MTRN4110 Robot Design
Week 2 – Localisation I

Liao “Leo” Wu, Lecturer

School of Mechanical and Manufacturing Engineering

University of New South Wales, Sydney, Australia

https://sites.google.com/site/wuliaothu/

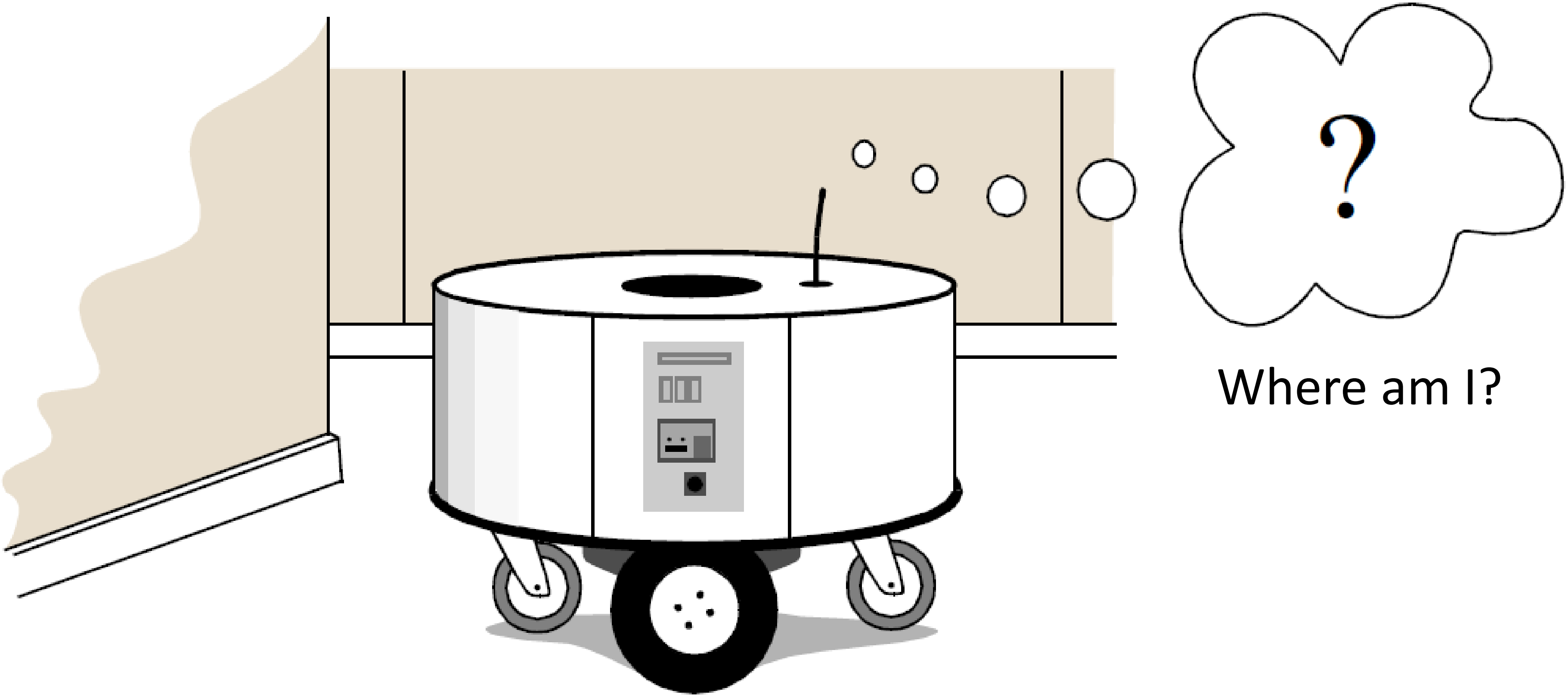
Today’s agenda

