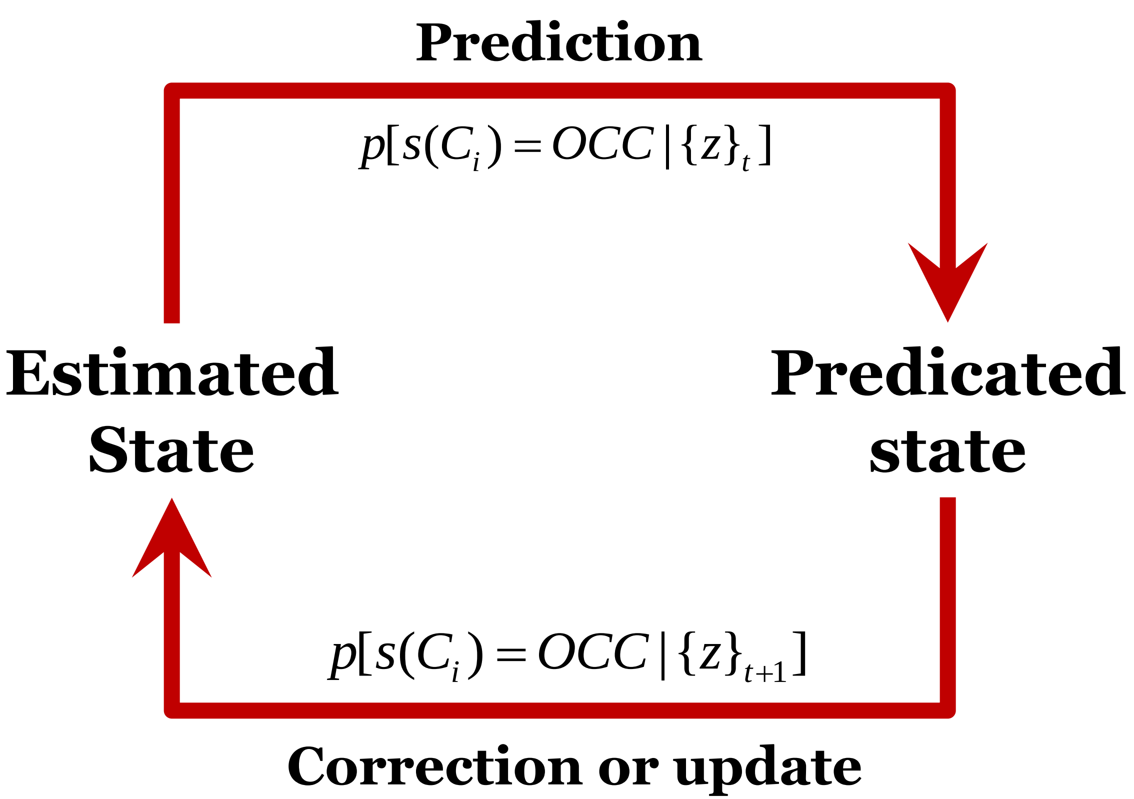



$$p[s(C_i)=OCC\,|\,\{z\}_{t+1}] = \frac{p[z_{t+1}\,|\,s(C_i)=OCC].p[s(C_i)=OCC\,|\,\{z\}_t]}{\sum_{s(C_i)} p[z_{t+1}\,|\,s(C_i)].p[s(C_i)\,|\,\{z\}_t]}$$

$$\text{new cell state estimate} = \frac{\text{likelihood.previous cell state estimate}}{\text{evidence}}$$

# Occupancy Grid Framework

- **Bayesian Estimation Process:**
  - ◇ In this recursive formulation, the previous estimate of the cell state, $\mathbf{p[s(C_i)=OCC|\{z\}_t]}$, serves as the prior and is obtained directly from the occupancy grid.
  - ◇ The new cell state estimate , $\mathbf{p[s(C_i)=OCC|\{z\}_{t+1}]}$ , is sequentially stored again in the map.

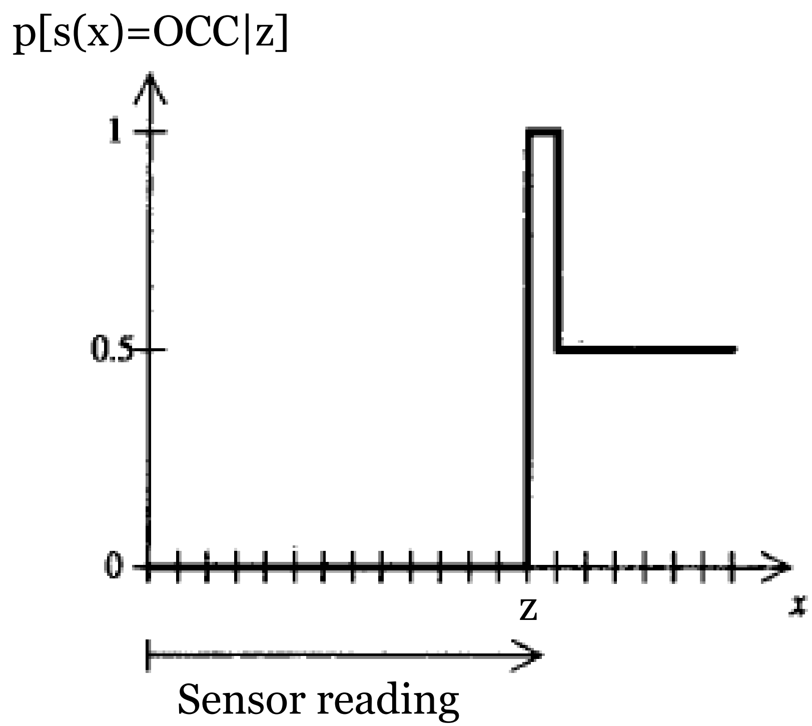


# Occupancy Grid Framework

- **Bayesian Estimation Process:**
  - ◊ Obtaining **p[z|s($C_i$)]** distribution from the sensor model **p(z|r)** is done using **Komogoroff's theorem**.
  - ◊ For **Ideal range sensor**:
  - ◊ This figure shows the occupancy profile derived for the case of a one-dimensional ideal range sensor, characterized by:

