Robotics

Estimation and Learning
with Dan Lee

Week 3.
Robotic Mapping

3.1 Introduction to Mapping

Penn Engineering

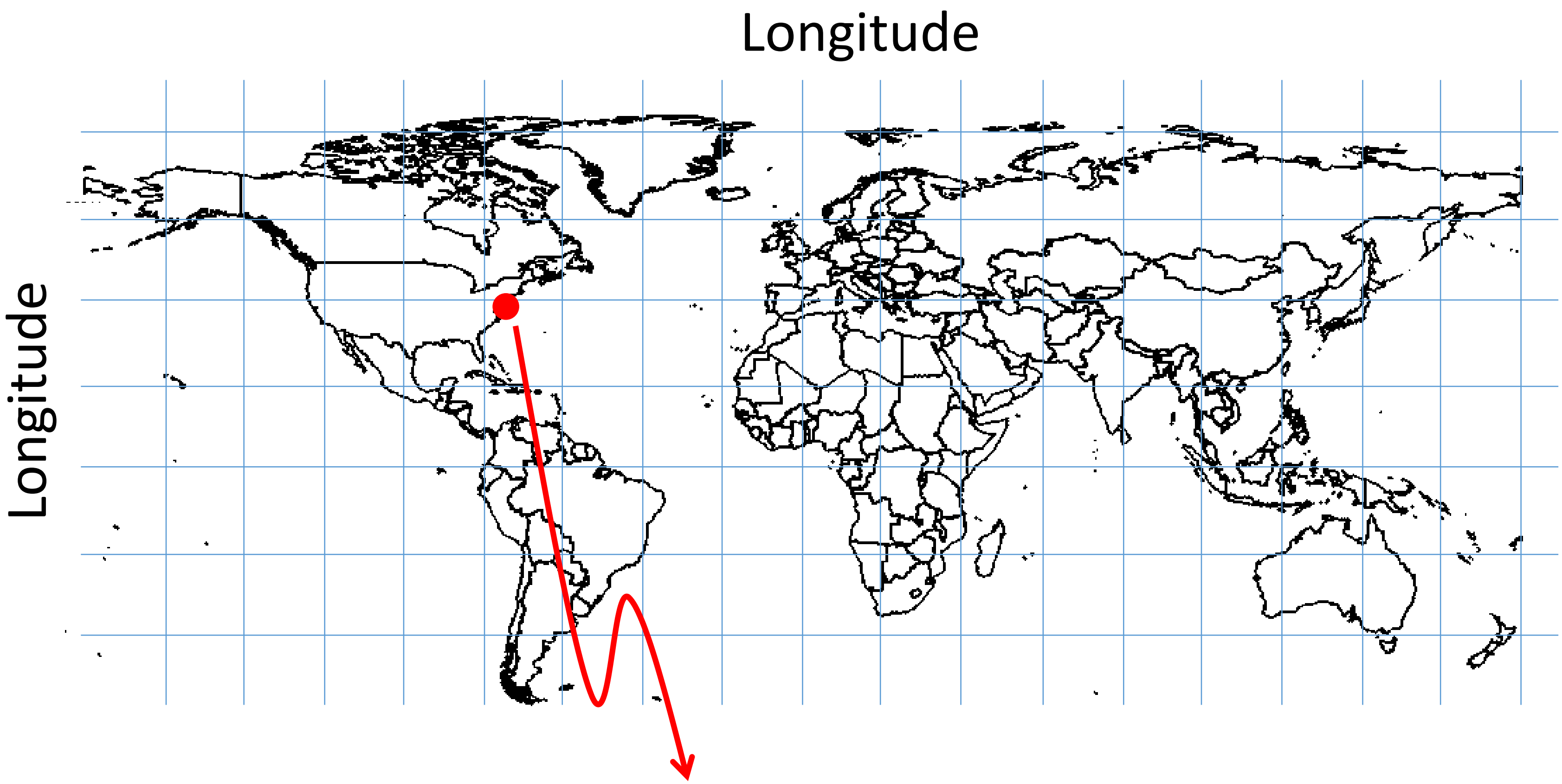
Map and Mapping

- Map is a spatial model of a robot’s environment.
- Mapping is a process for building a map.
- Consideration for mapping
  - Map representation
  - Available sensors
  - Purpose of mapping

Types of Map

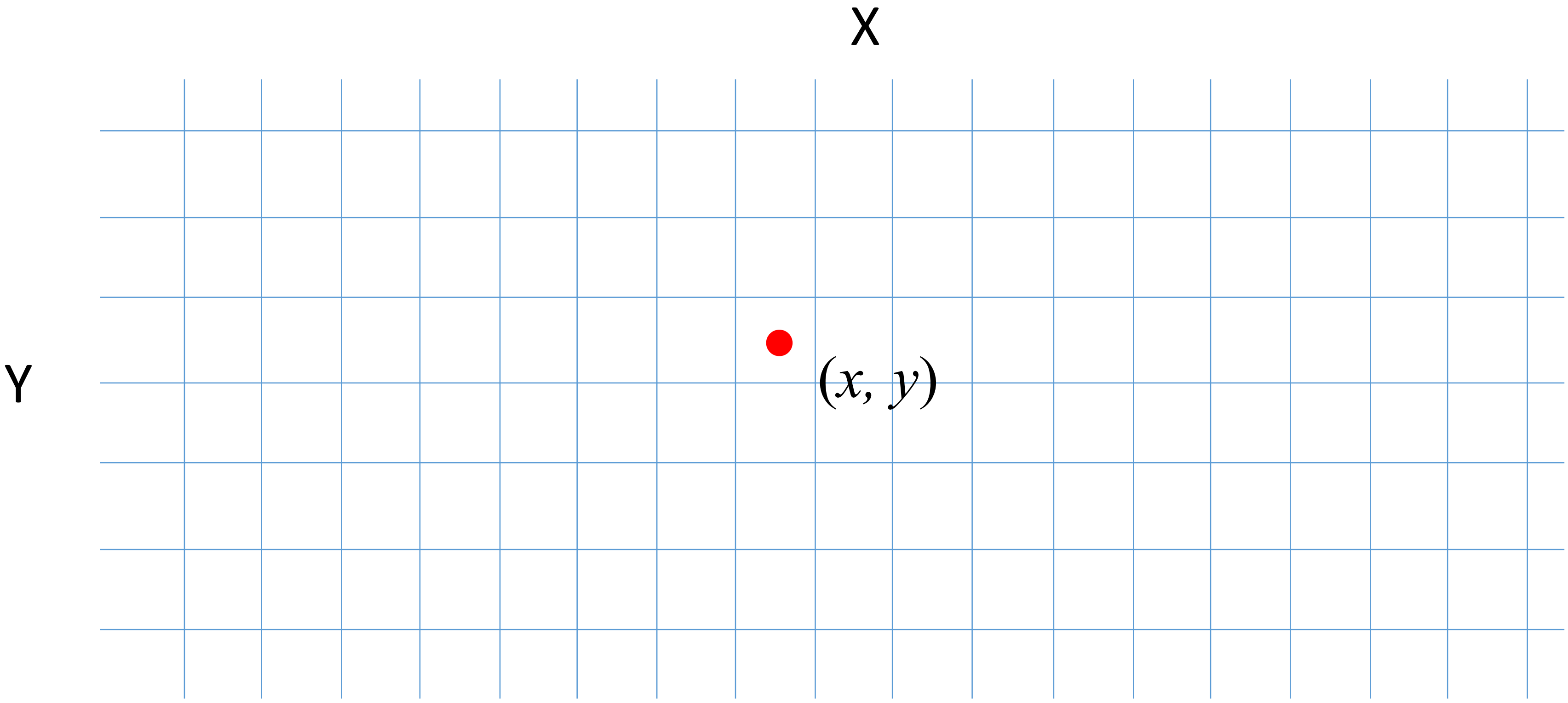
- Metric Map
- Topological Map
- Semantic Map

Types of Map (1) – Metric map



UPenn (39.95°N, 75.19°W)

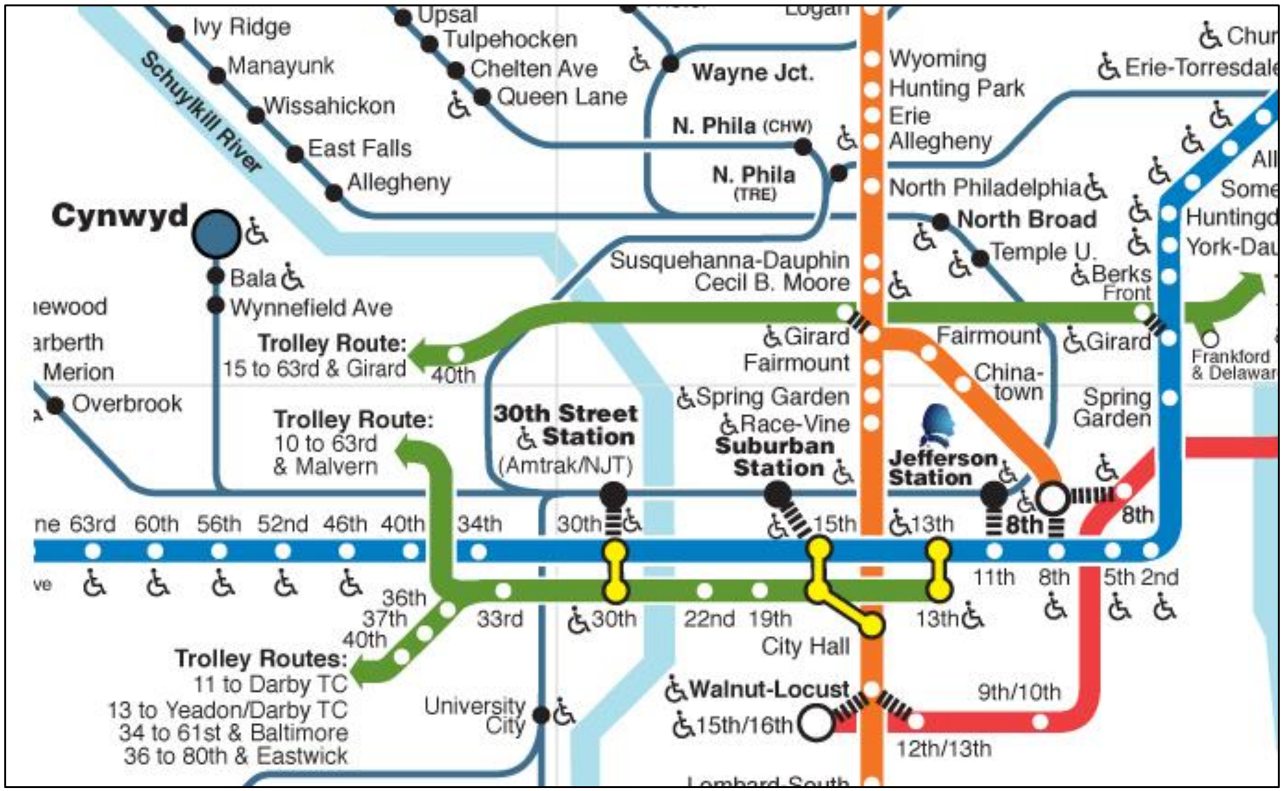
Types of Map (1) – Metric map



A location is represented as a coordinate.