- Introduction to Localisation
- Map Representation
- Localisation Methods

Introduction to Localisation

Localisation

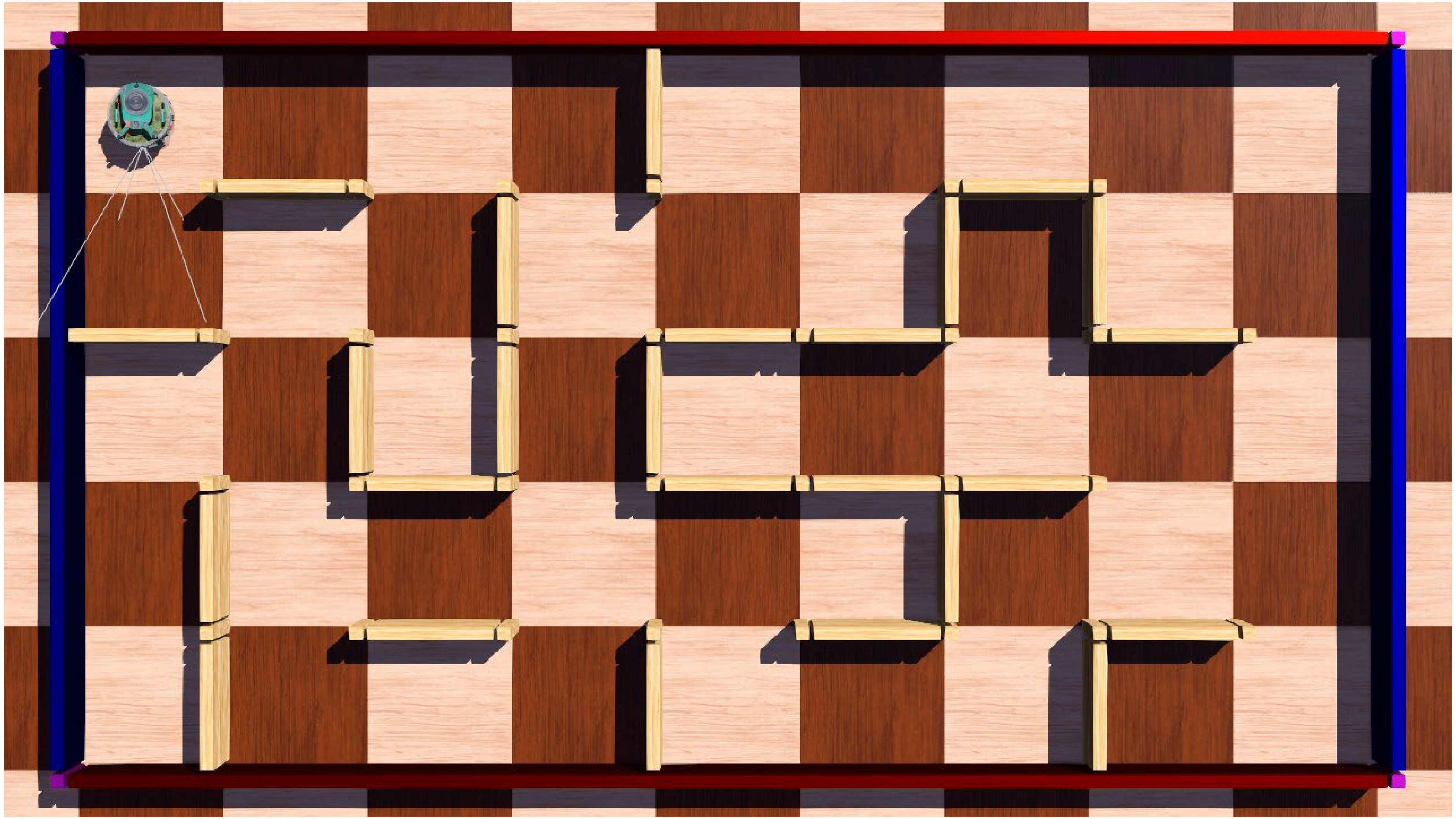
- The process that the robot determines its position in the environment



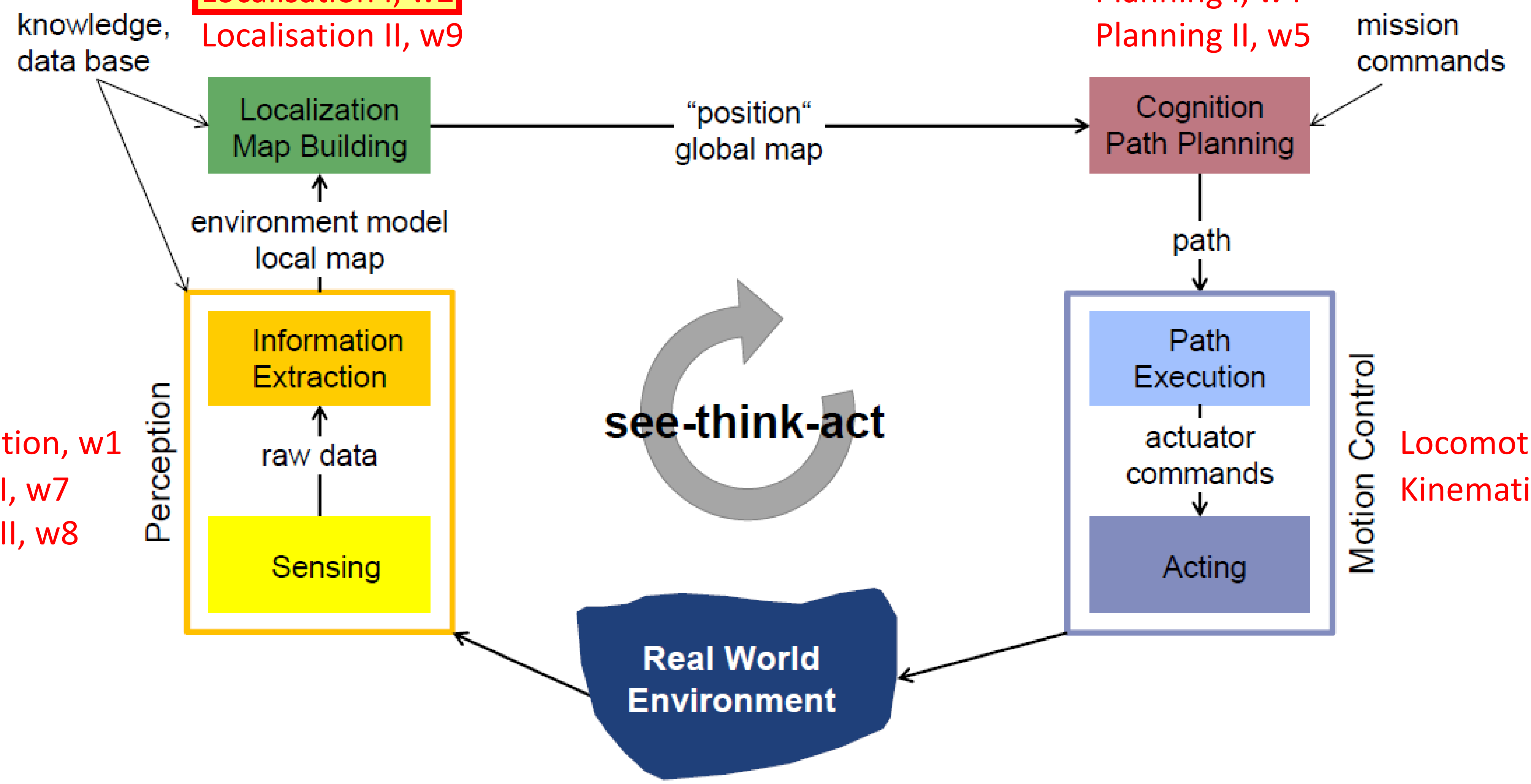
Where am I?