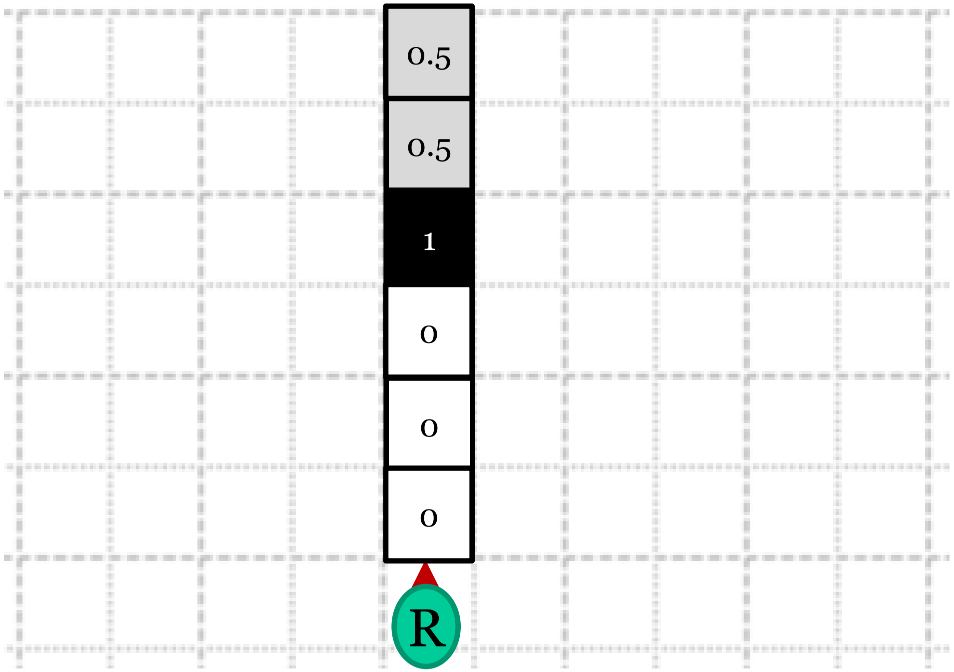
$$p(z\,|\,r)=\delta(z-r)$$

where δ is Dirac delta function



# Occupancy Grid Framework

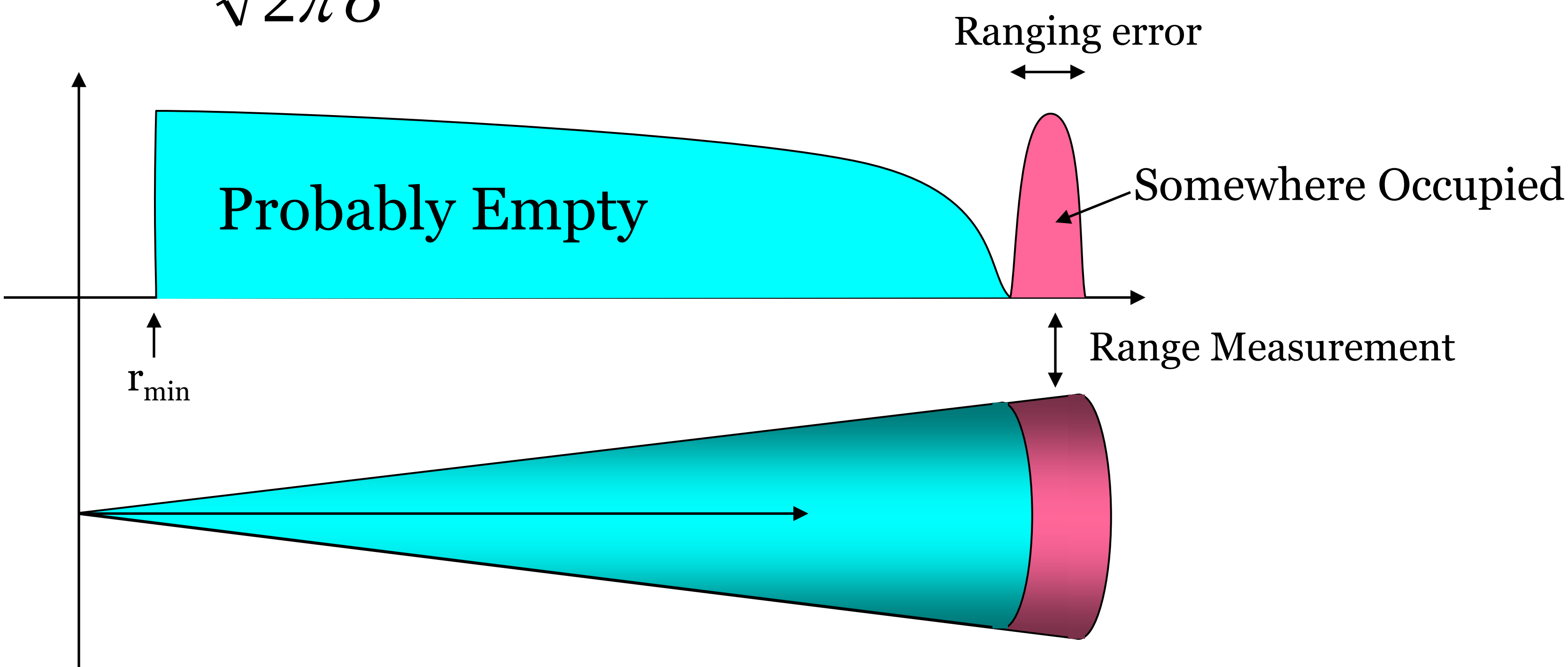
- **Bayesian Estimation Process:**
  - ◇ Given a **range reading z**, the corresponding cell has occupancy **probability 1**.
  - ◇ The **preceding cells** are **empty** and have occupancy **probability 0**.
  - ◇ The **succeeding cells** have not been observed and are therefore **unknown**, so the occupancy probability is **0.5**.



# Occupancy Grid Framework

- **Bayesian Estimation Process:**
  - ◇ For **One-dimensional Gaussian range sensor**:

$$p(z \mid r) = \frac{2}{\sqrt{2\pi\sigma}} \exp^{\left(\frac{-(z-r)^2}{2\sigma^2}\right)}$$



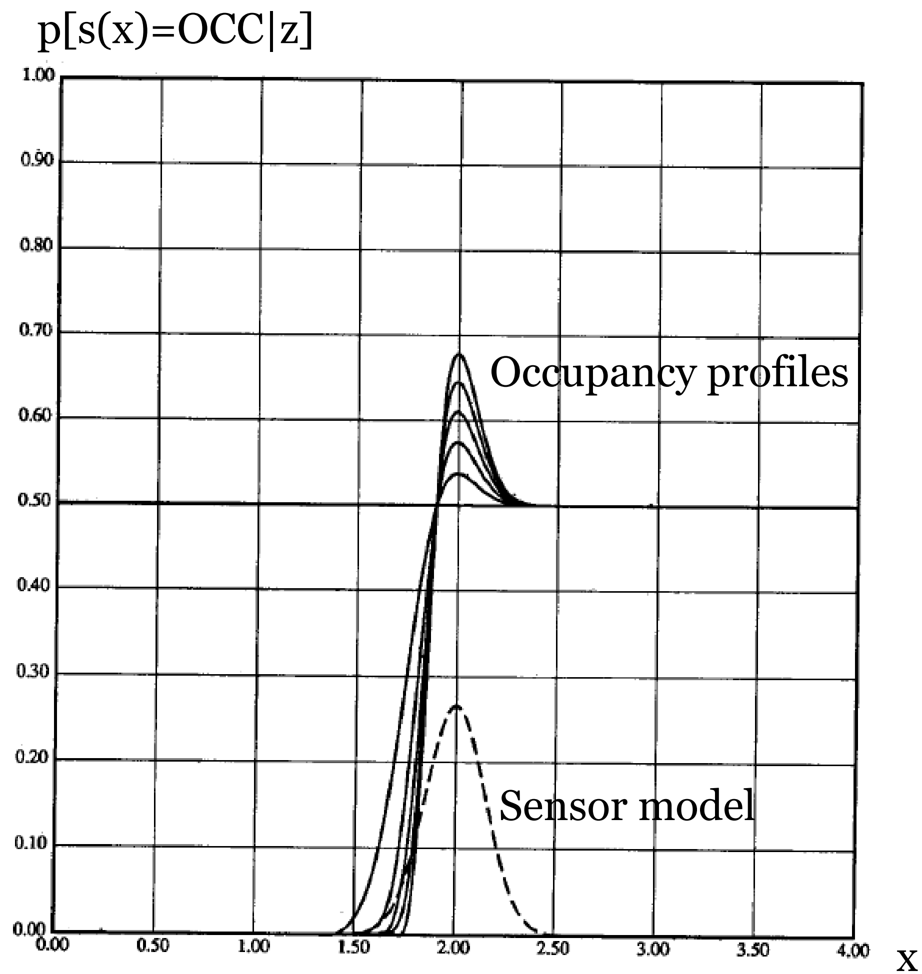

Gaussian Uncertainty in Range

# Occupancy Grid Framework

- **Bayesian Estimation Process:**
  - ◇ For **One-dimensional Gaussian range sensor**:

$$p(z \mid r) = \frac{2}{\sqrt{2\pi\sigma}} \exp^{\left(\frac{-(z-r)^2}{2\sigma^2}\right)}$$

  - ◇ Sensor positioned at x=0.0 and z=2.0. The grid was initialized to p[s(x)=OCC]=0.5.
  - ◇ Here the occupancy grid **converges** towards the behavior of the ideal sensor.

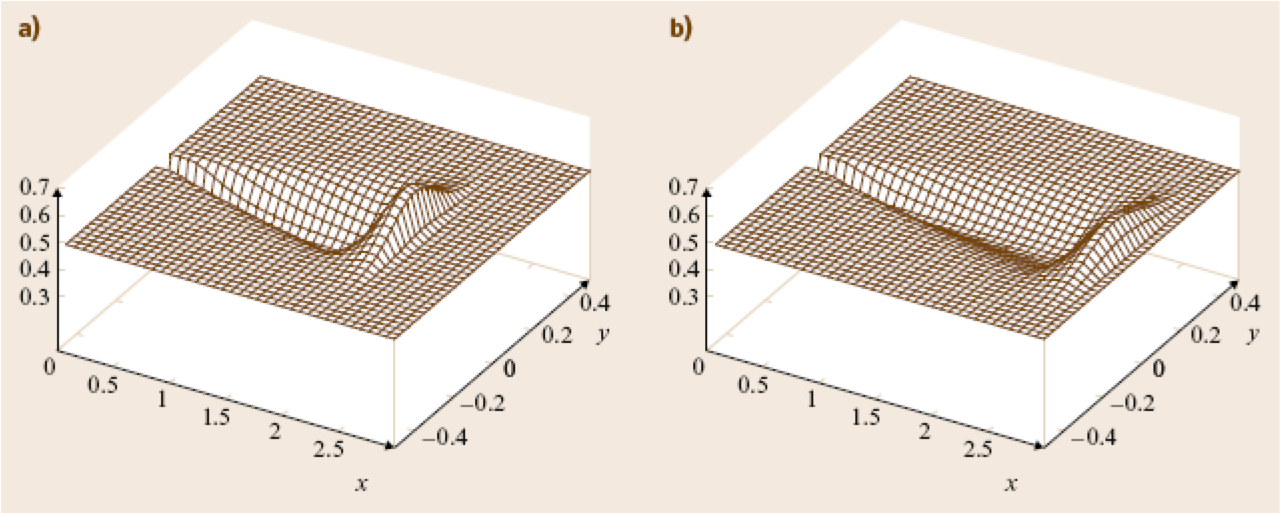


# Occupancy Grid Framework

- **Bayesian Estimation Process:**
  - ◇ For **Two-dimensional Gaussian range sensor**:

$$p(z \mid r,\theta) = \frac{1}{2\pi\sigma_r\sigma_\theta} \exp\left(\frac{-1}{2}\cdot\left(\frac{(z-r)^2}{\sigma_r^2} + \frac{\theta^2}{\sigma_\theta^2}\right)\right)$$

  - ◇ Sonar sensor is modeled with Gaussian uncertainty in both range and angle.