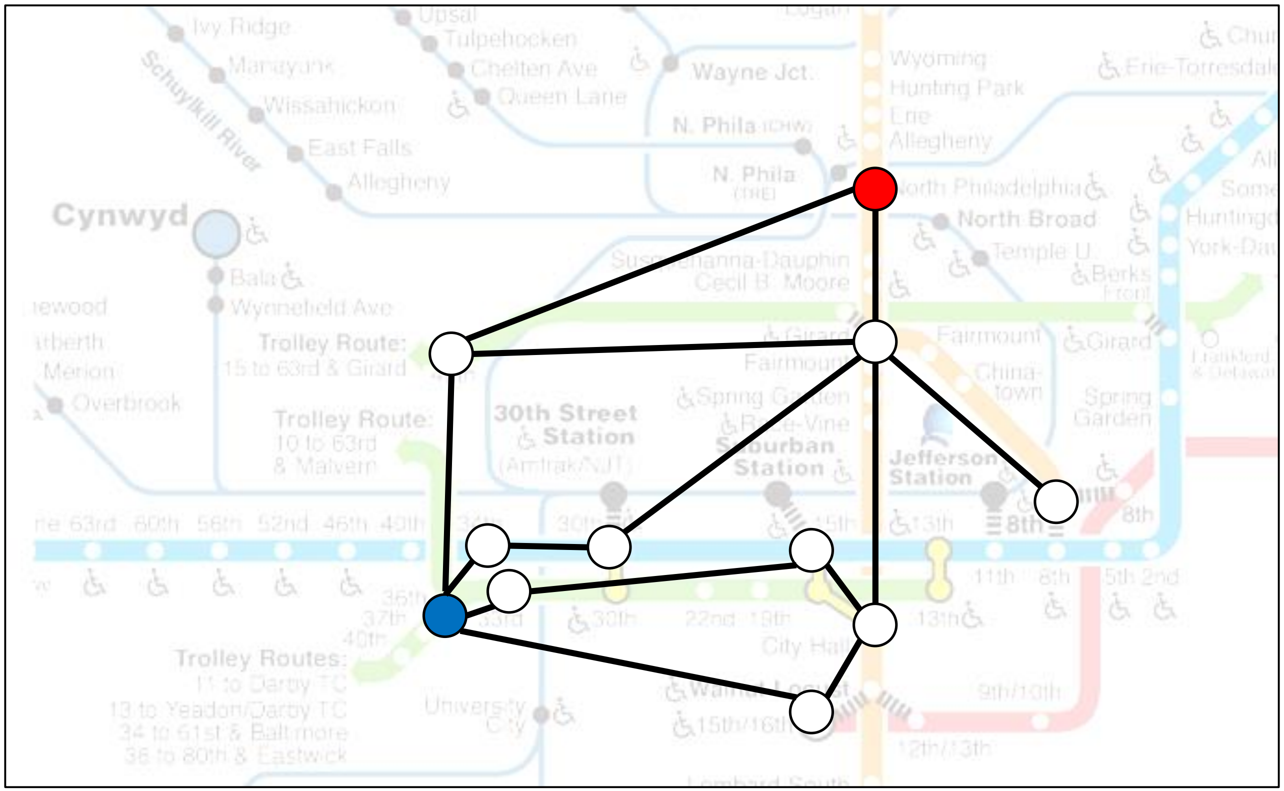
Types of Map (2) – Topological map

Part of SEPTA Train Map



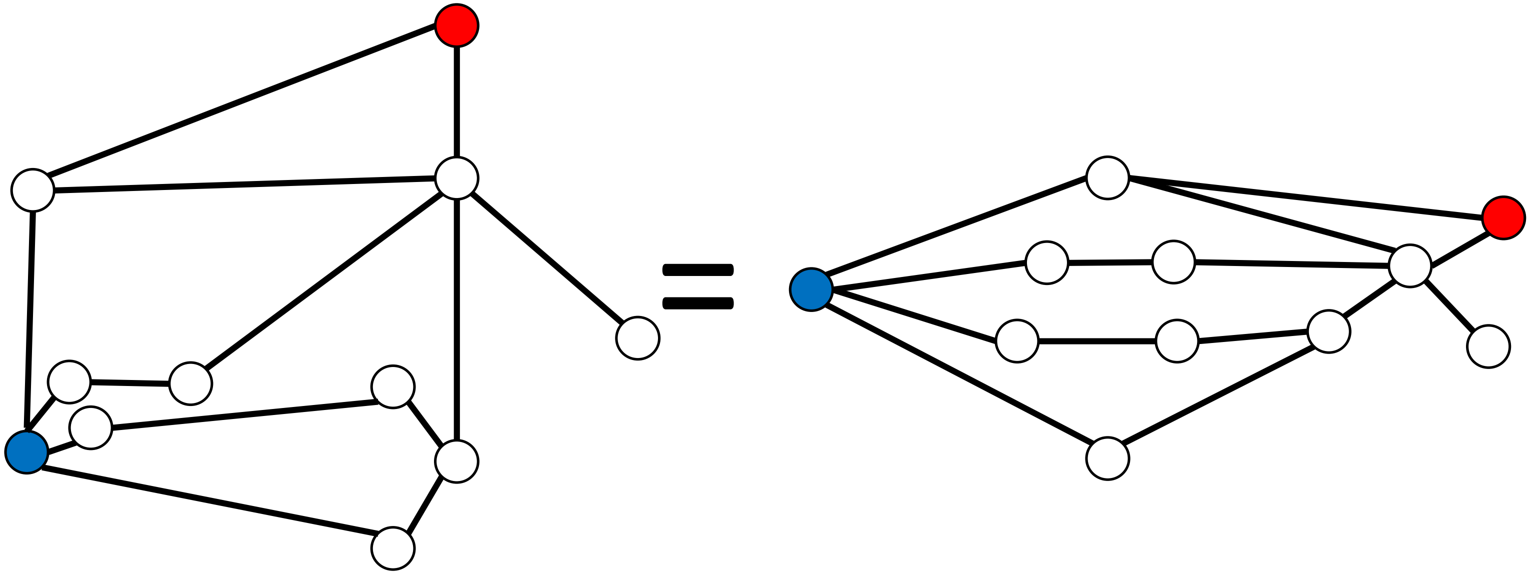
Types of Map (2) – Topological map

Part of SEPTA Train Map



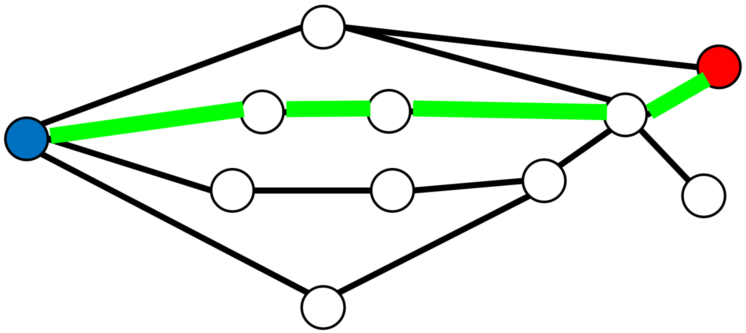
Locations are represented as nodes and their connectivity as arcs.

Types of Map (2) – Topological map



Only the connectivity between nodes matter.

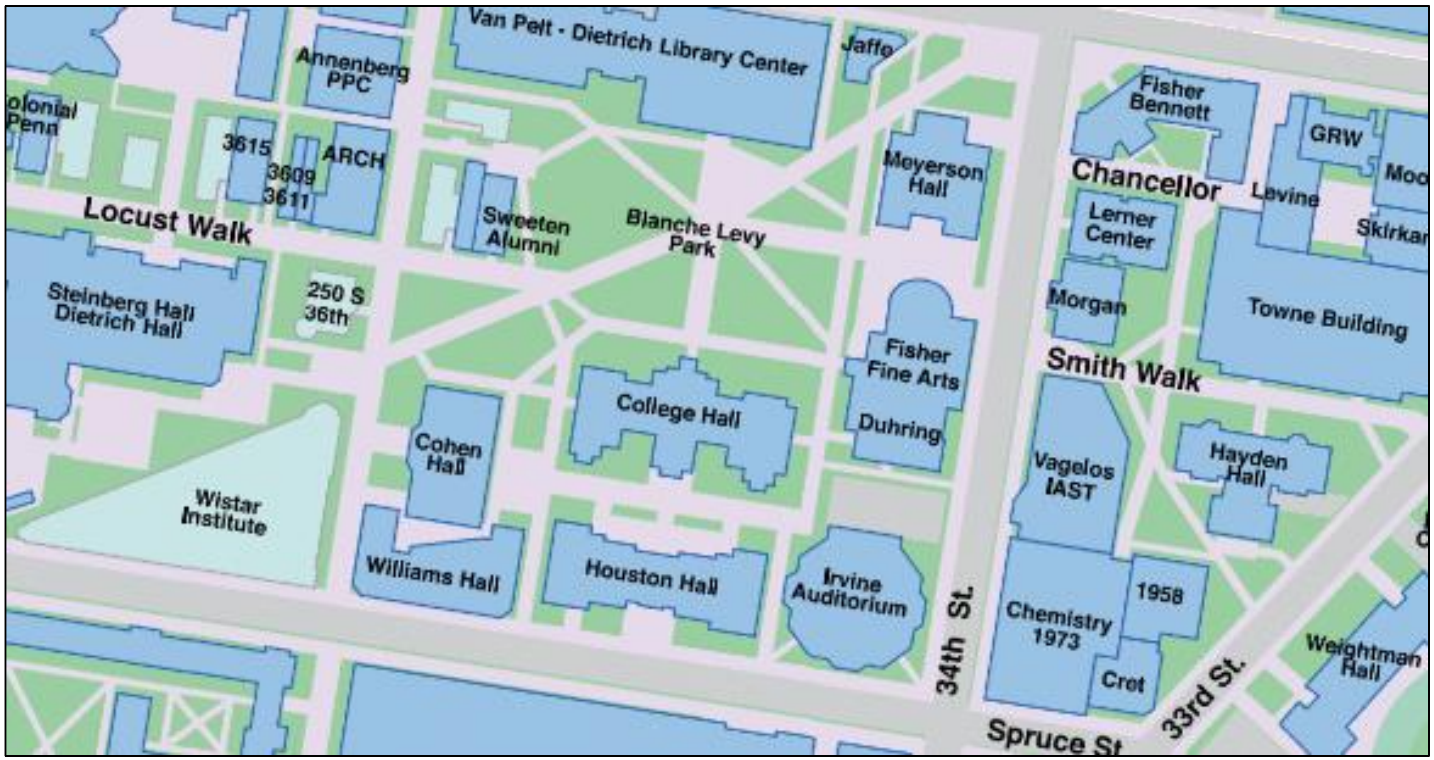
Types of Map (2) – Topological map



Graph representation is useful for path planning.

Types of Map (3) – Semantic map

Part of UPenn Campus Building Map



Semantic map is a map with labels.

Types of Map

- Metric Map
- Topological Map
- Semantic Map

Mapping

- What make it challenging?
  - $\boxed{\text{Noisy measurement in local coordinate}}$
  - Motion involved
  - Change over time

Acknowledgement

- Thanks to Rei Suzuki, Dan Lee’s master student at the University of Pennsylvania, for helping us create the lectures for WEEK 3.