R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Localisation in the maze-solving task



The See-Think-Act cycle



Localisation I, w2

Localisation II, w9

Planning I, w4

Planning II, w5

Perception, w1

Vision I, w7

Vision II, w8

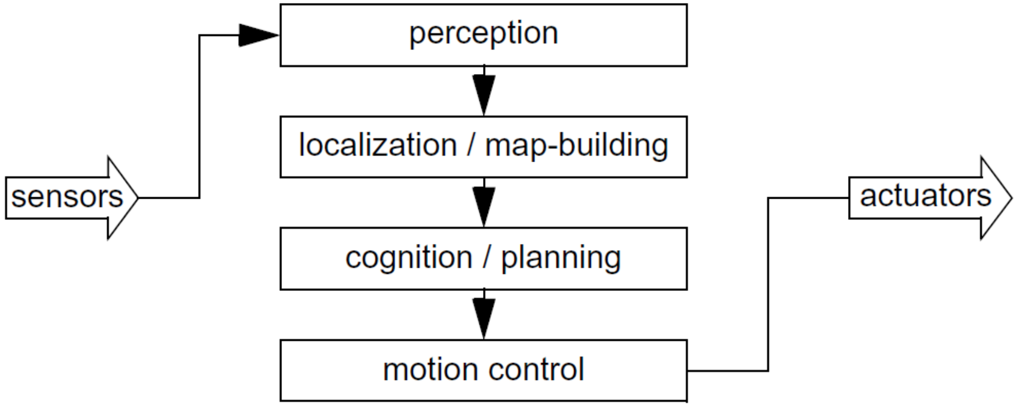
Locomotion, w1

Kinematics, w3

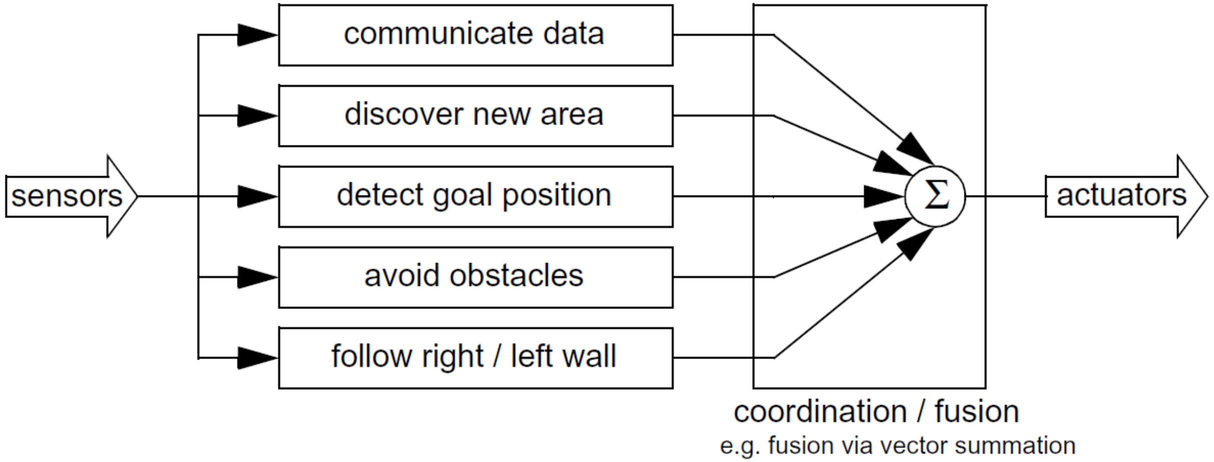
R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

To localise or not to localise

Map-based/Model-based navigation



Behaviour-based navigation



R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Which of the following is behaviour-based navigation?

https://www.sli.do/ #4110

-“Hi, I’m going to attend Leo’s lecture, do you know how to get to the theatre?”