Occupancy probability introduced by a single ultrasound measurement of
(a) z = 2.0m and
(b) z = 2.5m

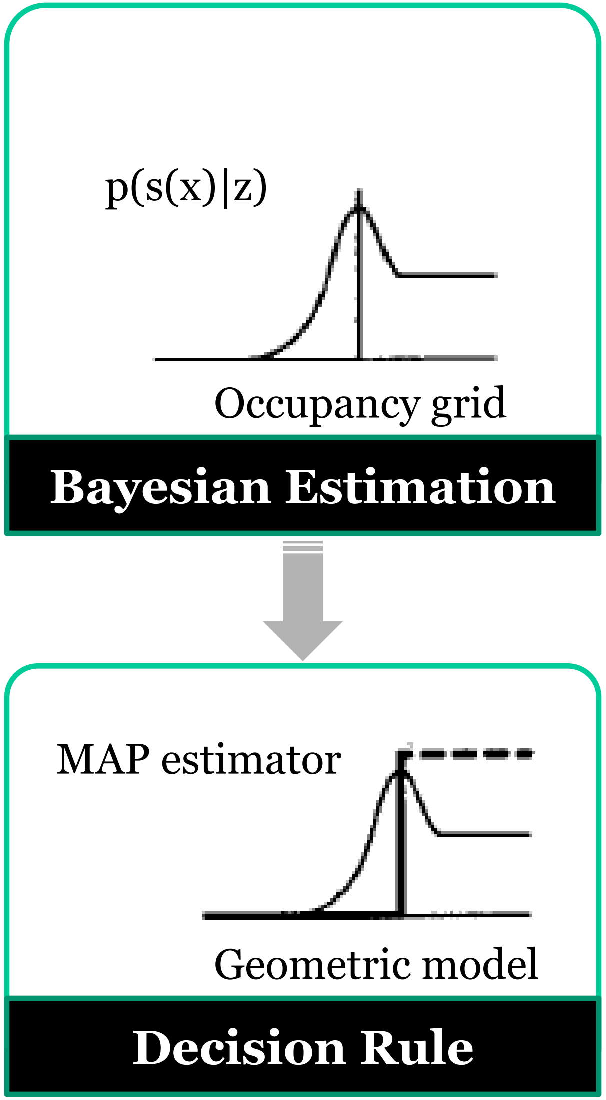
# Occupancy Grid Framework

- **Decision Making:**
  - ◇ For certain application, it may be necessary to assign a specific states to the cells of the occupancy grid.
  - ◇ An optimal estimate of the state of a cell is given by the **Maximum A Posteriori (MAP)** decision rule:

If $p[s(C) = OCC] > p[s(C) = EMP]$ Cell is occupied

If $p[s(C) = OCC] < p[s(C) = EMP]$ Cell is empty

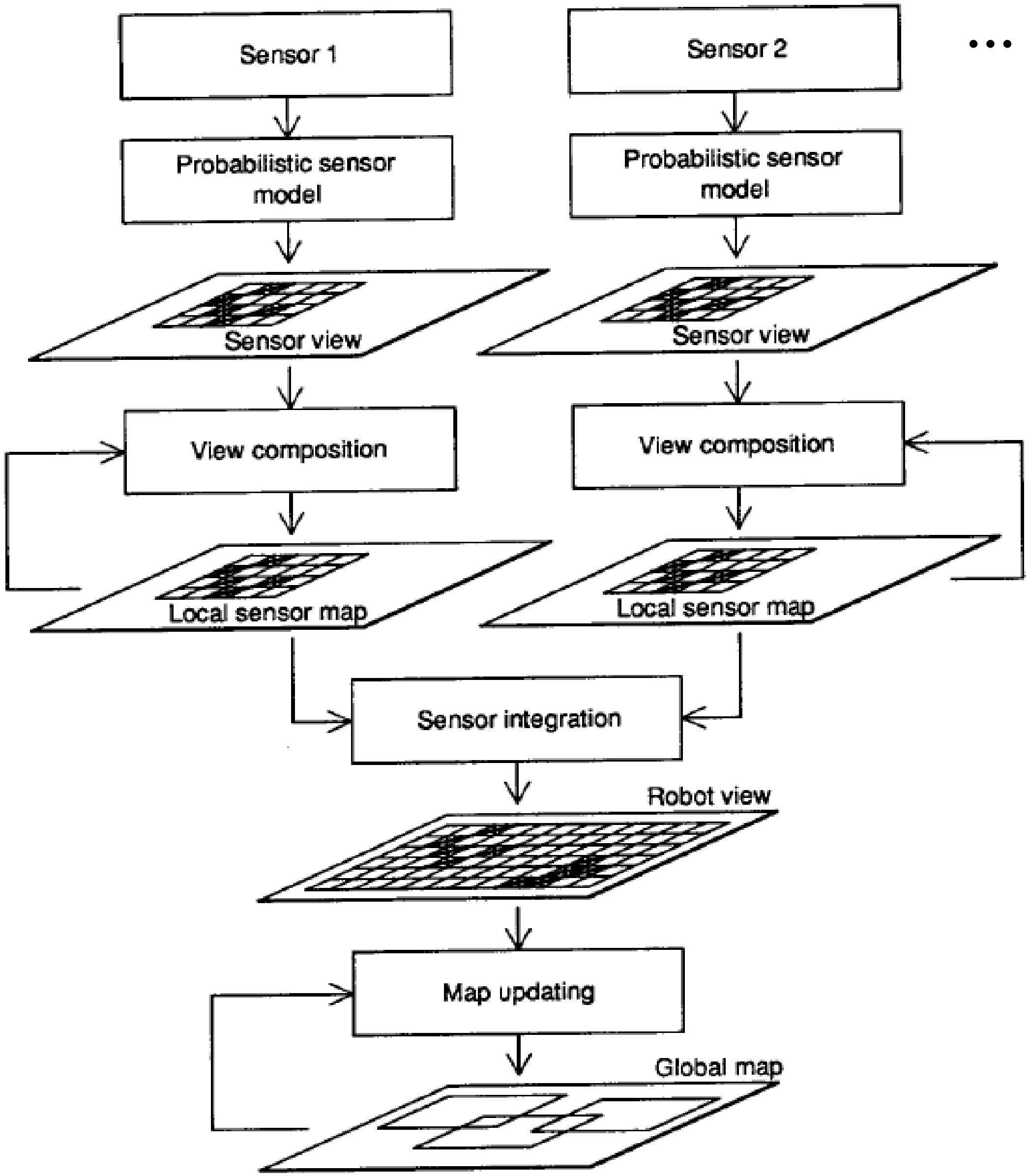
If $p[s(C) = OCC] = p[s(C) = EMP]$ Cell is unknown



# Outline

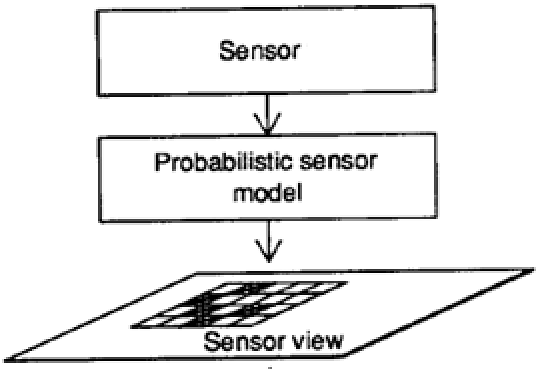
- Markov Localization
- World Modeling
- Metric Mapping
- Occupancy Grid Framework
- **Occupancy Grid-based Mapping**
- Case Study
- Summary

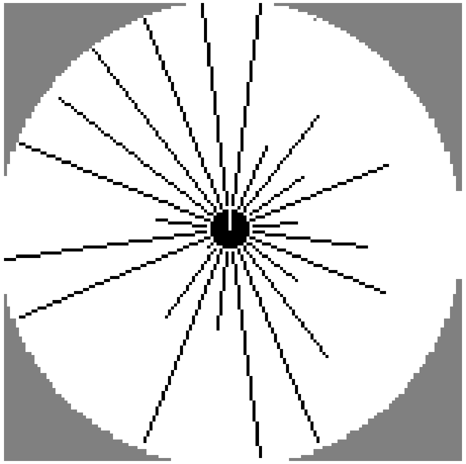
# Occupancy Grid-based Mapping



# Occupancy Grid-based Mapping

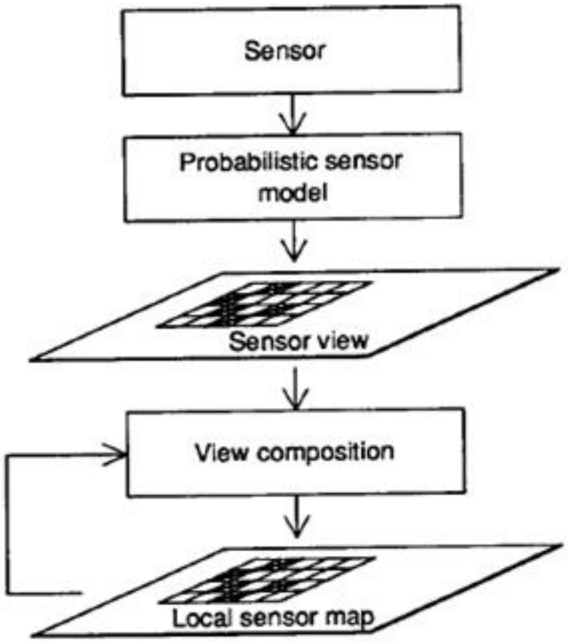
- **Data Acquisition:** The vehicle explores and maps its environment, acquiring information about the world.
- **Sensor View:** The data acquired from a single sensor reading is called a sensor view.