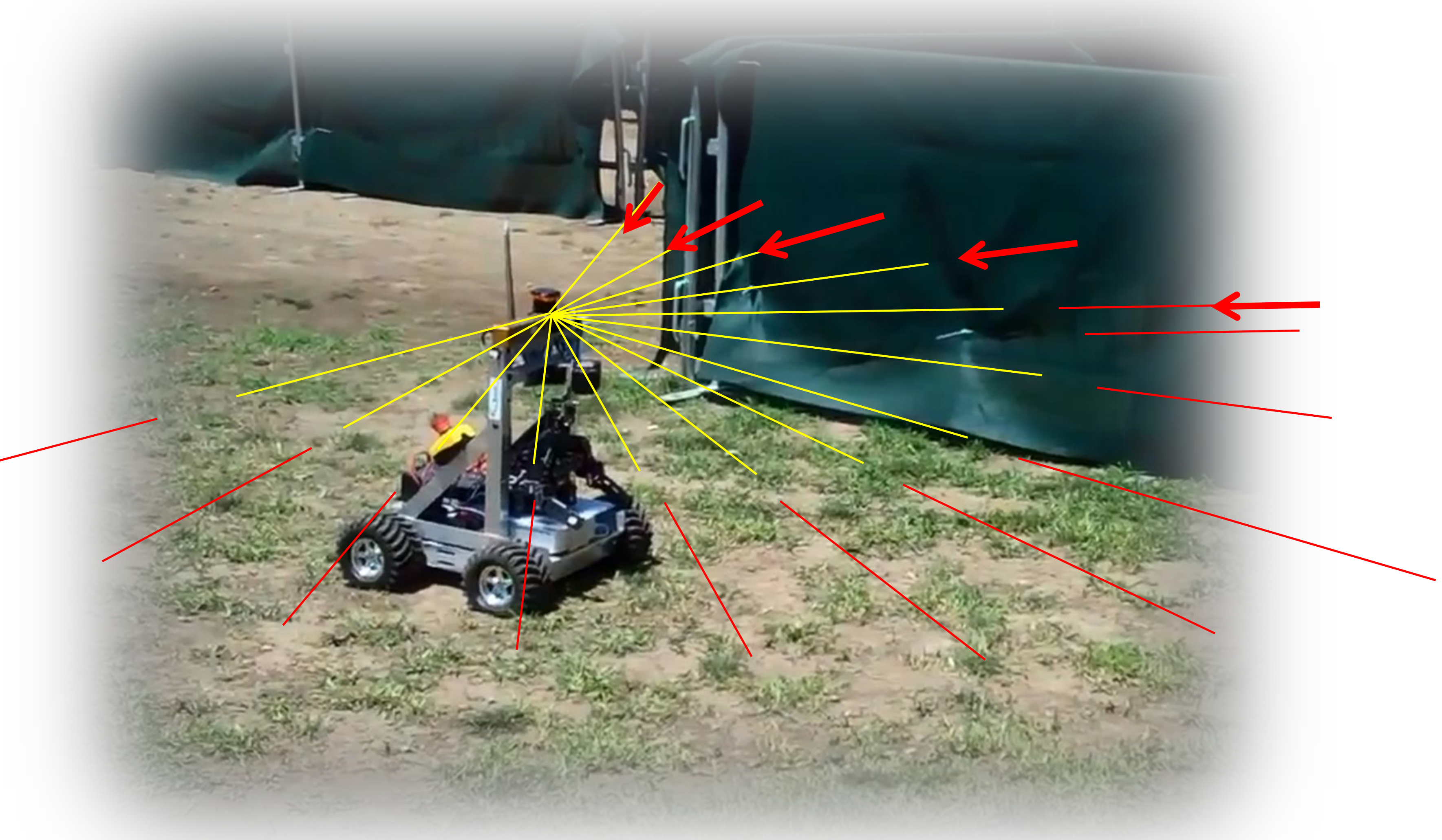
Robotics

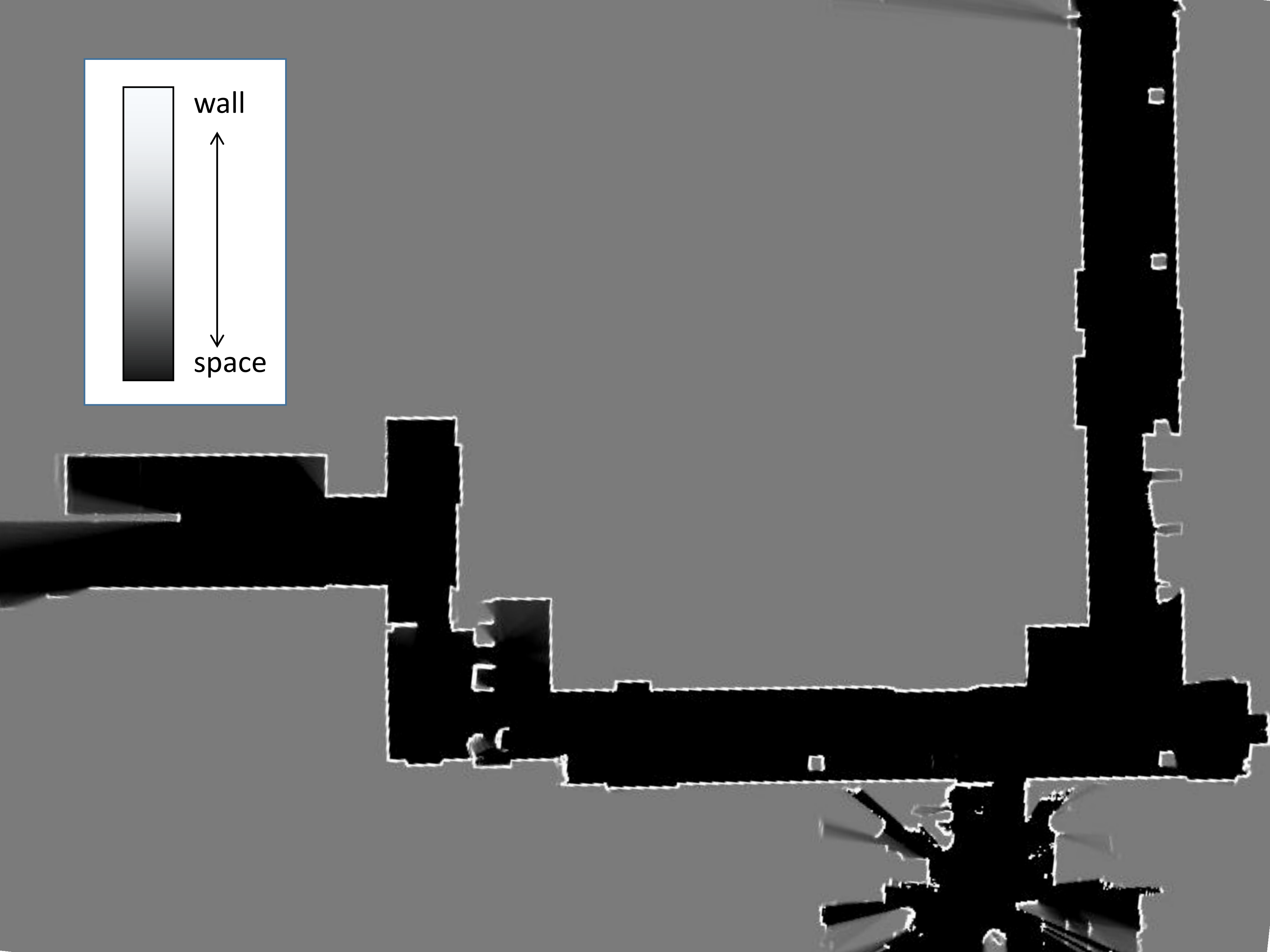
Estimation and Learning
with Dan Lee

Week 3.
Robotic Mapping

3.2 Occupancy Grid Mapping
3.2.1 Occupancy Grid Map

Penn Engineering




wall
space

Occupancy Grid Mapping

- Occupancy: binary R.V.

  $m_{x,y}: \{free, occupied\} \rightarrow \{ 0 , 1 \}$

[Review – Into Probability]

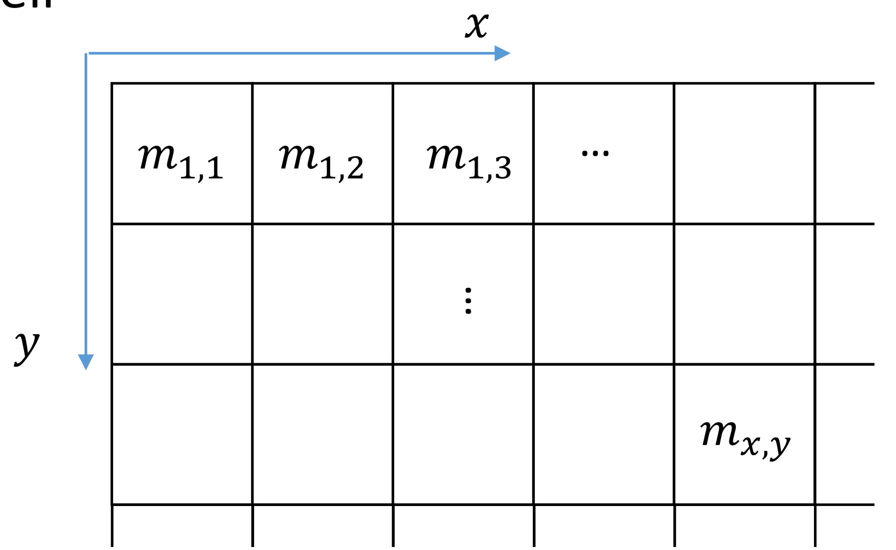
Given some probability space (Ω, P),
a **random variable** $X: \Omega \rightarrow R$ is a *function* that maps the sample space to the reals.

Occupancy Grid Mapping

- Occupancy: binary R.V.

$$m_{x,y}: \{free, occupied\} \rightarrow \{\,0\,,1\,\}$$

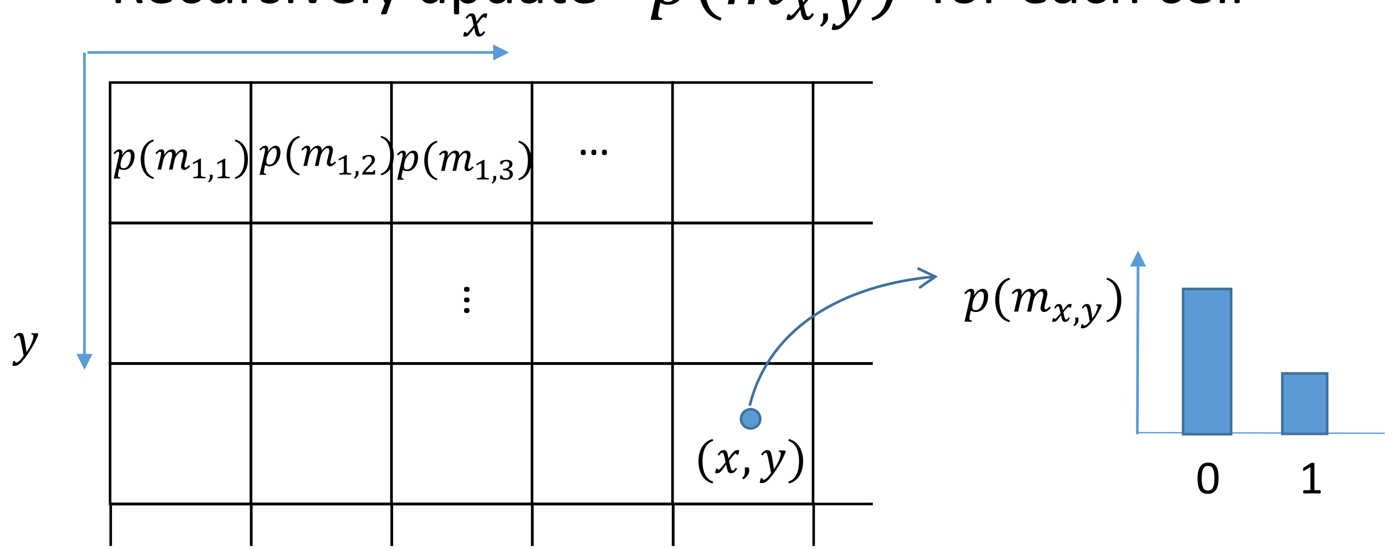
- Occupancy grid map
  : fine-grained grid map where an occupancy variable associated with each cell



Occupancy Grid Mapping

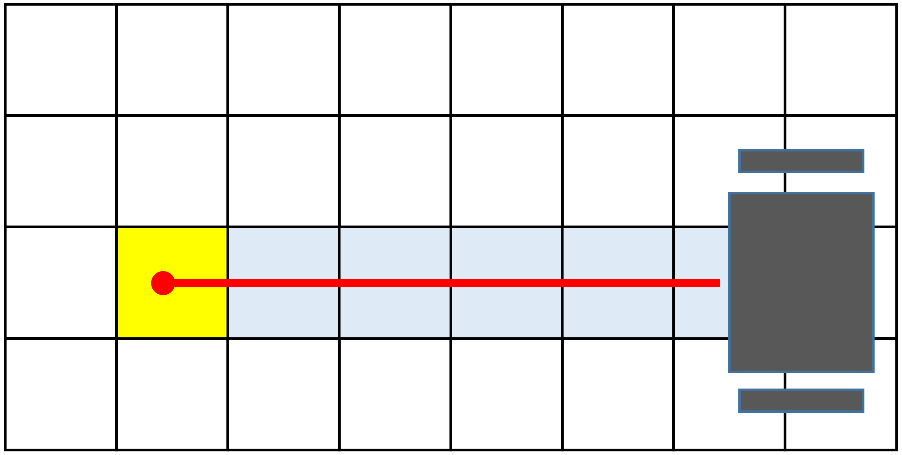
- Occupancy grid mapping
  : A Bayesian filtering to maintain a occupancy grid map.

Recursively update $p(m_{x,y})$ for each cell



Occupancy Grid Mapping

- Measurement



a range sensor

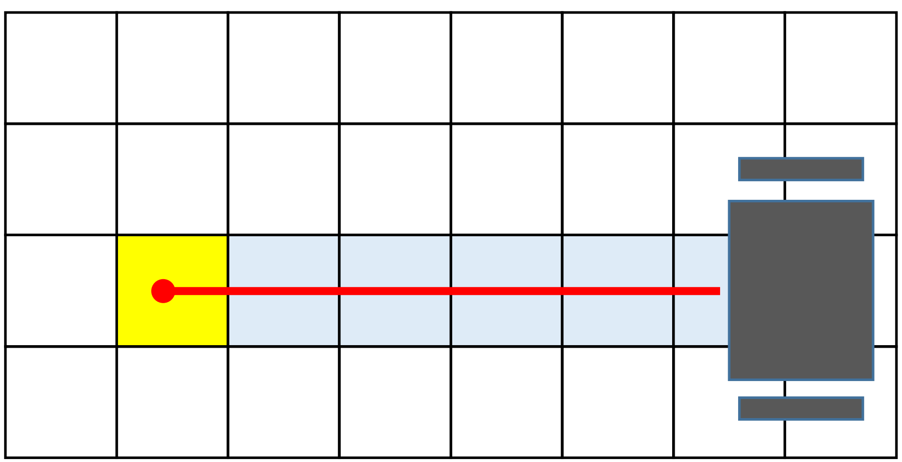
Occupancy Grid Mapping

- Measurement

Free

Occupied

$$z \sim \{0, 1\}$$



a range sensor

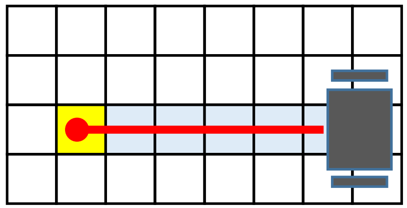
Occupancy Grid Mapping

- Measurement

Free

Occupied

$$z \sim \{0, 1\}$$



- Measurement model

$$p(z|m_{x,y})$$

$p(z = 1|m_{x,y} = 1)$: True **occupied** measurement

$p(z = 0|m_{x,y} = 1)$: False **free** measurement

$p(z = 1|m_{x,y} = 0)$: False **occupied** measurement

$p(z = 0|m_{x,y} = 0)$: True **free** measurement

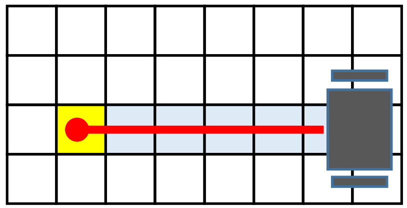
Occupancy Grid Mapping

- Measurement

Free

Occupied

$$z \sim \{\,0\,,1\,\}$$



- Measurement model

$$p(z|m_{x,y})$$

[Review – Into Probability]

$P(A^C|B) = 1 - P(A|B)$

$$\boxed{p\big(z = 1|m_{x,y} = 1\big)}$$

$$p\big(z = 0|m_{x,y} = 1\big) = 1 - \boxed{p\big(z = 1|m_{x,y} = 1\big)}$$

$$\boxed{p\big(z = 1|m_{x,y} = 0\big)}$$

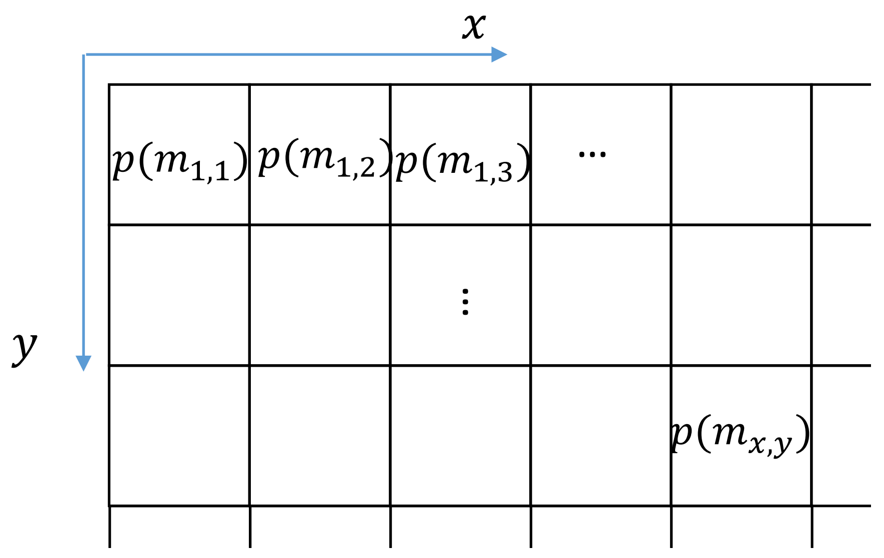
$$p\big(z = 0|m_{x,y} = 0\big) = 1 - \boxed{p\big(z = 1|m_{x,y} = 0\big)}$$