-“Sure, you are now at the centre of Quadrangle Lawn, you just need to go south for 50 meters and then go west for 40 meters. You’ll be able to find the Webster Theatres. The lecture is on the second floor.”

-“Hi, the lecture is really boring. I’m gonna take a tram to join my friend’s party at the CBD, do you know where the nearest station is?”

-“Easy, you just need to get out of the door and step down the stairs to the ground floor. Walk around the Robert Webster Building clockwise until you reach the University Mall Road, then follow it in the downward direction for about 5min until you cross the Anzac Parade. You should then be able to see the station on your right-hand side. ***But seriously, why taking a tram when a bus is much faster???***

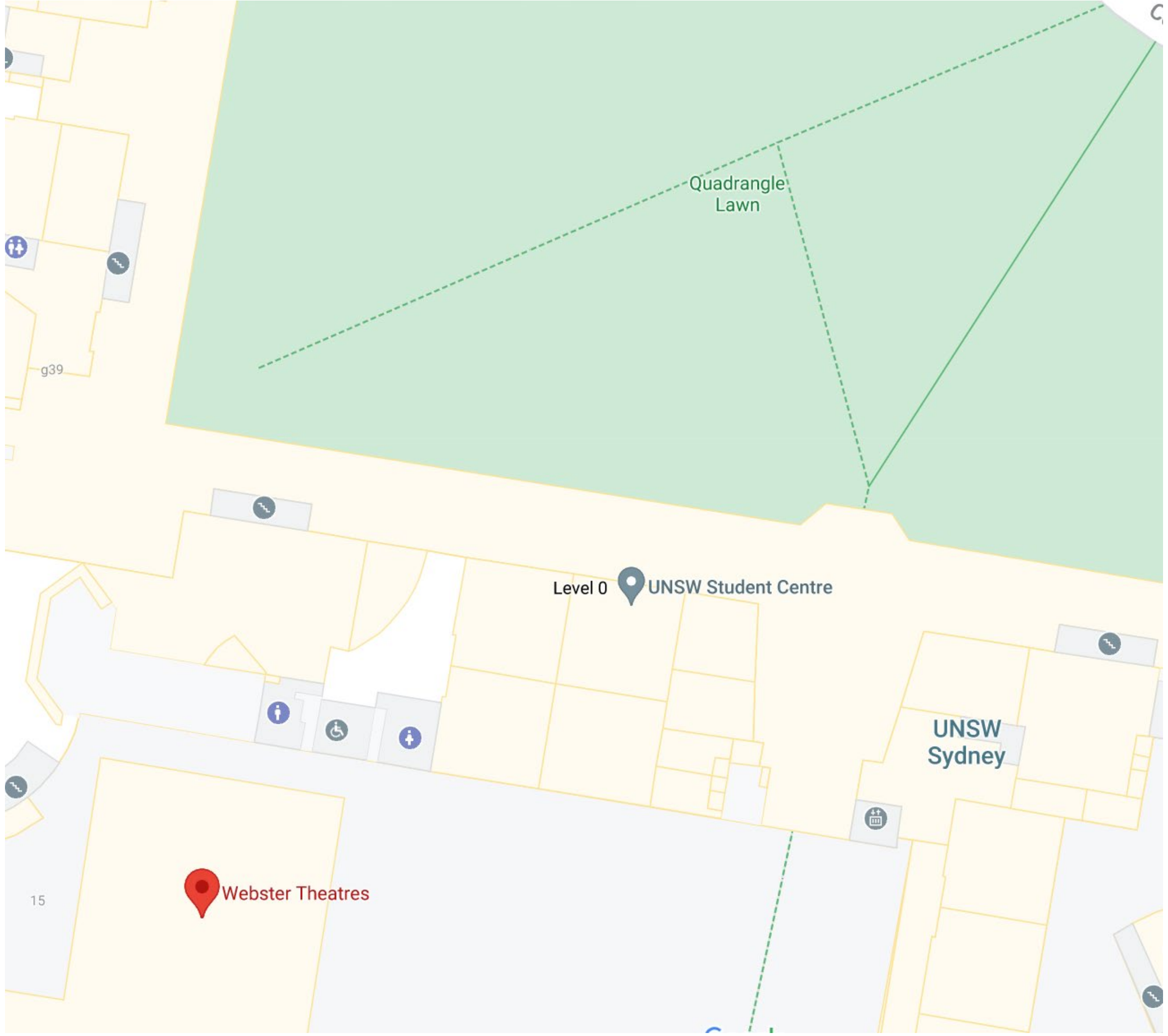
Which of the shown methods is behaviour-based navigation?

ⓘ Start presenting to display the poll results on this slide.