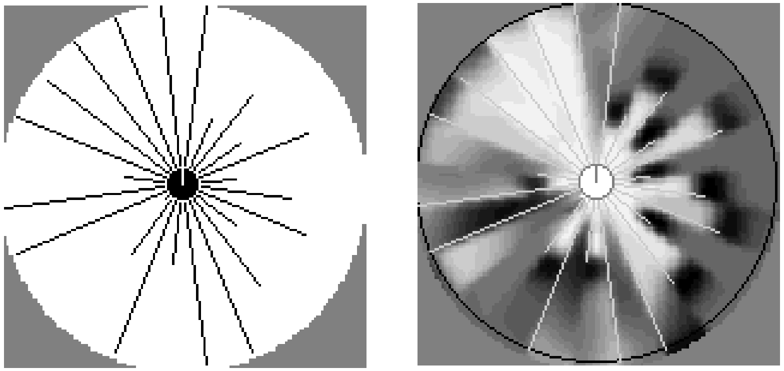



# Occupancy Grid-based Mapping

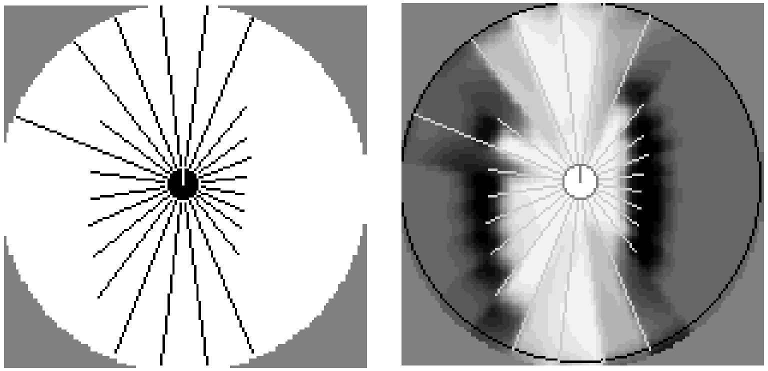
- **Local Sensor Map Building:** Various sensor views taken from a single robot position can be composed into a local sensor map.




Ex-1

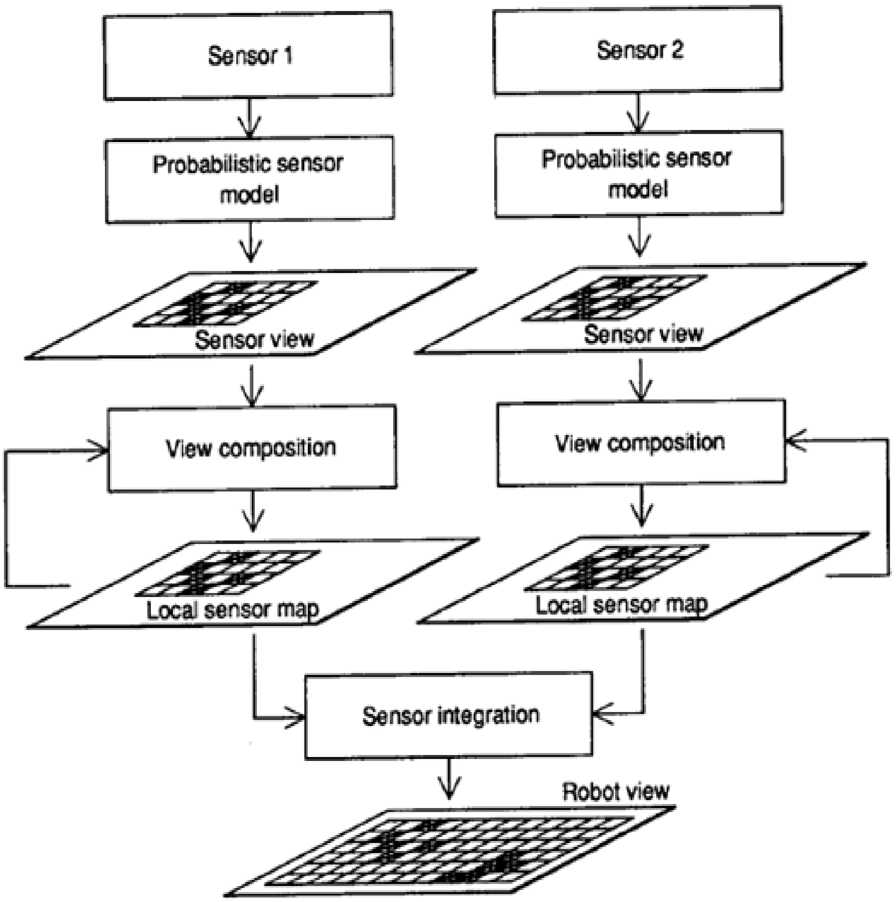


Ex-2



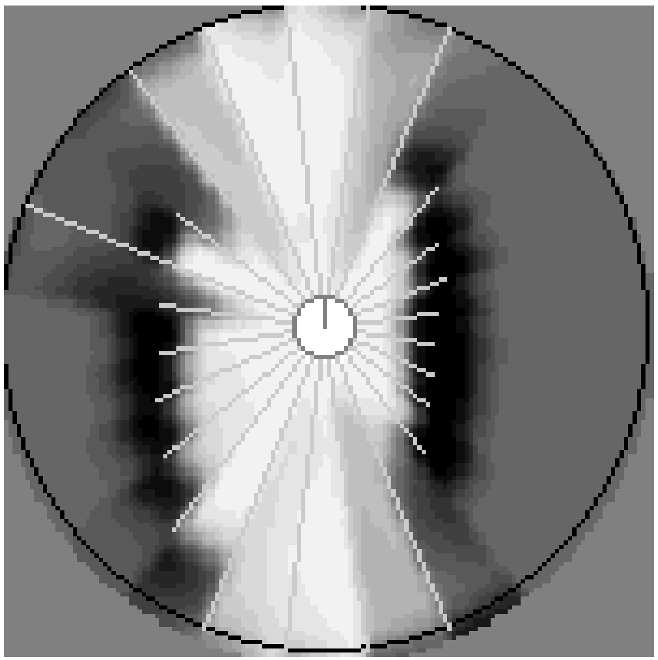
# Occupancy Grid-based Mapping

- **Robot View:** Multiple sensor maps can be maintained separately for different sensors. To obtain an integrated description of the robot's surroundings, sensor fusion of the separate local sensor maps is performed to yield a robot view, which encapsulates the total sensor information recovered from a single sensing position.