Occupancy Grid Mapping

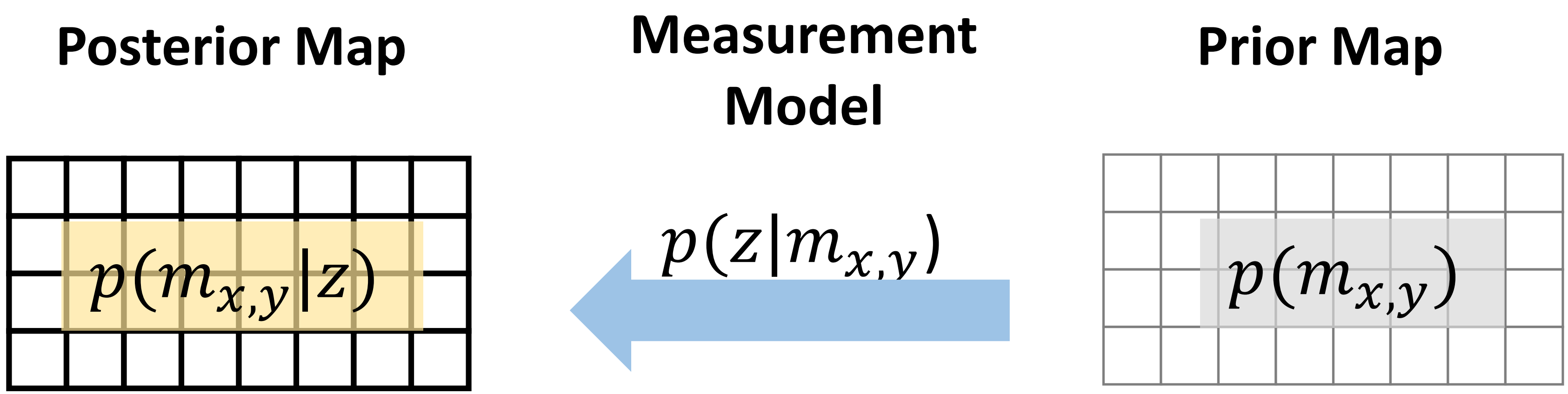
Measurement Model

$$p(z|m_{x,y})$$

Map



Occupancy Grid Mapping



Likelihood Prior

Posterior

$$p(m_{x,y}|z) = \frac{p(z|m_{x,y})\,p(m_{x,y})}{p(z)}$$

Evidence

Robotics

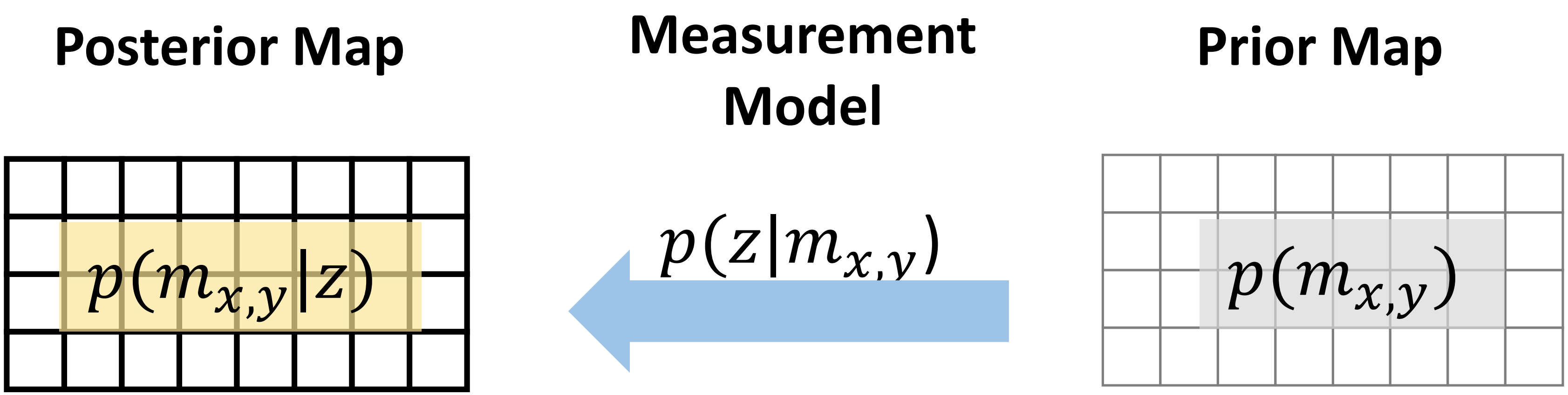
Estimation and Learning
with Dan Lee

Week 3.
Robotic Mapping

3.2 Occupancy Grid Mapping
3.2.2 Log-odd Update

Penn Engineering

Occupancy Grid Mapping



Bayes' Rule:

$$p\left(m_{x,y}|z\right) = \frac{p(z|m_{x,y})p(m_{x,y})}{p(z)}$$

Occupancy Grid Mapping

$$\boxed{Odd := \frac{(X\ happens)}{(X\ not\ happens)} = \frac{p(X)}{p(X^c)}}$$

- More convenient when we use "Odd"

$$Odd((m_{x,y} = 1)\ given\ z) = \frac{p(m_{x,y} = 1|z)}{p(m_{x,y} = 0|z)}$$

Occupancy Grid Mapping

- Odd

(Bayes' Rule)

$$p(m_{x,y}=1|z)=\frac{p(z|m_{x,y}=1)p(m_{x,y}=1)}{p(z)}$$

$$Odd=\frac{p(m_{x,y}=1|z)}{p(m_{x,y}=0|z)}=\frac{p(z|m_{x,y}=1)p(m_{x,y}=1)/p(z)}{p(m_{x,y}=0|z)}$$

Occupancy Grid Mapping

- Odd

$$Odd = \frac{p(m_{x,y}=1|z)}{p(m_{x,y}=0|z)} = \frac{p(z|m_{x,y}=1)p(m_{x,y}=1)/\cancel{p(z)}}{p(z|m_{x,y}=0)p(m_{x,y}=0)/\cancel{p(z)}}$$

$$p(m_{x,y}=0|z) = \frac{p(z|m_{x,y}=0)p(m_{x,y}=0)}{p(z)}$$

(Bayes' Rule)

Occupancy Grid Mapping

- Take the log!

Odd:
$$\frac{p(m_{x,y}=1|z)}{p(m_{x,y}=0|z)}=\frac{p(z|m_{x,y}=1)p(m_{x,y}=1)}{p(z|m_{x,y}=0)p(m_{x,y}=0)}$$

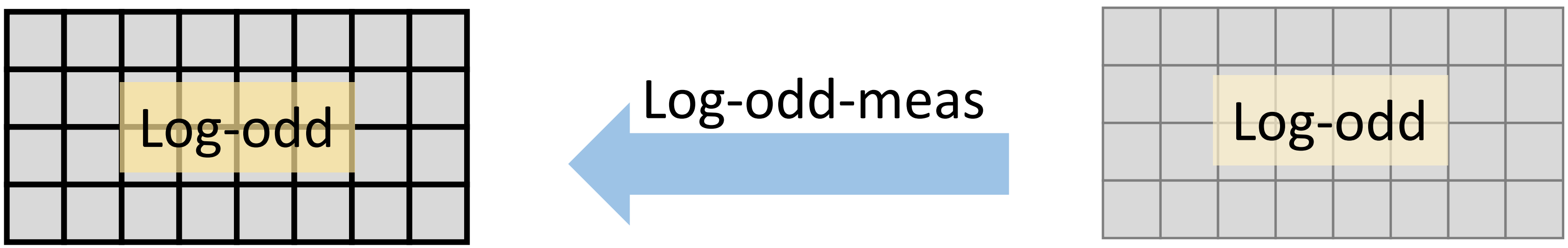
Log-Odd:
$$\log\frac{p(m_{x,y}=1|z)}{p(m_{x,y}=0|z)}=\log\frac{p(z|m_{x,y}=1)p(m_{x,y}=1)}{p(z|m_{x,y}=0)p(m_{x,y}=0)}$$
$$=\log\frac{p\left(z|m_{x,y}=1\right)}{p\left(z|m_{x,y}=0\right)}+\log\frac{p\left(m_{x,y}=1\right)}{p\left(m_{x,y}=0\right)}$$

$$\boxed{\log odd^{+}=\log odd\ meas+\log odd^{-}}$$

Occupancy Grid Mapping

- Log-odd update

Posterior Map **Measurement Model** **Prior Map**

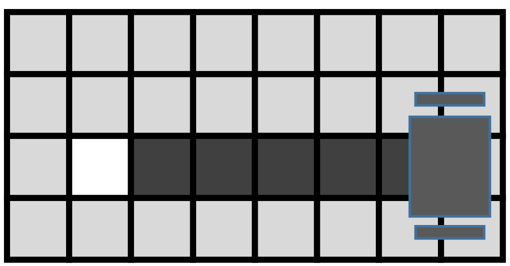


$$\log odd^{+} = \log odd\ meas + \log odd^{-}$$

Occupancy Grid Mapping

- Log-odd update

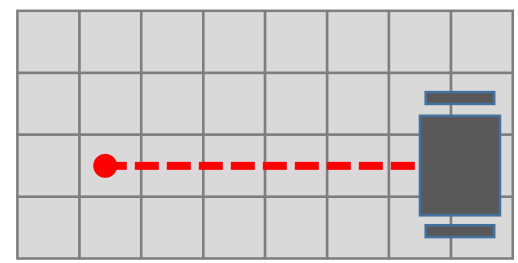
Posterior Map



Measurement Model

Log-odd-meas

Prior Map



$$\log odd^{+} = \log odd\ meas + \log odd^{-}$$

Occupancy Grid Mapping

- Measurement model in log-odd form

$$\log \frac{p(z|m_{x,y}=1)}{p(z|m_{x,y}=0)}$$

- **Two possible measurement:**

Case I : cells with z=1

$$\log odd\_occ := \log \frac{p(z=1|m_{x,y}=1)}{p(z=1|m_{x,y}=0)}$$

Case II : cells with z=0

$$\log odd\_free := \log \frac{p(z=0|m_{x,y}=0)}{p(z=0|m_{x,y}=1)}$$

(Trivial Case : cells not measured)

Occupancy Grid Mapping

- Example

Constant Measurement Model

$$\log odd\_occ := 0.9$$

$$\log odd\_free := 0.7$$

Initial Map:

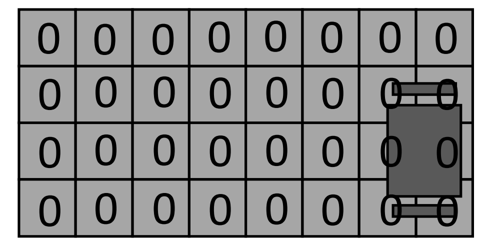
$\log odd = 0$ for all (x,y)

$$p(m_{x,y} = 1) = p(m_{x,y} = 0) = 0.5$$

Update Rule:

$$\boxed{\log\ odd \mathrel{+}= \log odd\_meas}$$

t0



Occupancy Grid Mapping

- Example

Constant Measurement Model

$$\log odd\_occ \coloneqq 0.9$$

$$\log odd\_free \coloneqq 0.7$$

Update

- Case I : cells with z=1

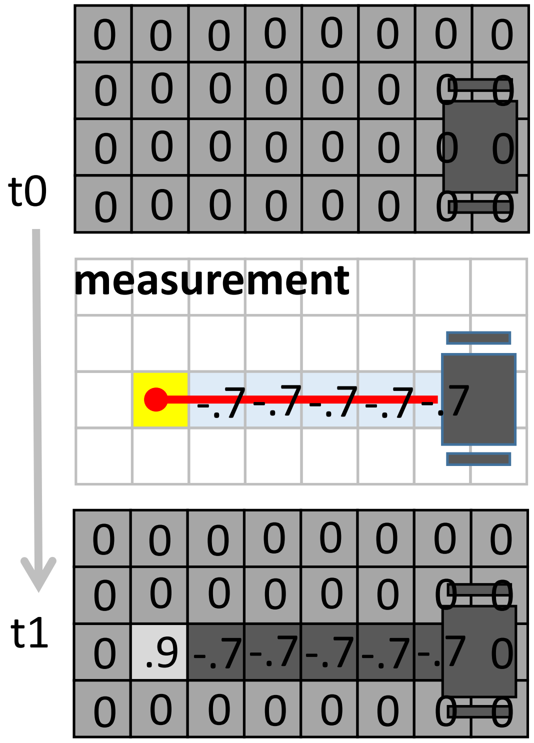
$$\log odd \leftarrow 0 + \log odd\_occ$$

- Case II : cells with z=0

$$\log odd \leftarrow 0 - \log odd\_free$$

Update Rule:

$$\log odd \mathrel{+}= \log odd\_meas$$



Occupancy Grid Mapping

- Example

<u>Constant Measurement Model</u>

$$\log odd\_occ := 0.9$$

$$\log odd\_free := 0.7$$

<u>Update</u>

- Case I : cells with z=1

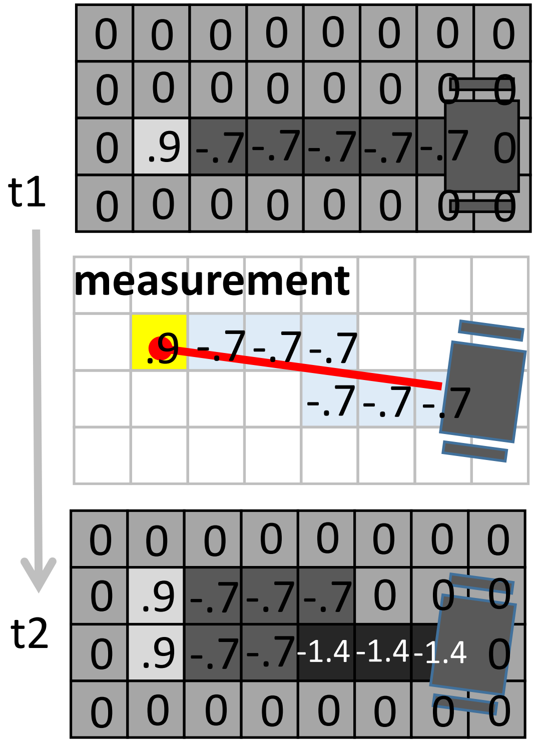
$$\log odd \leftarrow 0 + \log odd\_occ$$