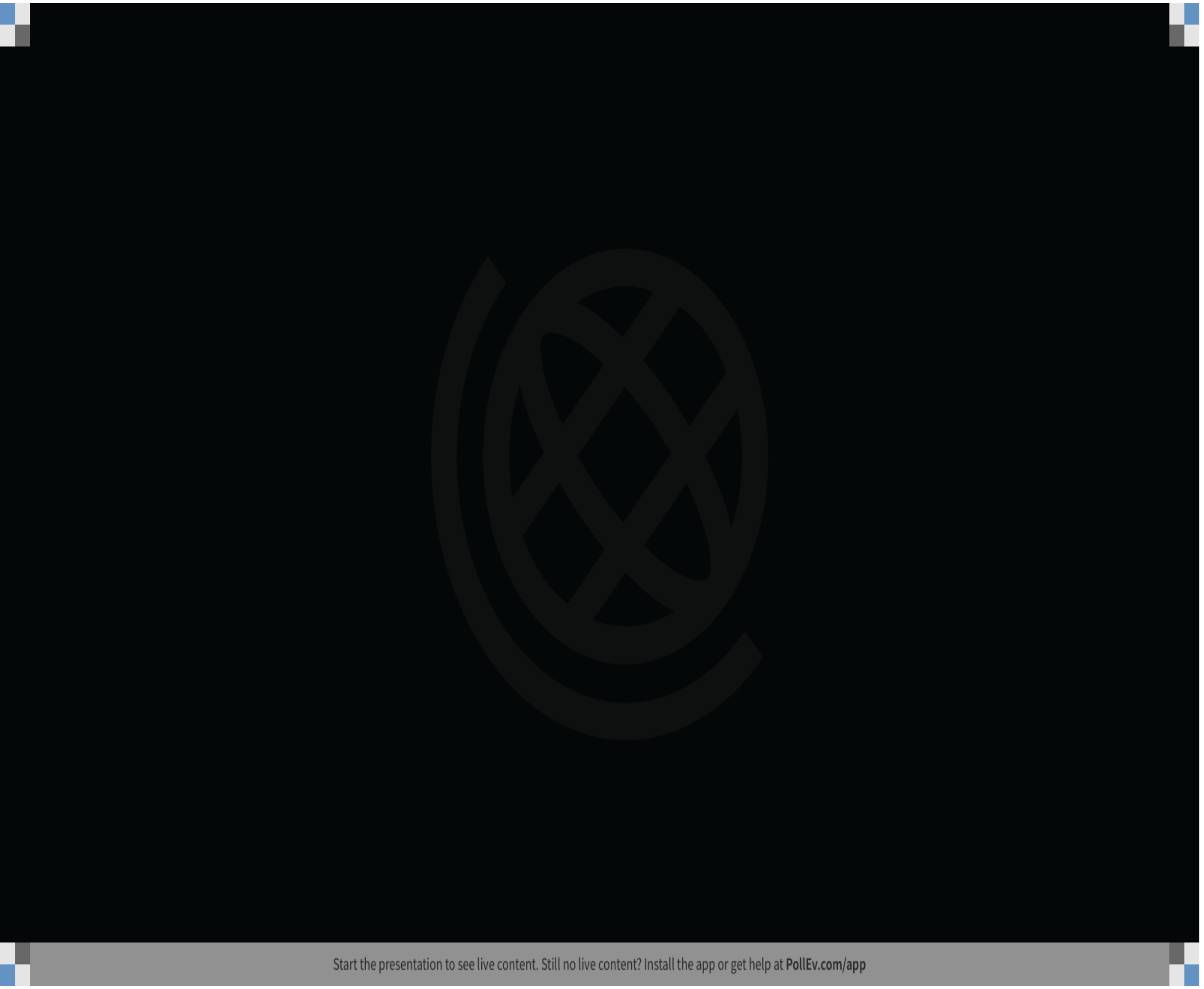
Which of the following is behaviour-based navigation?

-“Hi, I’m going to attend Leo’s lecture, do you know how to get to the theatre?”

-“Sure, you are now at the centre of Quadrangle Lawn, you just need to go south for 50 meters and then go west for 40 meters. You’ll be able to find the Webster Theatres. The lecture is on the second floor.”



Which of the following is behaviour-based navigation?