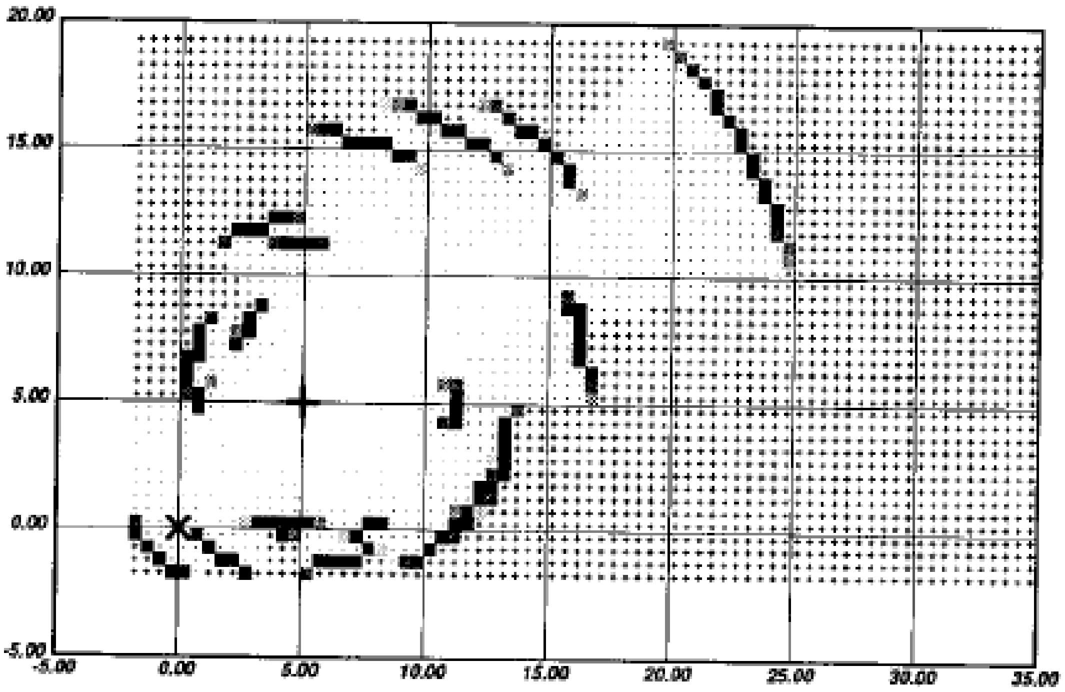


# Occupancy Grid-based Mapping

- **Robot View:**



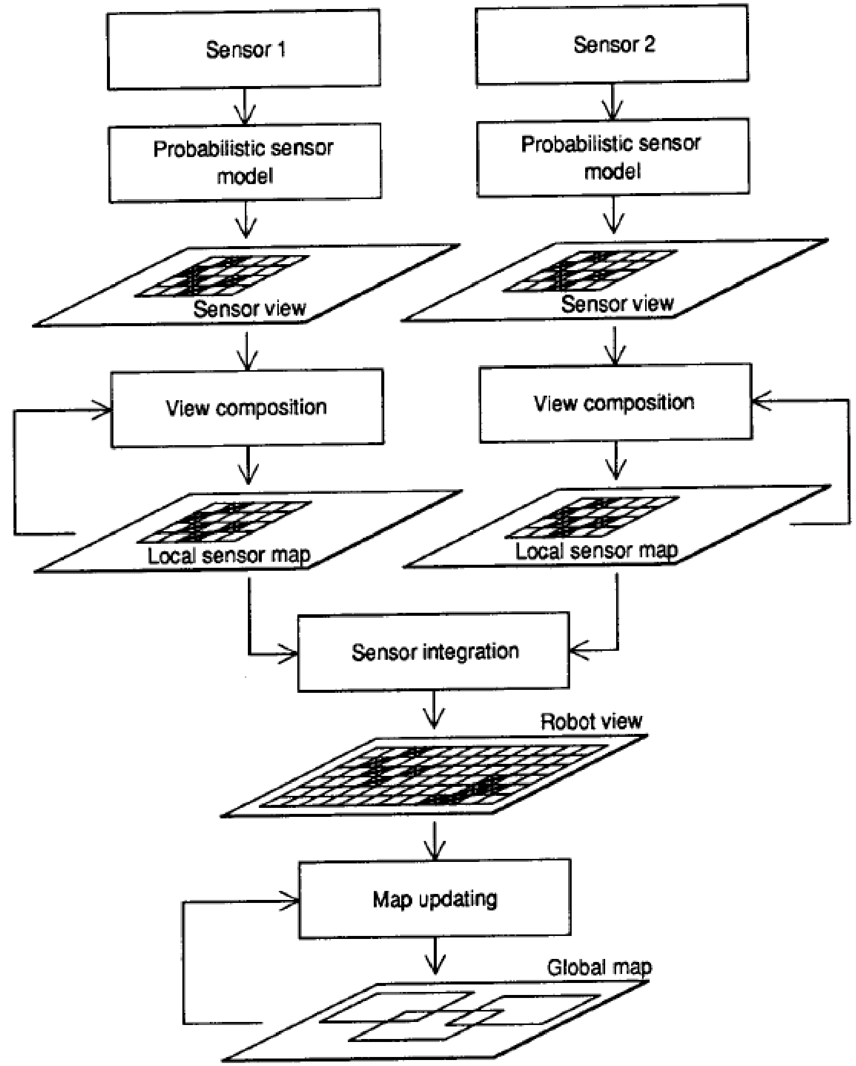
Local Sensor Map




Robot View

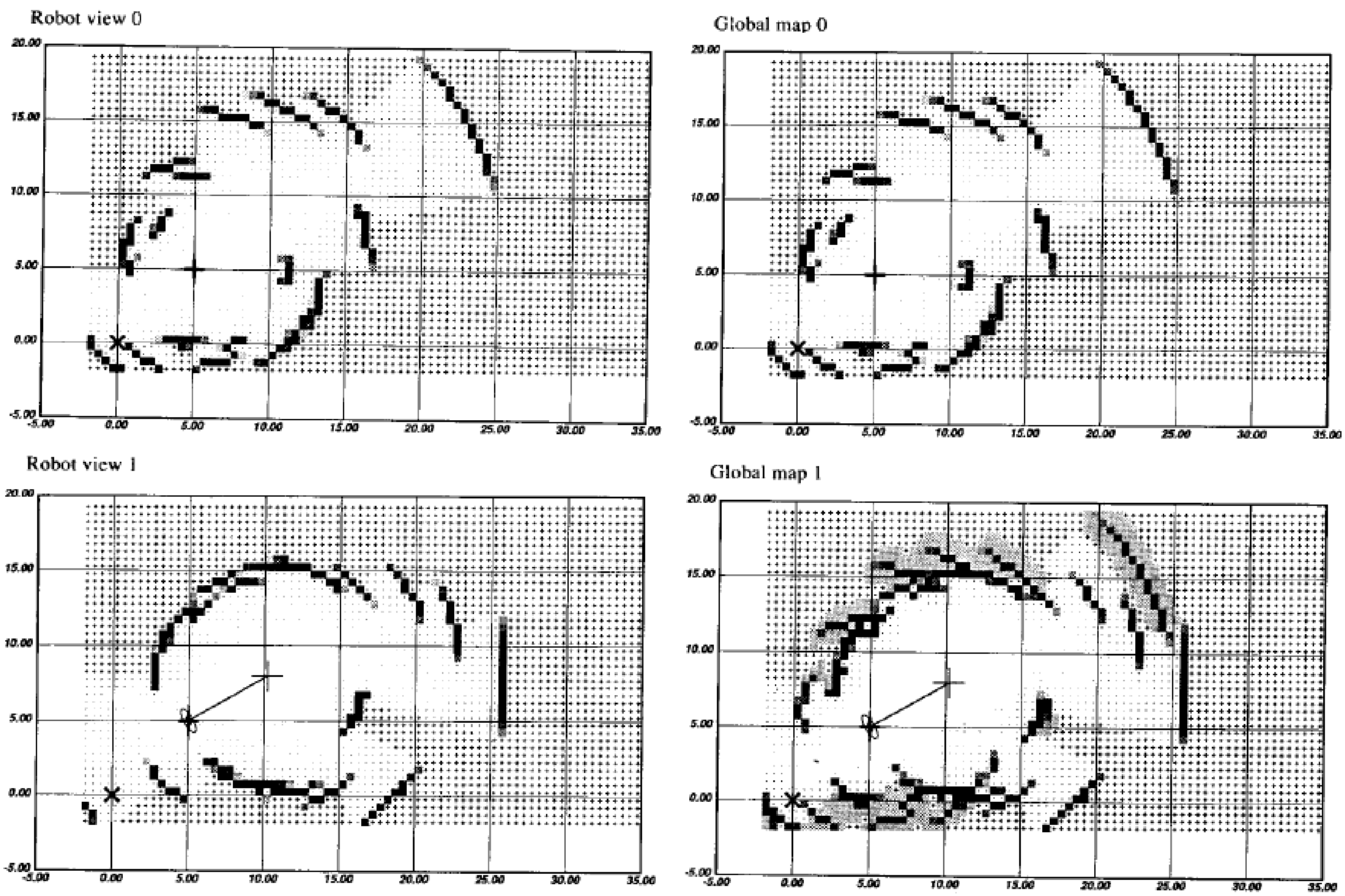
# Occupancy Grid-based Mapping

- **Global View:** As the vehicle travels through its terrain of operation, robot views taken from multiple data-gathering locations are composed into a global map of the environment. This requires the registration of the robot views to a common frame of reference.