- Case II : cells with z=0

$$\log odd \leftarrow 0 - \log odd\_free$$

Update Rule:

$$\boxed{\log odd \mathrel{+}= \log odd\_meas}$$



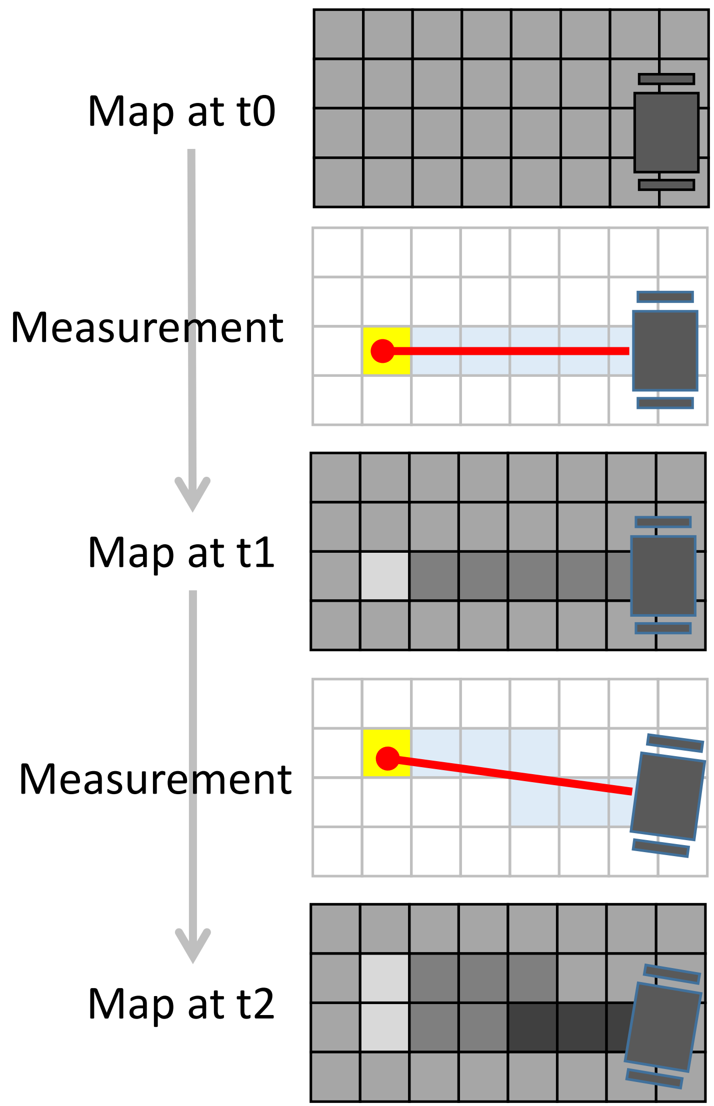
Robotics

Estimation and Learning
with Dan Lee

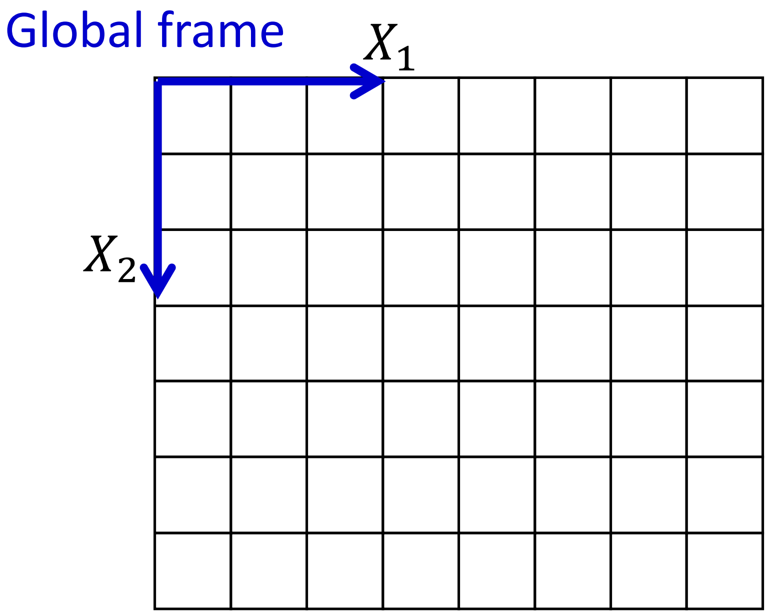
Week 3.
Robotic Mapping

3.2 Occupancy Grid Mapping
3.2.3 Handling Range Sensor

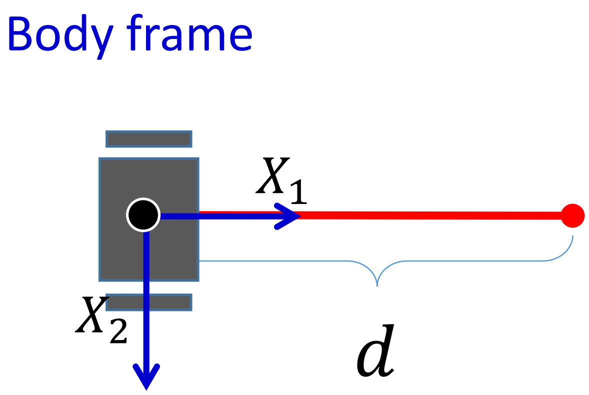
Penn Engineering


Map at t0
Measurement
Map at t1
Measurement
Map at t2

Handling Range Measurement on Grid

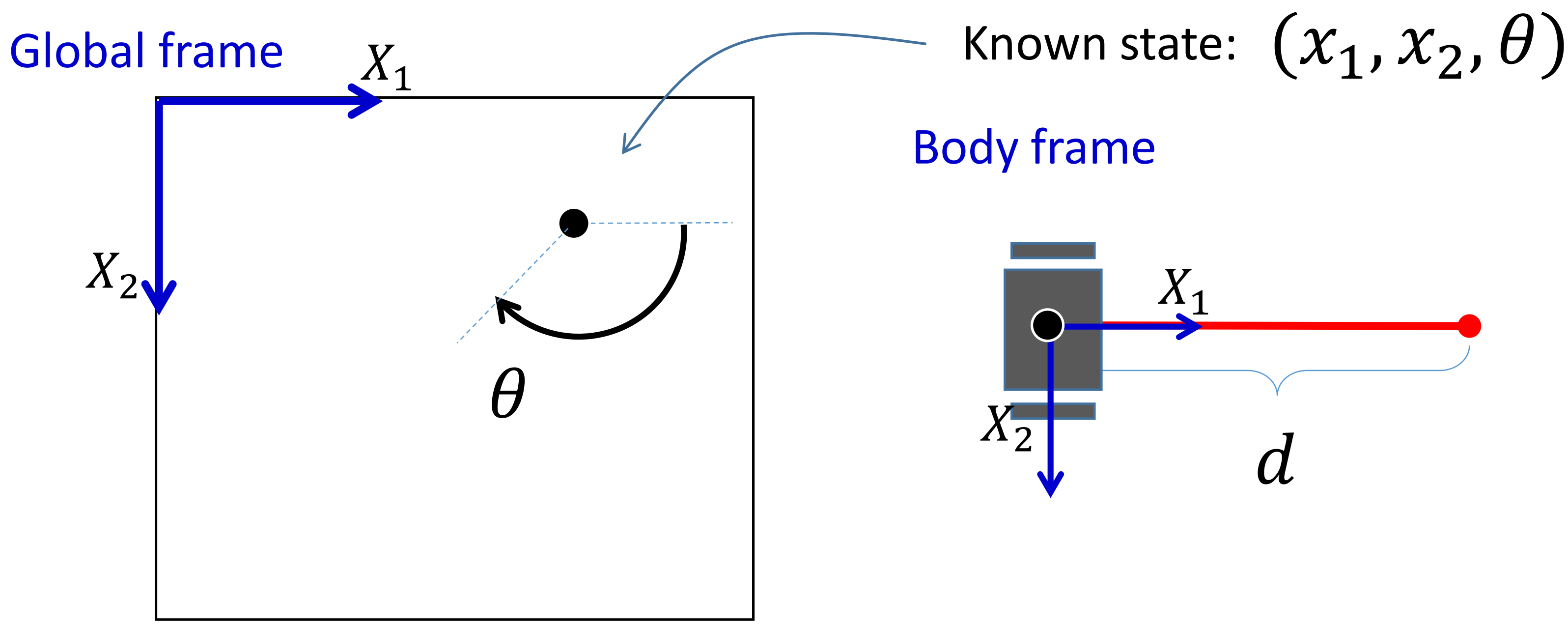


The Map



The Robot

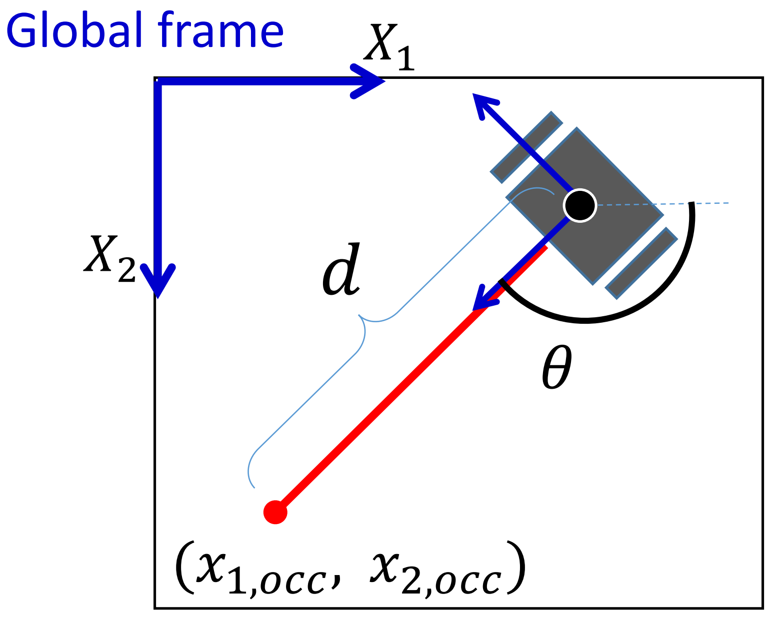
Handling Range Measurement on Grid



The Map

The Robot

Handling Range Measurement on Grid



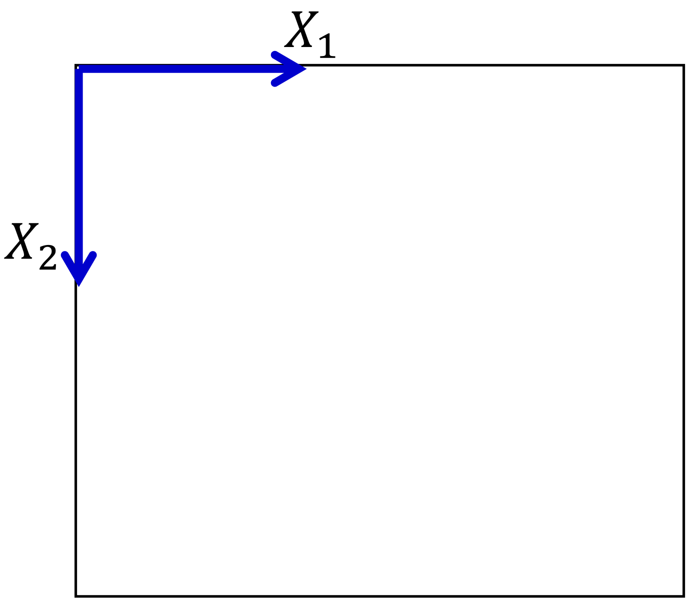
The Map

Distance measurement: d

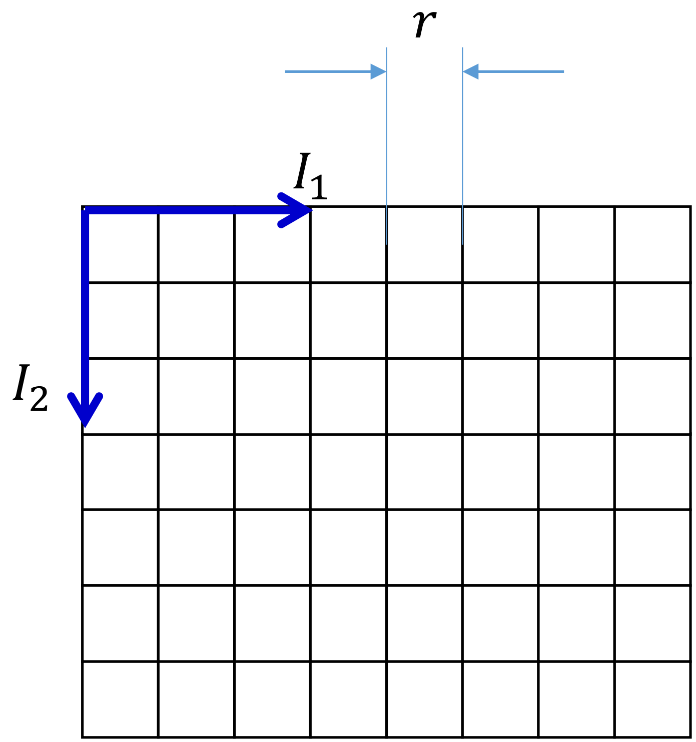
Known state: (x_1, x_2, θ)

$$\begin{bmatrix} x_{1,occ} \\ x_{2,occ} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Handling Range Measurement on Grid