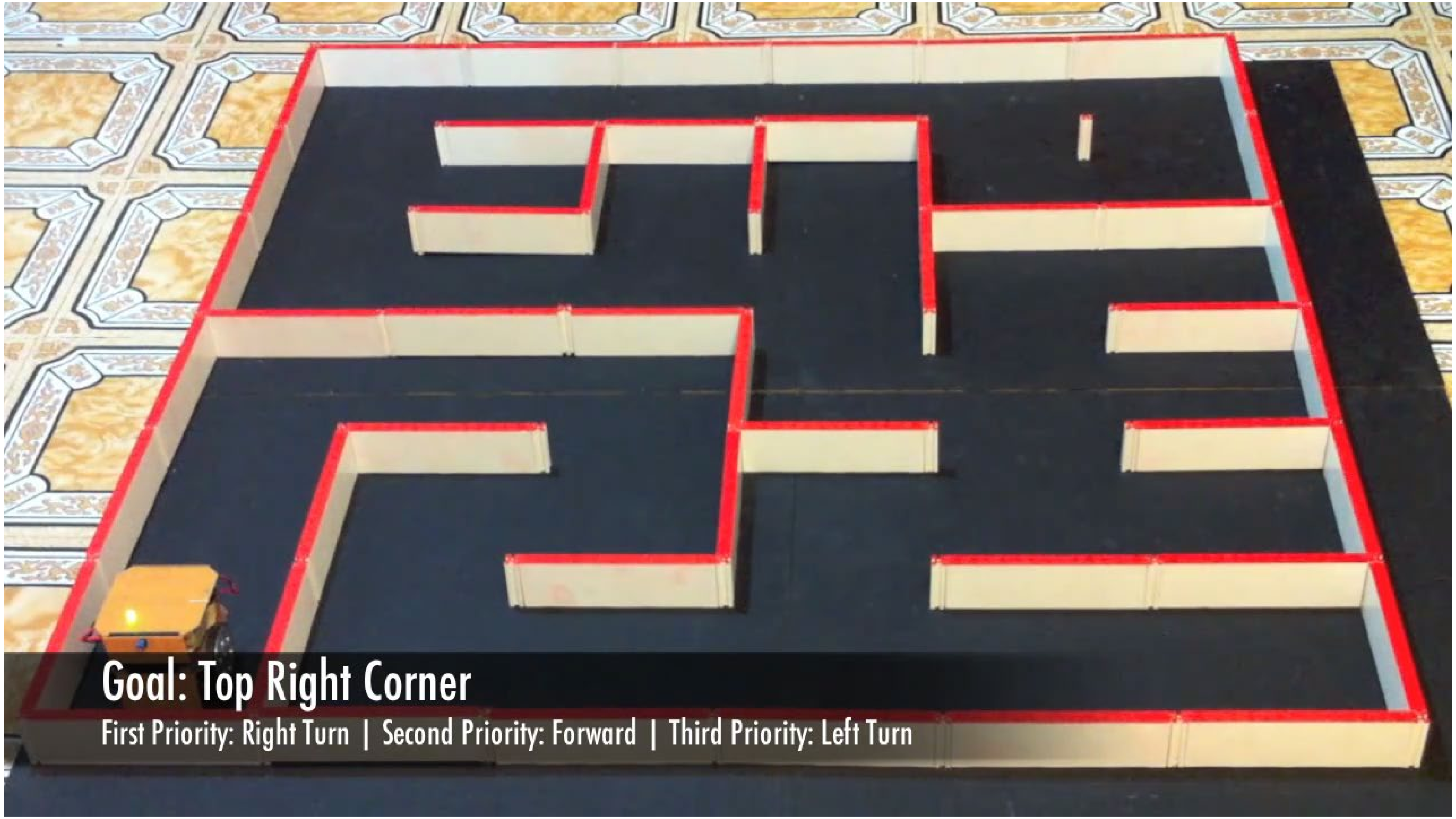


Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

-“Hi, the lecture is really boring. I’m gonna take a tram to join my friend’s party at the CBD, do you know where the nearest station is?”

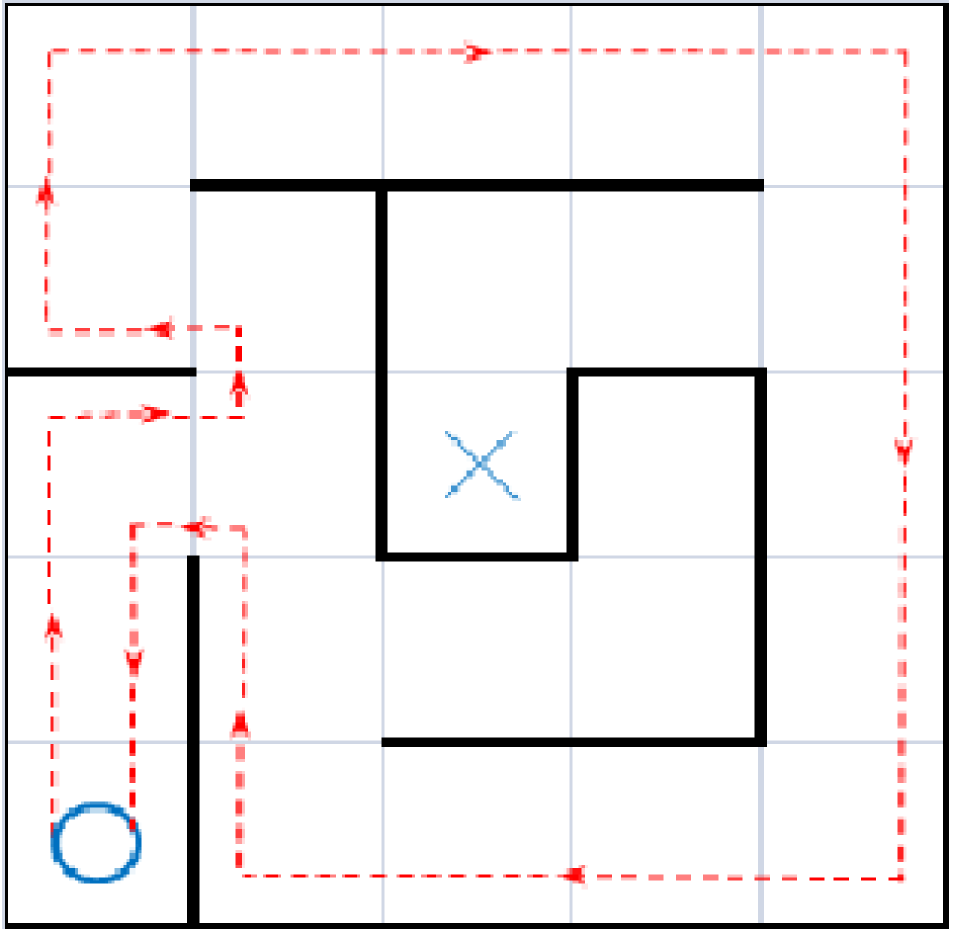
-“Easy, you just need to get out of the door and step down the stairs to the ground floor. Walk around the Robert Webster Building clockwise until you reach the University Mall Road, then follow it in the downward direction for about 5min until you cross the Anzac Parade. You should then be able to see the station on your right-hand side. ***But seriously, why taking a tram when a bus is much faster???***