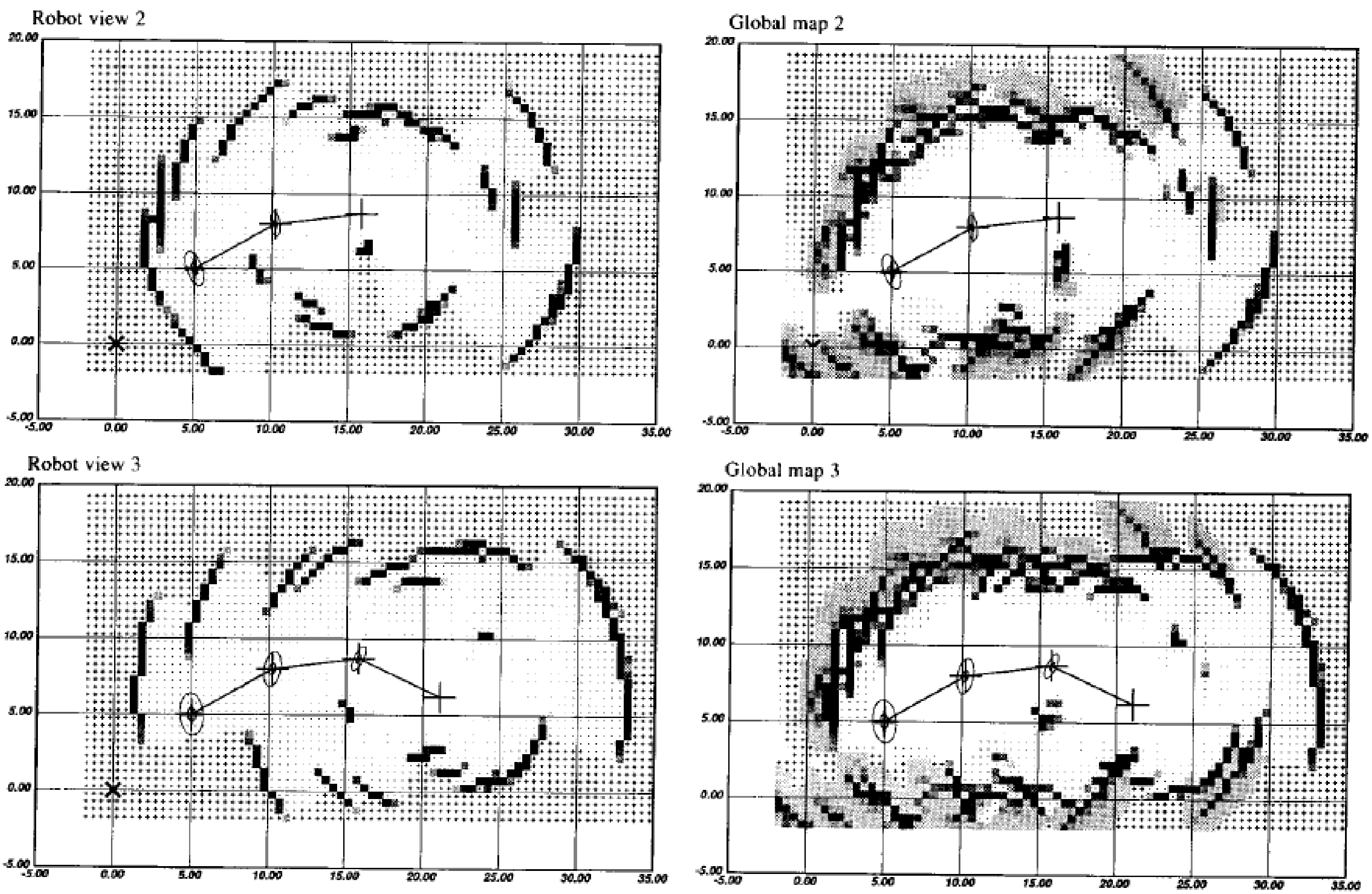
# Occupancy Grid-based Mapping

- **Global View:**



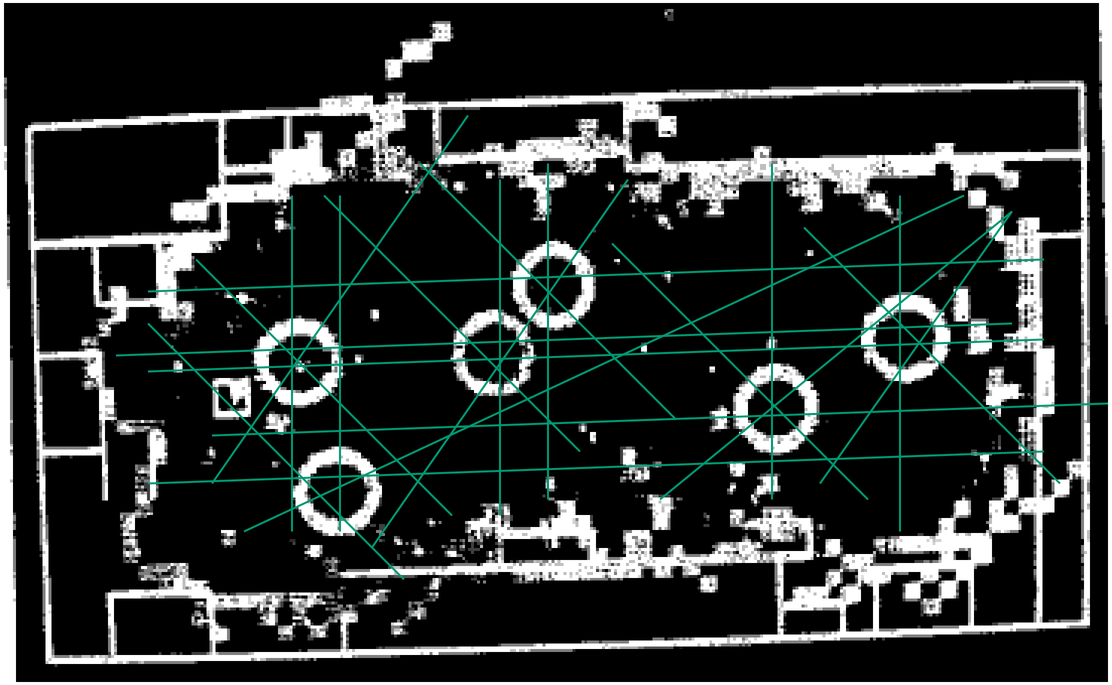
# Occupancy Grid-based Mapping

- **Global View:**



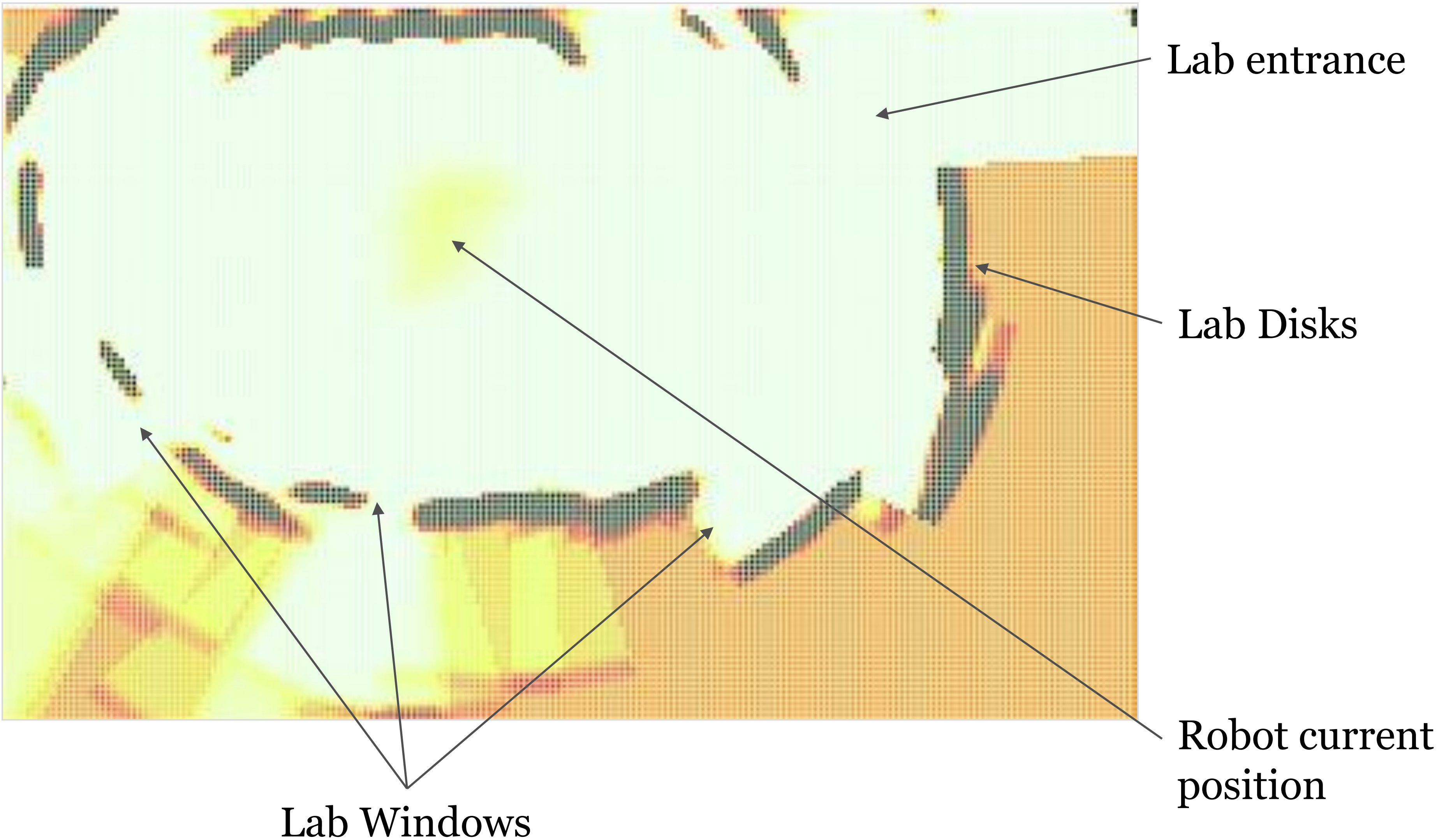
# Occupancy Grid-based Mapping

- 2-D sonar map produced by occupancy grid framework. Circles represent positions of the sonar ring where sensor readings were taken.



- White squares are areas where objects have been sensed, with a grey level proportional to the probability of occupancy. Thick while lines are the boundaries in a pre-recorded map, and thin white lines are the edges of the sonar beams.

# Occupancy Grid-based Mapping



# Outline

- Markov Localization
- World Modeling
- Metric Mapping
- Occupancy Grid Framework
- Occupancy Grid-based Mapping
- **Case Study**
- Summary

# Case Study

- A mobile robot equipped with 4 idealized sonar sensor is navigating in unknown environment as shown below.