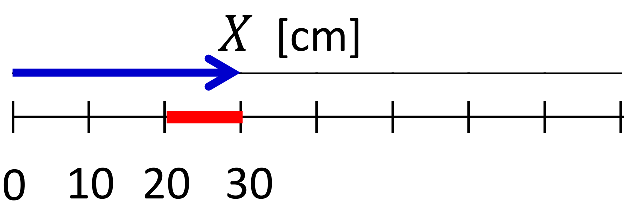
The (Continuous) Map

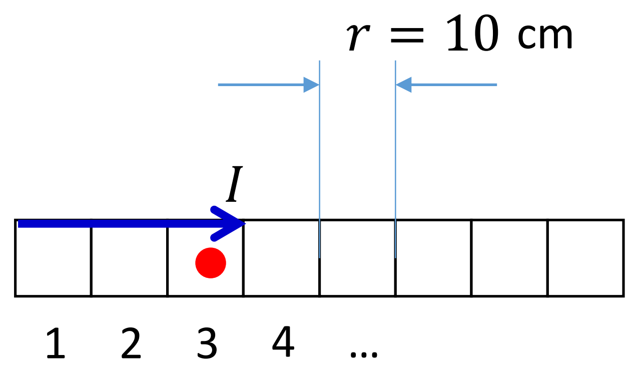


The **Discretized** Map

Handling Range Measurement on Grid

- Example

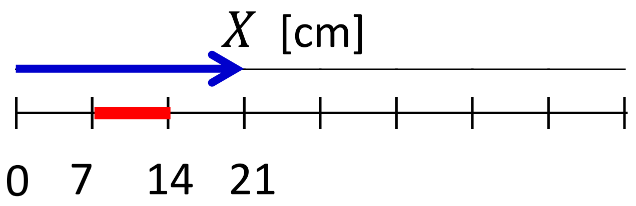


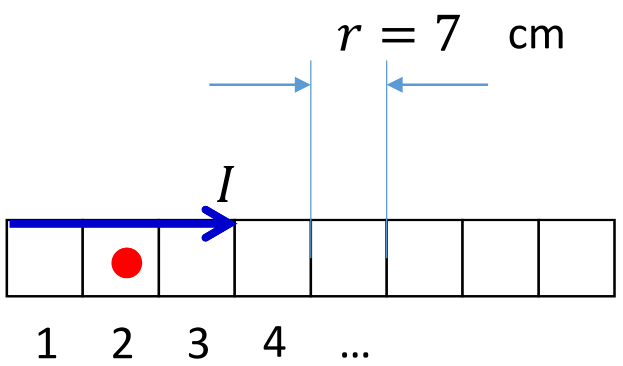


[cm]	$0 < x \leq 10$	$\Rightarrow$	$i = 1$	[index]
	$10 < x \leq 20$	$\Rightarrow$	$i = 2$	
	$20 < x \leq 30$	$\Rightarrow$	$i = 3$	
	$\vdots$		$\vdots$	

Handling Range Measurement on Grid

- Example



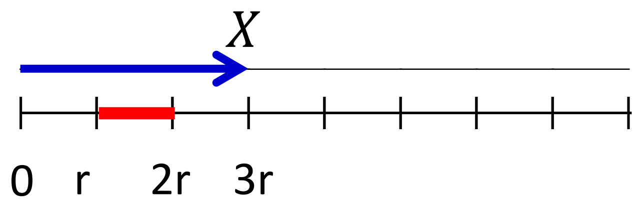


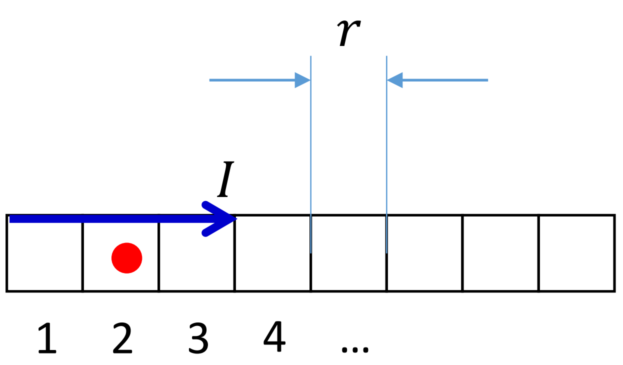
[cm] $0 < x \leq 7$ $i = 1$ [index]

$7 < x \leq 14$ $i = 2$

Handling Range Measurement on Grid

- Example

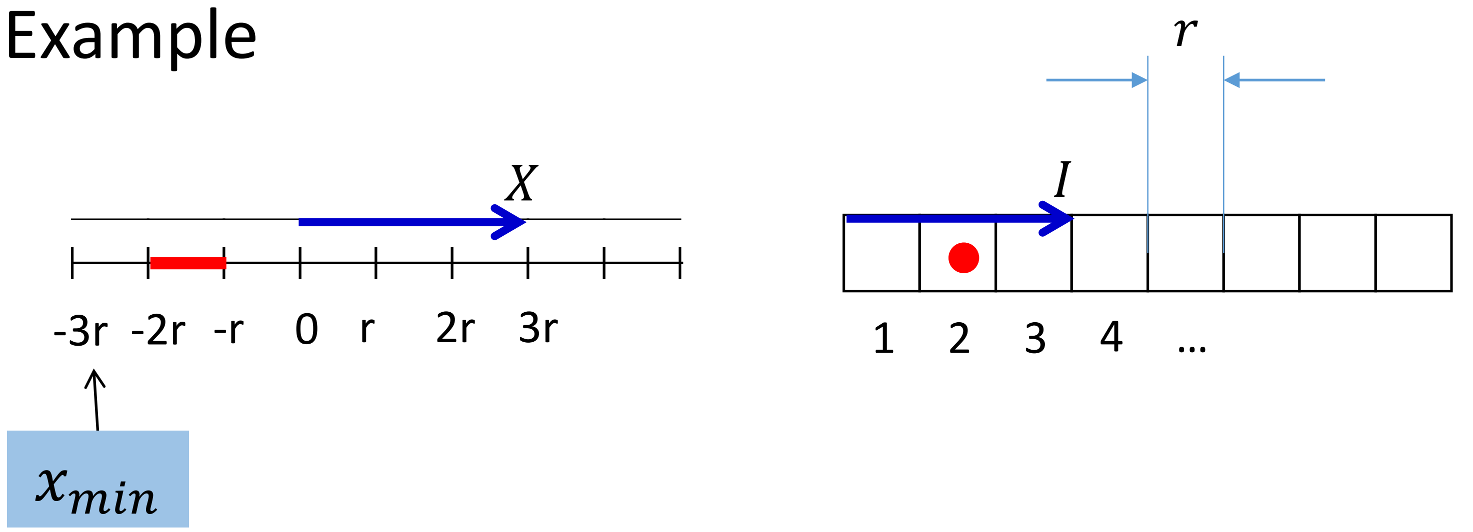




$$\boxed{i = ceil(x/r)}$$

Handling Range Measurement on Grid

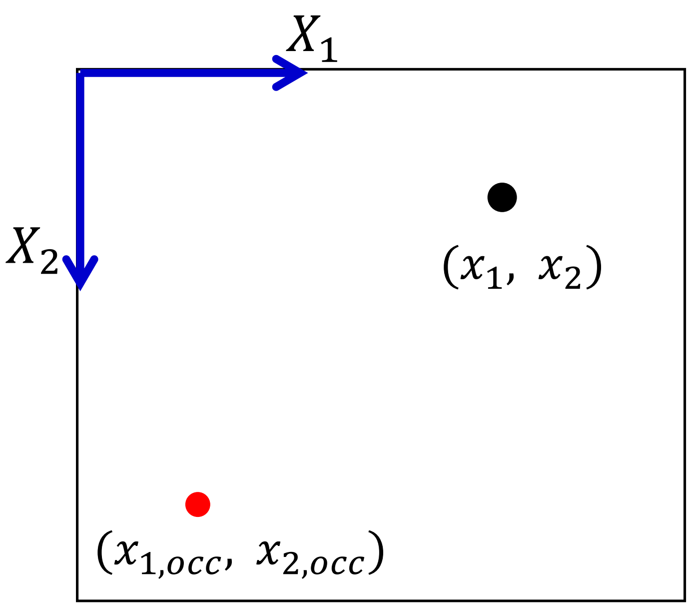
- Example



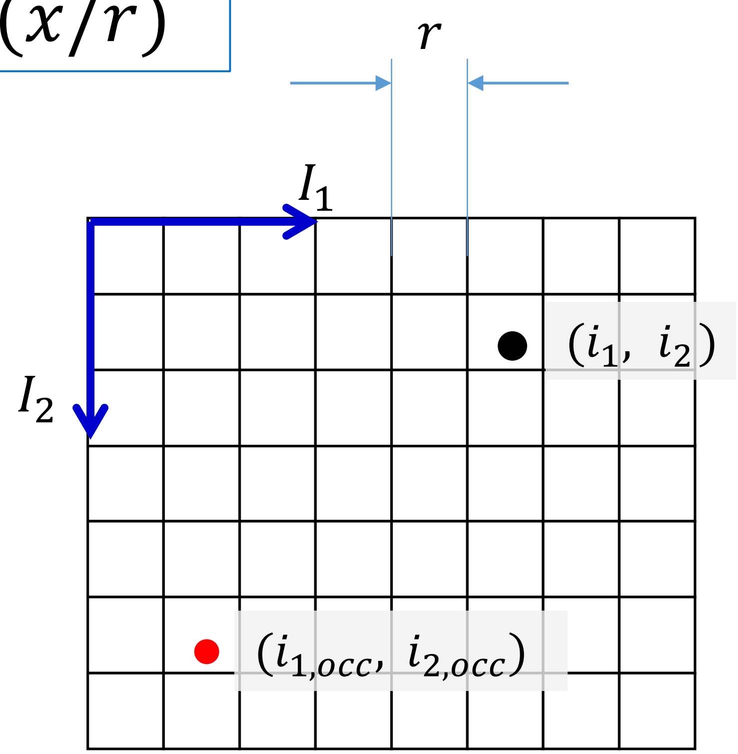
$$i = ceil((x - x_{min})/r)$$

Handling Range Measurement on Grid

$$i = ceil(x/r)$$

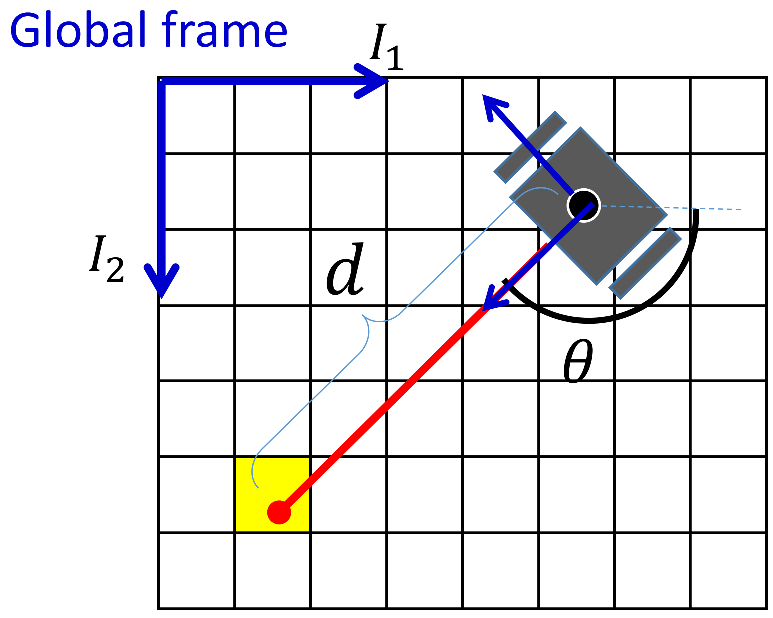


The (Continuous) Map



The **Discretized** Map

Handling Range Measurement on Grid



The Map

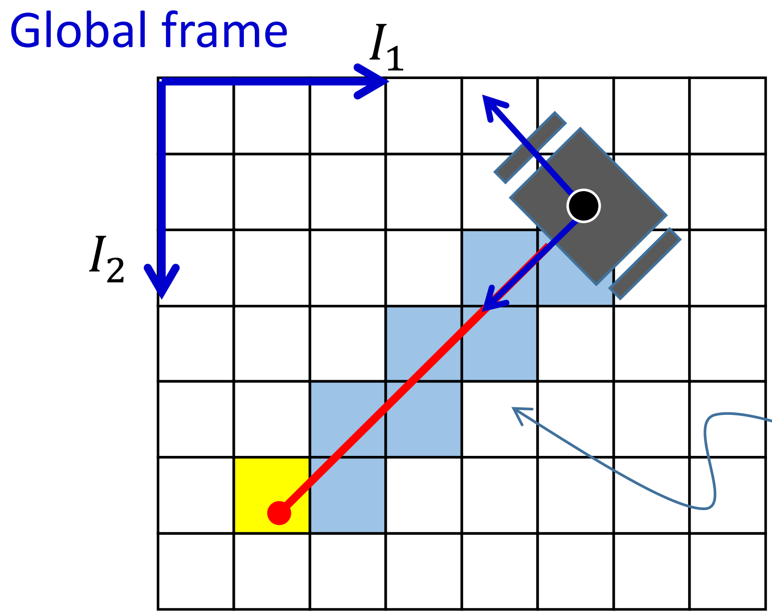
Distance measurement: d

Known state: (x_1, x_2, θ)

$$\begin{bmatrix} x_{1,occ} \\ x_{2,occ} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} i_{1,occ} \\ i_{2,occ} \end{bmatrix} = ceil\left(\frac{1}{r}\begin{bmatrix} x_{1,occ} \\ x_{2,occ} \end{bmatrix}\right)$$

Handling Range Measurement on Grid



The Map

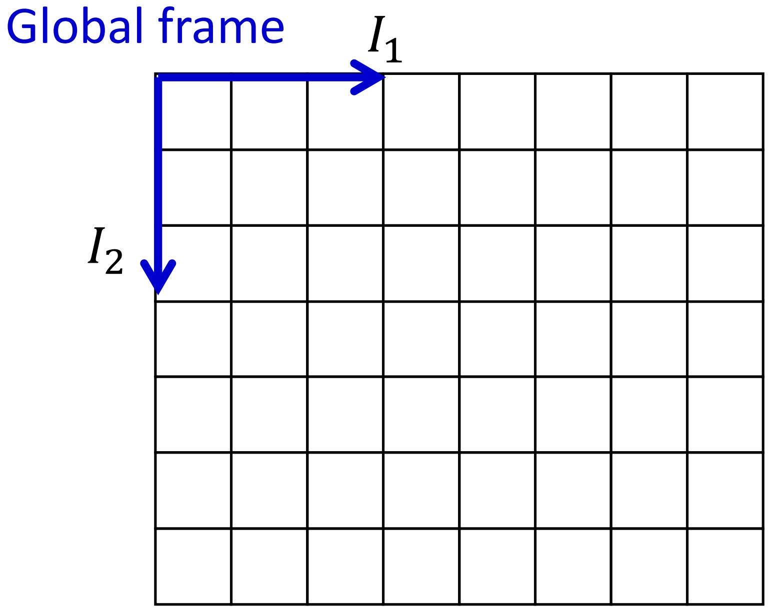
Distance measurement: d

Known state: (x_1, x_2, θ)