Behaviour based approach - Right (or Left) Wall Follower


Goal: Top Right Corner
First Priority: Right Turn | Second Priority: Forward | Third Priority: Left Turn

https://www.youtube.com/watch?v=LB99PqelIMw

What about this one?



Saman, A.B., & Abdramane, I. (2013). Solving a Reconfigurable Maze using Hybrid Wall Follower Algorithm.

Map based navigation

When a robot comes into a new house

https://www.youtube.com/watch?v=MX2qcDM11N4

Behavior based approach vs. Map based approach

Behavior based approach

- Pros
  - Avoids inaccuracy of mapping
  - Easy to implement (if works)
- Cons
  - Does not directly scale to other environments or to larger environments
  - Must be carefully designed to produce the desired behaviour
  - May have multiple active behaviours at any one time

Map based approach

- Pros
  - Position available to human operators
  - The map, if created by the robot, can be used by humans as well
  - Ability to scale – changing maps
- Cons
  - More up-front development effort
  - May go diverging even if the raw sensor values are transiently incorrect

Map Representation

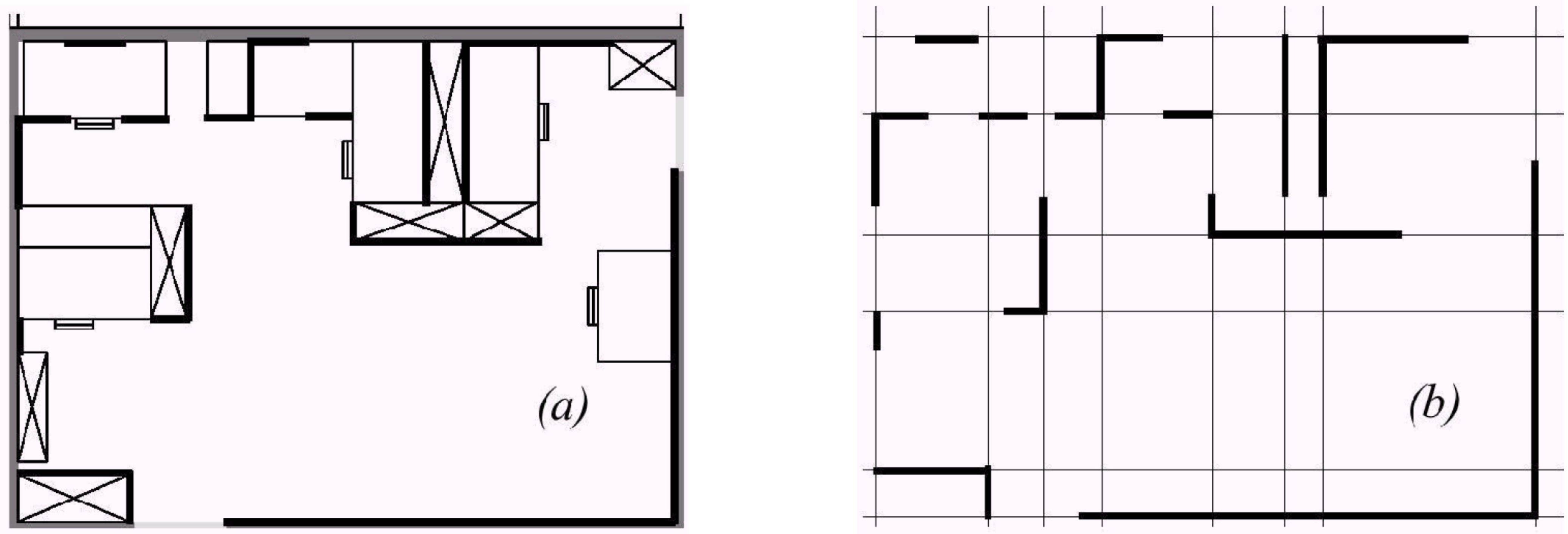
Map representation

- Continuous line-based
- Cell decomposition
  - Exact cell decomposition
  - Fixed cell decomposition
  - Adaptive cell decomposition
- Topological map