- Show how the robot can build a map for this environment using 3 ultrasound scans in three different positions.
- **Assumptions:** static environment and robot position is known

# Case Study

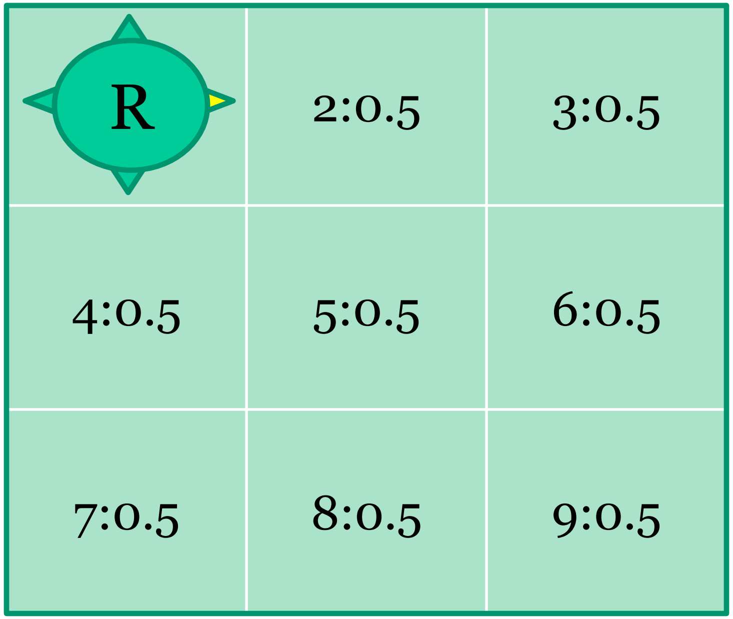
- **Initialization**
  - ◇ Divide the space into 9 evenly size cells as shown below.
  - ◇ Assign 0.5 (unknown) as a prior belief or occupancy probability for the state of all the cells:

| | | |
|---|---|---|
| 1:0.5 | 2:0.5 | 3:0.5 |
| 4:0.5 | 5:0.5 | 6:0.5 |
| 7:0.5 | 8:0.5 | 9:0.5 |

$$p[s(C_i) = OCC \mid \{z\}_0] = 0.5$$

# Case Study

- **First Position: Sensor Views**
  - ◇ Front Sensor View: $z_{F1}$={2 units} ⇒ sensor can see cell $C_2$ & $C_3$
  - ◇ Right Sensor View: $z_{R1}$={2 units} ⇒ sensor can see cell $C_4$ & $C_7$
  - ◇ Left Sensor View: $z_{L1}$={0 units}
  - ◇ Back Sensor View: $z_{B1}$={0 units}



# Case Study

- **First Position: Front Sensor Local Map**
  - ◇ Front Sensor View: $z_{F1}$={2 units}
  - ◇ First Cell $C_i$=1

| | | |
|---|---|---|
| 1:1 | 2:0.5 | 3:0.5 |

$$p[s(C_1)=OCC\,|\,\{z_F\}_1]=\frac{p[z_{F1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_F\}_0]}{\sum_{s(C_1)} p[z_{F1}\,|\,s(C_1)].p[s(C_1)\,|\,\{z_F\}_0]}$$

$$p[s(C_1)=OCC\,|\,\{z_F\}_1]=\frac{p[z_{F1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_F\}_0]}{p[z_{F1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_F\}_0]+p[z_{F1}\,|\,s(C_1)=EMP].p[s(C_1)=EMP\,|\,\{z_F\}_0]}$$

$$p[s(C_1)=OCC\,|\,\{z_F\}_1]=\frac{1\times 0.5}{1\times 0.5+0\times 0.5}=1$$

# Case Study

- **First Position: Front Sensor Local Map**
  - ◇ Front Sensor View: $z_{F1}$={2 units}
  - ◇ Second Cell $C_i$=2

| | | |
|---|---|---|
| 1:1 | 2:0 | 3:0.5 |

$$p[s(C_2)=OCC\,|\,\{z_F\}_1]=\frac{p[z_{F1}\,|\,s(C_2)=OCC].p[s(C_2)=OCC\,|\,\{z_F\}_0]}{\sum_{s(C_2)} p[z_{F1}\,|\,s(C_2)].p[s(C_2)\,|\,\{z_F\}_0]}$$

$$p[s(C_2)=OCC\,|\,\{z_F\}_1]=\frac{p[z_{F1}\,|\,s(C_2)=OCC].p[s(C_2)=OCC\,|\,\{z_F\}_0]}{p[z_{F1}\,|\,s(C_2)=OCC].p[s(C_2)=OCC\,|\,\{z_F\}_0]+p[z_{F1}\,|\,s(C_2)=EMP].p[s(C_2)=EMP\,|\,\{z_F\}_0]}$$

$$p[s(C_2)=OCC\,|\,\{z_F\}_1]=\frac{0\times 0.5}{0\times 0.5+1\times 0.5}=0$$

# Case Study

- **First Position: Front Sensor Local Map**
  - ◇ Front Sensor View: $z_{F1}$={2 units}
  - ◇ Third Cell $C_i$=3

| | | |
|---|---|---|
| 1:1 | 2:0 | 3:0 |

Front Sensor Local Map

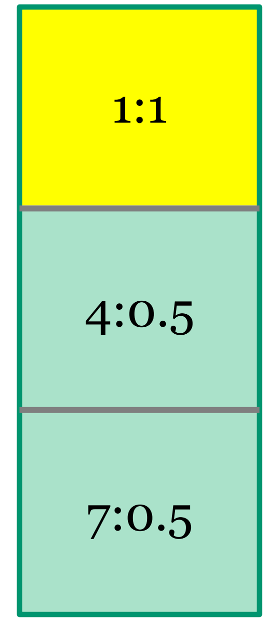
$$p[s(C_3)=OCC\,|\,\{z_F\}_1]=\frac{p[z_{F1}\,|\,s(C_3)=OCC].p[s(C_3)=OCC\,|\,\{z_F\}_0]}{\sum_{s(C_3)} p[z_{F1}\,|\,s(C_3)].p[s(C_3)\,|\,\{z_F\}_0]}$$

$$p[s(C_3)=OCC\,|\,\{z_F\}_1]=\frac{p[z_{F1}\,|\,s(C_3)=OCC].p[s(C_3)=OCC\,|\,\{z_F\}_0]}{p[z_{F1}\,|\,s(C_3)=OCC].p[s(C_3)=OCC\,|\,\{z_F\}_0]+p[z_{F1}\,|\,s(C_3)=EMP].p[s(C_3)=EMP\,|\,\{z_F\}_0]}$$

$$p[s(C_3)=OCC\,|\,\{z_F\}_1]=\frac{0\times 0.5}{0\times 0.5+1\times 0.5}=0$$

# Case Study

- **First Position: Right Sensor Local Map**
  - ◇ Right Sensor View: $z_{R1}$={2 units}
  - ◇ First Cell $C_i$=1



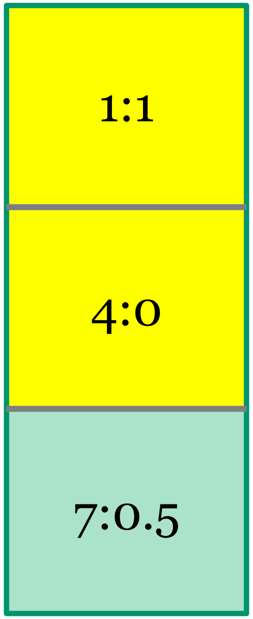

$$p[s(C_1) = OCC \mid \{z_R\}_1] = \frac{p[z_{R1} \mid s(C_1) = OCC].p[s(C_1) = OCC \mid \{z_R\}_0]}{\sum_{s(C_1)} p[z_{R1} \mid s(C_1)].p[s(C_1) \mid \{z_R\}_0]}$$

$$p[s(C_1) = OCC \mid \{z_R\}_1] = \frac{p[z_{R1} \mid s(C_1) = OCC].p[s(C_1) = OCC \mid \{z_R\}_0]}{p[z_{R1} \mid s(C_1) = OCC].p[s(C_1) = OCC \mid \{z_R\}_0] + p[z_{R1} \mid s(C_1) = EMP].p[s(C_1) = EMP \mid \{z_R\}_0]}$$

$$p[s(C_1) = OCC \mid \{z_R\}_1] = \frac{1 \times 0.5}{1 \times 0.5 + 0 \times 0.5} = 1$$

# Case Study

- **First Position: Right Sensor Local Map**
  - ◇ Right Sensor View: $z_{R1}$={2 units}
  - ◇ Second Cell $C_i$=2




$$p[s(C_2)=OCC\,|\,\{z_R\}_1]=\frac{p[z_{R1}\,|\,s(C_2)=OCC].p[s(C_2)=OCC\,|\,\{z_R\}_0]}{\sum\limits_{s(C_2)} p[z_{R1}\,|\,s(C_2)].p[s(C_2)\,|\,\{z_R\}_0]}$$

$$p[s(C_2)=OCC\,|\,\{z_R\}_1]=\frac{p[z_{R1}\,|\,s(C_2)=OCC].p[s(C_2)=OCC\,|\,\{z_R\}_0]}{p[z_{R1}\,|\,s(C_2)=OCC].p[s(C_2)=OCC\,|\,\{z_R\}_0]+p[z_{R1}\,|\,s(C_2)=EMP].p[s(C_2)=EMP\,|\,\{z_R\}_0]}$$

$$p[s(C_2)=OCC\,|\,\{z_R\}_1]=\frac{0\times 0.5}{0\times 0.5+1\times 0.5}=0$$

# Case Study

- **First Position: Right Sensor Local Map**
  - ◇ Right Sensor View: $z_{R1}$={2 units}
  - ◇ Third Cell $C_i$=3

$$p[s(C_3)=OCC\,|\,\{z_R\}_1]=\frac{p[z_{R1}\,|\,s(C_3)=OCC].p[s(C_3)=OCC\,|\,\{z_R\}_0]}{\sum_{s(C_3)} p[z_{R1}\,|\,s(C_3)].p[s(C_3)\,|\,\{z_R\}_0]}$$