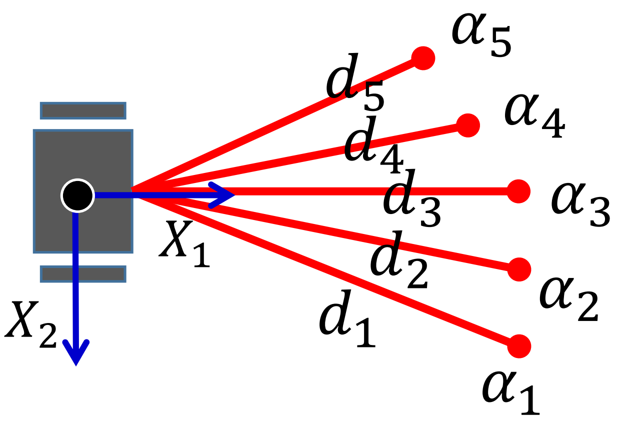
Occupied cell: $(x_{1,occ}, x_{2,occ})$

***Bresenham's line algorithm**

Handling Range Measurement on Grid

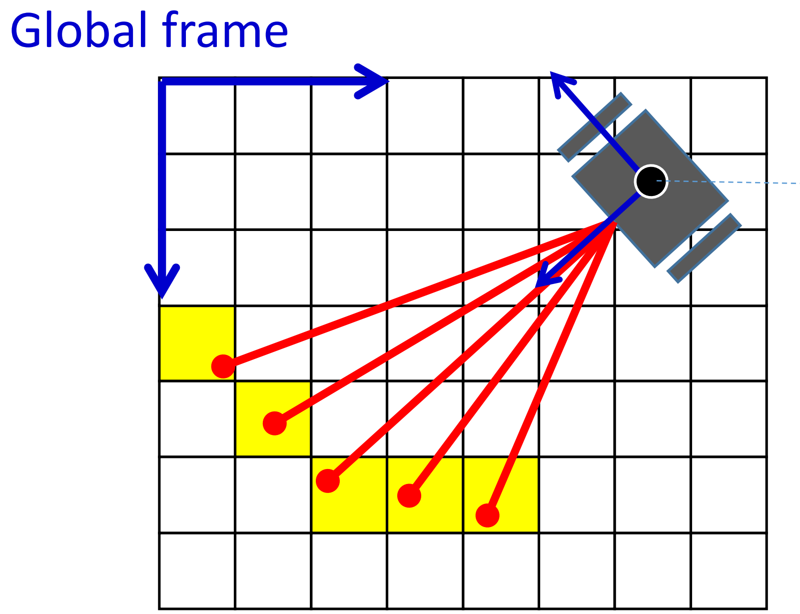


The Map



The Robot

Handling Range Measurement on Grid



The Map

Distance measurement:

$$(d_1, d_2, d_3, d_4, d_5)$$

Directions of rays:

$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$$

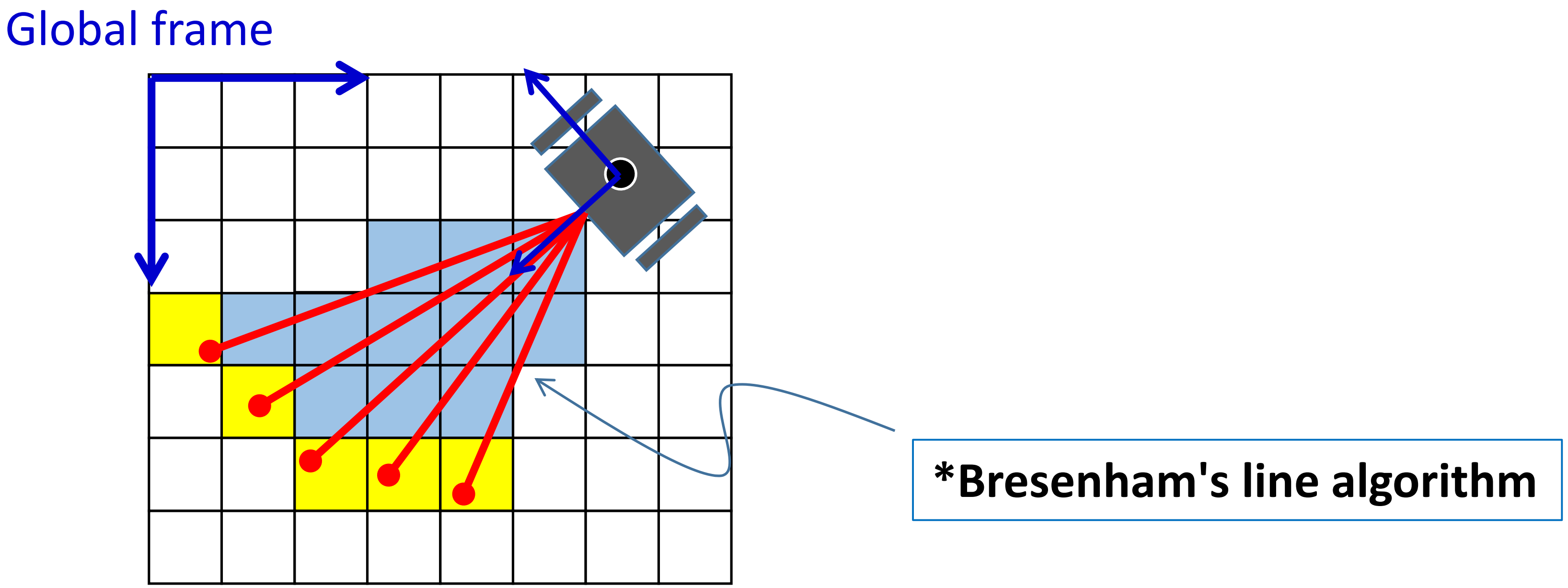
Known state: (x_1, x_2, θ)

For *k*-th occupied cell:

$$\begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix} = \begin{bmatrix} d_k \cos(\theta + \alpha_k) \\ -d_k \sin(\theta + \alpha_k) \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} i_{1k} \\ i_{2k} \end{bmatrix} = ceil\left(\frac{1}{r}\begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix}\right)$$

Handling Range Measurement on Grid



The Map

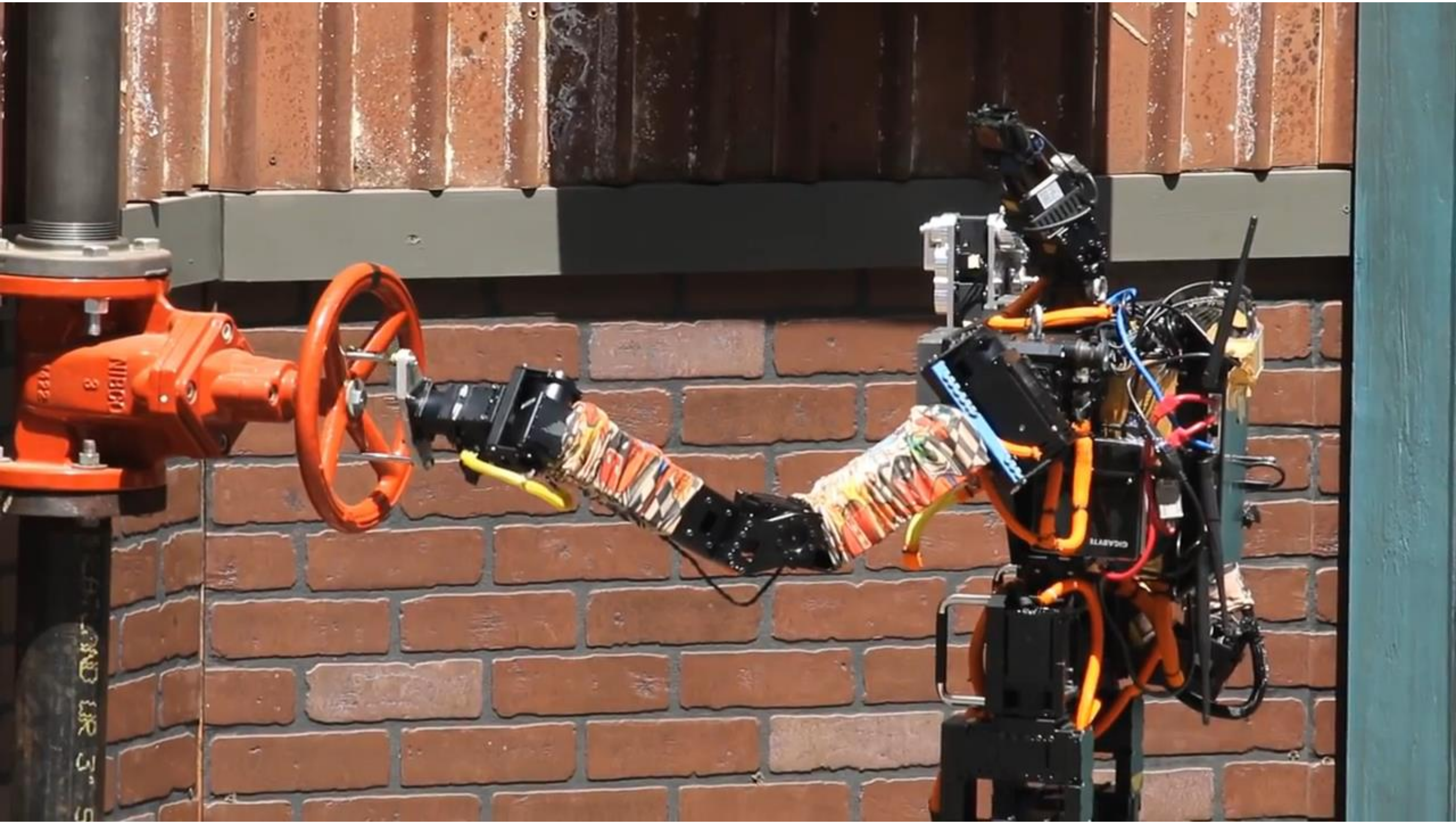
Robotics

Estimation and Learning
with Dan Lee

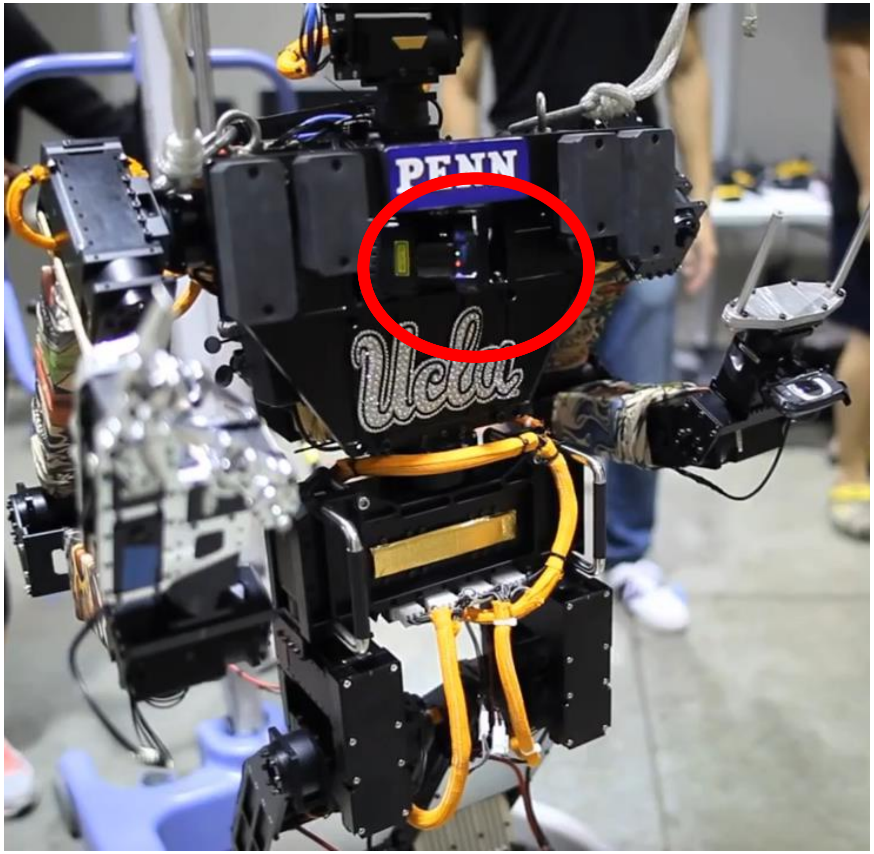
Week 3.
Robotic Mapping

3.3 3D Mapping
3.3.1 3D Sensors and 3D Map Representation

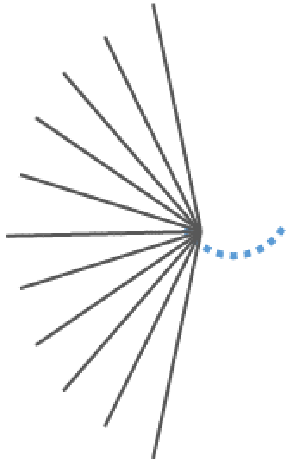
Penn Engineering


NIBCO
3

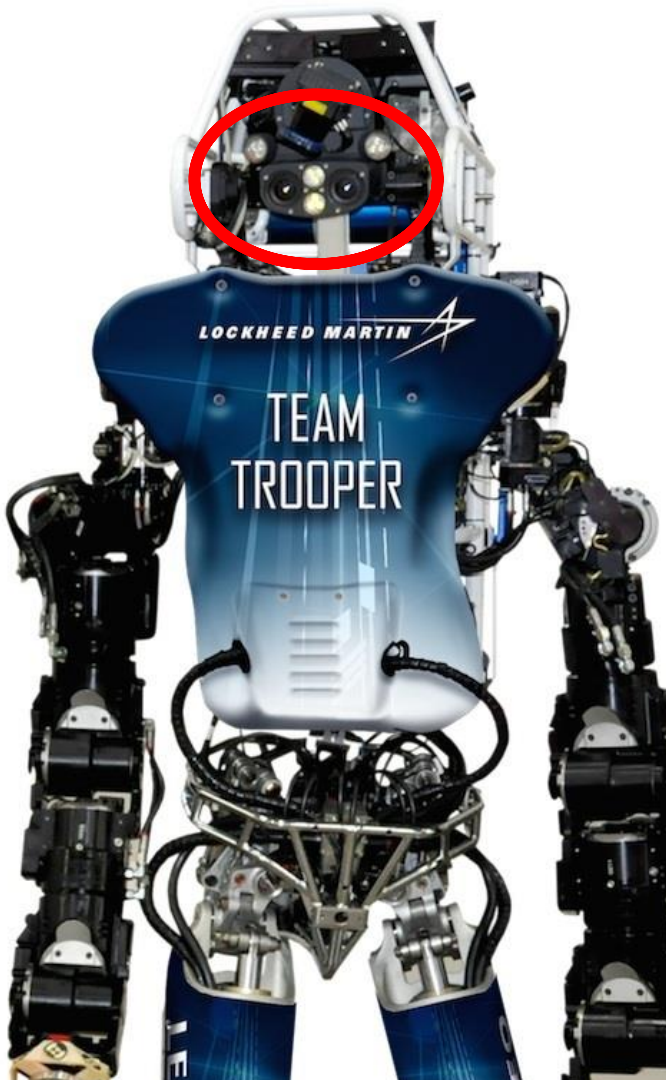
Sensors for 3D Mapping



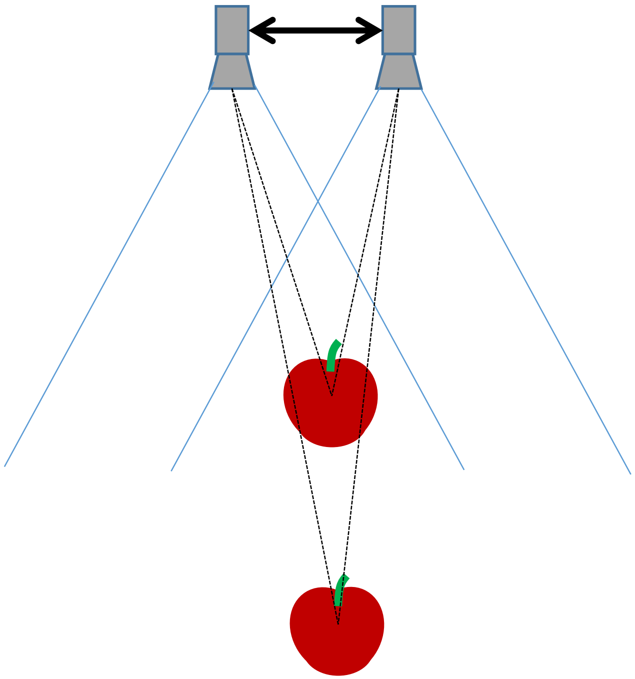
- **3D Range Sensor**



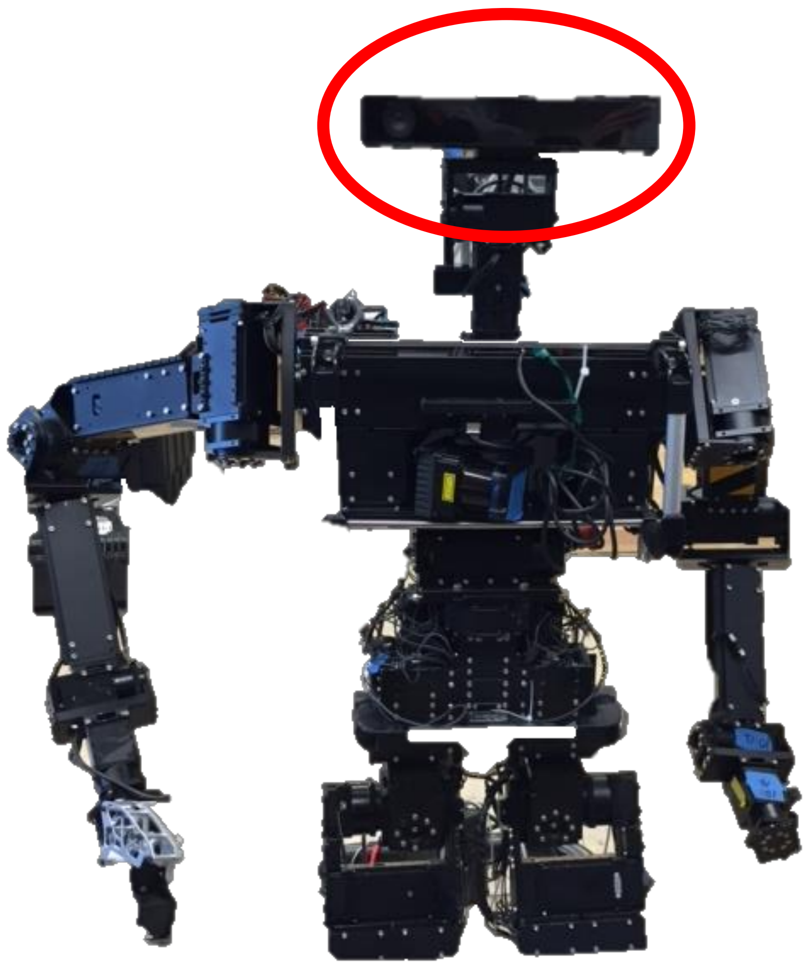
Sensors for 3D Mapping



- **Stereo Camera**



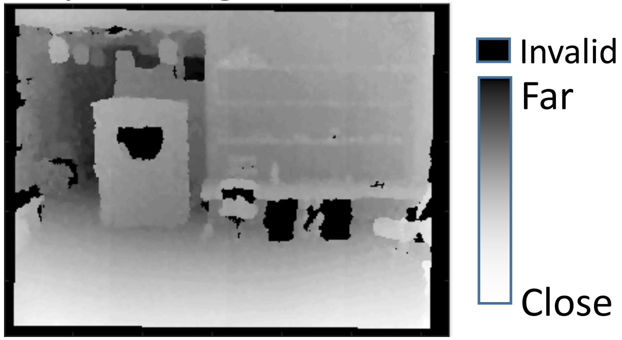
Sensors for 3D Mapping



- **Depth Camera**

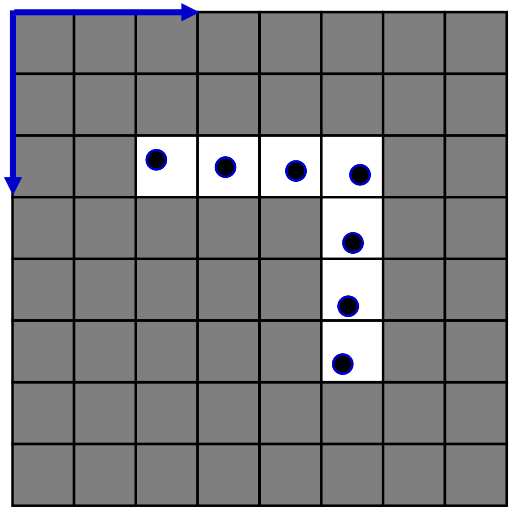
Scene

Depth Image



Map Representation

Grid
Representation

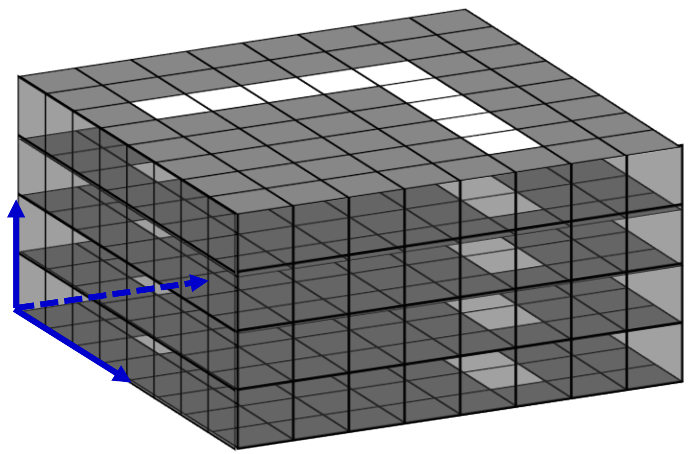


- Immediate access to a cell

- Requires Large memory
  (map size) ~(map range)/(resolution)
- Lose information from discretization

Map Representation

Grid
Representation



In 3D, most cells will be empty.

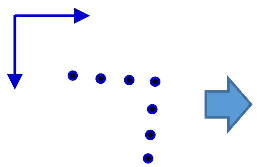
- Immediate access to a cell

- Requires Large memory
  (map size) ~(map range)/(resolution)

- Lose information from discretization

Map Representation

List
Representation



2.3, 2.3
2.5, 3.4
2.6, 4.6
2.7, 5.7
3.8, 5.5
4.8, 5.4
5.8, 5.2

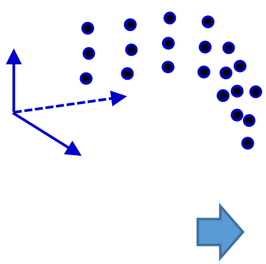
- Takes long to search (O(N))

- Requires less memory
  (map size) ~(# Occupied Points=*N*)

- No discretization

Map Representation

List
Representation



2.3, 2.3, 1.0
2.5, 3.4, 1.1
2.6, 4.6, 1.0
2.7, 5.7, 1.1
3.8, 5.5, 1.2
4.8, 5.4, 1.0
5.8, 5.2, 1.1
…

In 3D, N is usually very large.

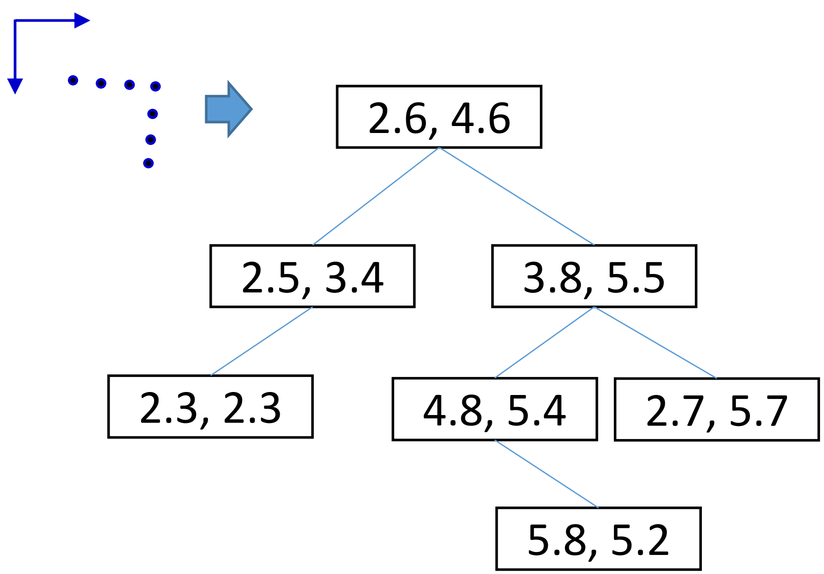
- Takes long to search (O(N))

- Requires less memory
  (map size) ~(# Occupied Points=*N*)

- No discretization

Map Representation

Tree Representation



- Reasonable search time (O(logN))

- Requires less memory (map size) ~(# Points)

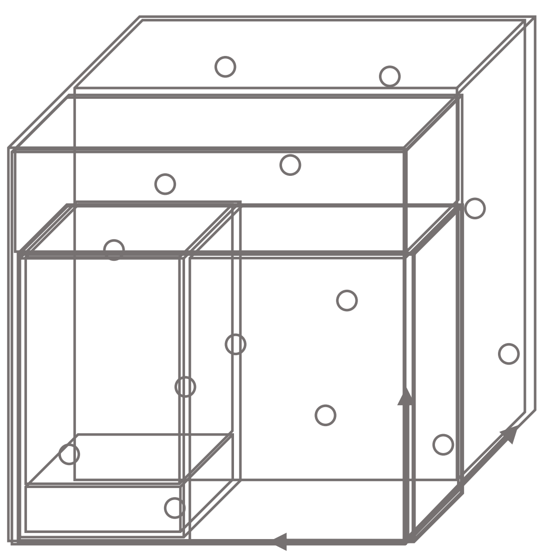
- No discretization

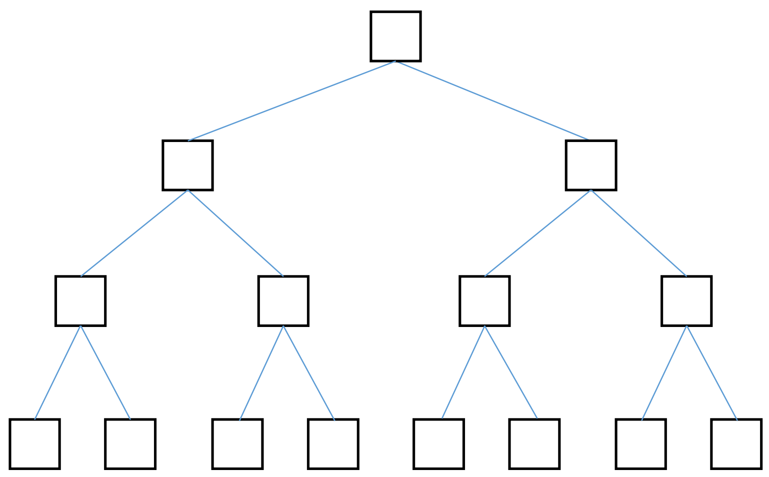
3D Map Representation

- Requires organized data structure for efficient maintenance of the map

3D Map Representation

- Example (1) *k*d-tree





3D Map Representation

- Example (2) Octree