Right Sensor Local Map

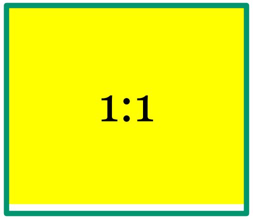
| |
|---|
| 1:1 |
| 4:0 |
| 7:0 |

$$p[s(C_3)=OCC\,|\,\{z_R\}_1]=\frac{p[z_{R1}\,|\,s(C_3)=OCC].p[s(C_3)=OCC\,|\,\{z_R\}_0]}{p[z_{R1}\,|\,s(C_3)=OCC].p[s(C_3)=OCC\,|\,\{z_R\}_0]+p[z_{R1}\,|\,s(C_3)=EMP].p[s(C_3)=EMP\,|\,\{z_R\}_0]}$$

$$p[s(C_3)=OCC\,|\,\{z_R\}_1]=\frac{0\times 0.5}{0\times 0.5+1\times 0.5}=0$$

# Case Study

- **First Position: Left Sensor Local Map**
  - ◇ Left Sensor View: $z_{L1}$={0 units}
  - ◇ First Cell $C_i$=1




Left Sensor Local Map

$$p[s(C_1)=OCC\,|\,\{z_L\}_1]=\frac{p[z_{L1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_L\}_0]}{\sum_{s(C_1)} p[z_{L1}\,|\,s(C_1)].p[s(C_1)\,|\,\{z_L\}_0]}$$

$$p[s(C_1)=OCC\,|\,\{z_L\}_1]=\frac{p[z_{L1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_L\}_0]}{p[z_{L1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_L\}_0]+p[z_{L1}\,|\,s(C_1)=EMP].p[s(C_1)=EMP\,|\,\{z_L\}_0]}$$

$$p[s(C_1)=OCC\,|\,\{z_L\}_1]=\frac{1\times 0.5}{1\times 0.5+0\times 0.5}=1$$

# Case Study

- **First Position: Back Sensor Local Map**
  - ◇ Left Sensor View: $z_{B1}$={0 units}
  - ◇ First Cell $C_i$=1


Back Sensor Local Map

$$p[s(C_1)=OCC\,|\,\{z_B\}_1]=\frac{p[z_{B1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_B\}_0]}{\sum_{s(C_1)} p[z_{B1}\,|\,s(C_1)].p[s(C_1)\,|\,\{z_B\}_0]}$$

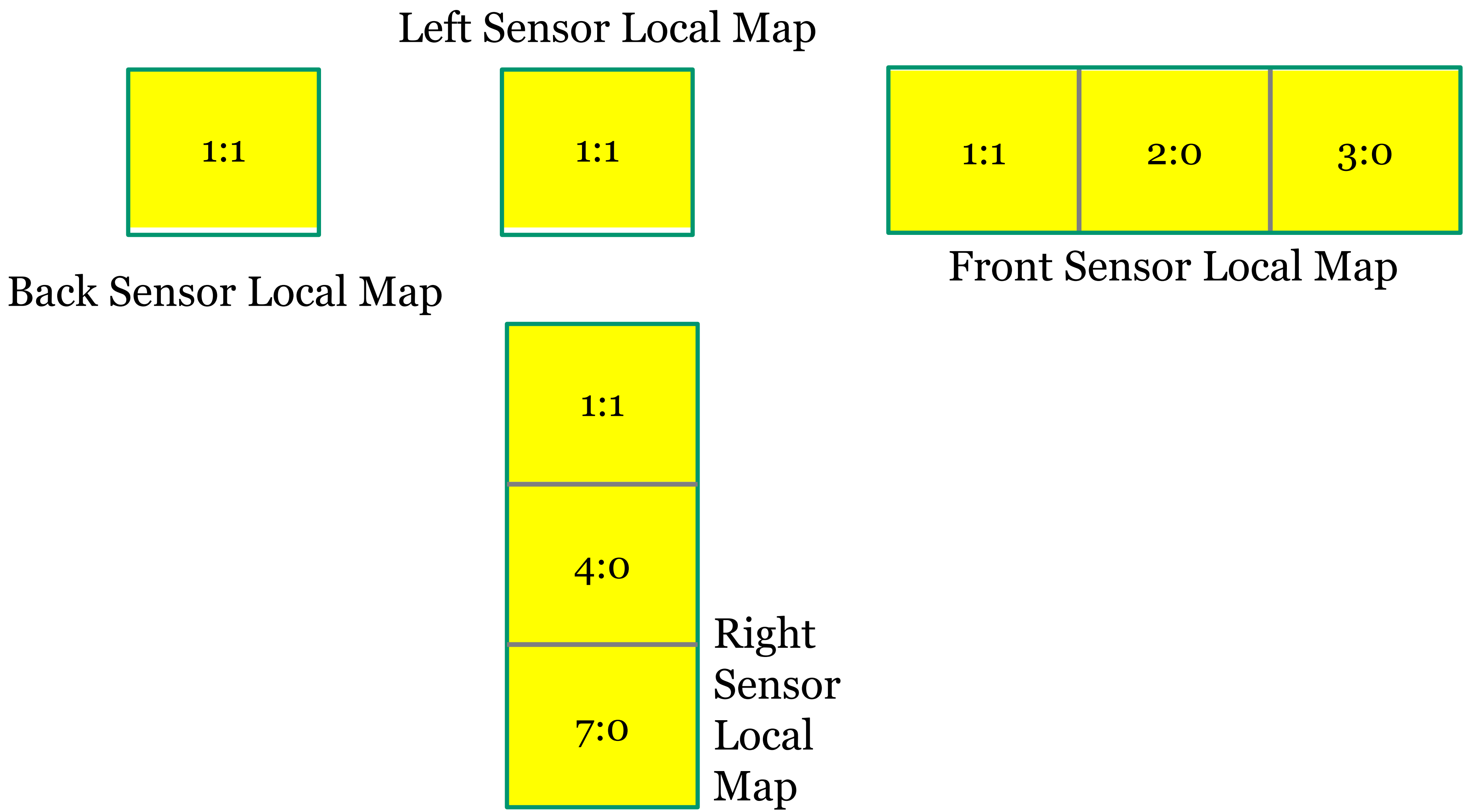
$$p[s(C_1)=OCC\,|\,\{z_B\}_1]=$$

$$\frac{p[z_{B1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_B\}_0]}{p[z_{B1}\,|\,s(C_1)=OCC].p[s(C_1)=OCC\,|\,\{z_B\}_0]+p[z_{B1}\,|\,s(C_1)=EMP].p[s(C_1)=EMP\,|\,\{z_B\}_0]}$$

$$p[s(C_1)=OCC\,|\,\{z_B\}_0]=\frac{1\times 0.5}{1\times 0.5+0\times 0.5}=1$$

# Case Study

- **First Position: Sensor Local Map**



# Case Study

- **First Position: Robot View**
  - ◇ The robot view is obtained by integrating the individual observations (sensor local map) into the map.
  - ◇ For the same cell overlap (cell $C_1$), apply **MAP**.
    - If p[s(C)=OCC]>p[s(C)=EMP]

      Cell is occupied.
    - If p[s(C)=OCC]<p[s(C)= EMP]

      Cell is empty
    - If p[s(C)=OCC]=p[s(C)= EMP]

      Cell is unknown

| | | |
|---|---|---|
| 1:1 | 2:0 | 3:0 |
| 4:0 | 5:0.5 | 6:0.5 |
| 7:0 | 8:0.5 | 9:0.5 |

# Case Study

- **Second and Third Positions:**
  - ◇ Repeat the procedure to obtain other two robot views…

# Case Study

- **Global Map:**
  - ◇ The global map is obtained by integrating the three robot view into the map using MAP.

If $p[s(C)=OCC] > p[s(C)=EMP]$ Cell is occupied

If $p[s(C)=OCC] < p[s(C)=EMP]$ Cell is empty

If $p[s(C)=OCC] = p[s(C)=EMP]$ Cell is unknown

| | | |
|---|---|---|
| 1:0 | 2:0 | 3:0 |
| 4:0 | 5:0 | 6:0 |
| 7:0 | 8:1 | 9:1 |

# Outline

- Markov Localization
- World Modeling
- Metric Mapping
- Occupancy Grid Framework
- Occupancy Grid-based Mapping
- Case Study
- **Summary**

# Summary

- To widen the range of application and deployment of robots, both in research and in industrial contexts, we need to develop more powerful and flexible robotic systems exhibiting higher degrees of autonomy and able to sense, plan, and operate in unstructured environments.
- For that, the robot must be able to interact coherently with its world, both by being able to recover robust and useful spatial descriptions (mapping) of its surroundings using sensory information and by efficiently utilizing these descriptions in appropriate short-term and long-term planning and decision-making activities.
- Mapping is the estimation problem of generating maps from sensor measurements.
- The occupancy grid framework provides a robust and a unified approach to a variety of problems in spatial robot perception and navigation.
- The disadvantage of this framework are its potential discretization errors and high memory requirements.

# References

This lecture is based on materials from the following sources:


- Ch. 2 of Sebastian Thrun, Wolfram Burgard and Dieter Fox. Probabilistic Robotics. MIT Press, 2005.
- A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation", Computer, 22(6):46-57, 1989.
- W. Burgard and M. Hebert. *World Modelling*. Ch.26 in Springer *Handbook of Robotics*, B. Siciliano, O. Khatib (Eds.), 2008.
- W. Burgard. Mapping in Dynamic Environments. Institut für Informatik, Universität Freiburg, Germany.
- C. Yee. *Grid-based Map Building and Navigation Algorithms for Mobile Robots*. M.Sc. Thesis, Sheffield Hallam University, 2008.