https://wallpaperaccess.com/marauders-map-harry-potter

Map representation – Continuous line-based

- Representation with set of finite or infinite lines
- *Closed-World Assumption* - Only need to store the information of the lines
  - *CWA: What is not currently known to be true, is false.*
  - *OWA: What is not currently known to be true, can be either true or false.*

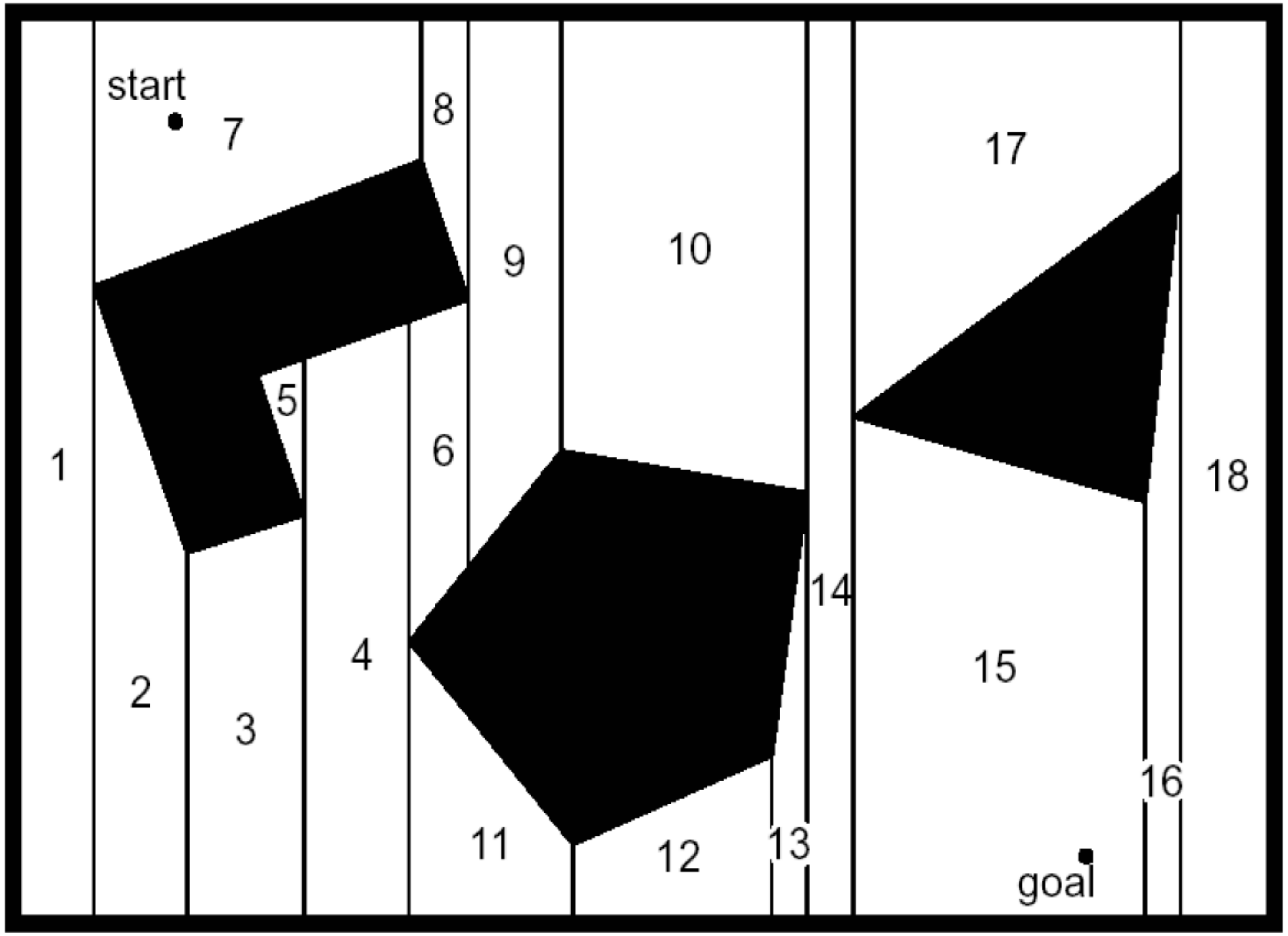


R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.
https://en.wikipedia.org/wiki/Closed-world_assumption

Map representation – Exact cell decomposition (Polygon)

- Pros:
  - Can be extremely compact

- Cons:
  - The information of the obstacles and free space may be expensive to collect

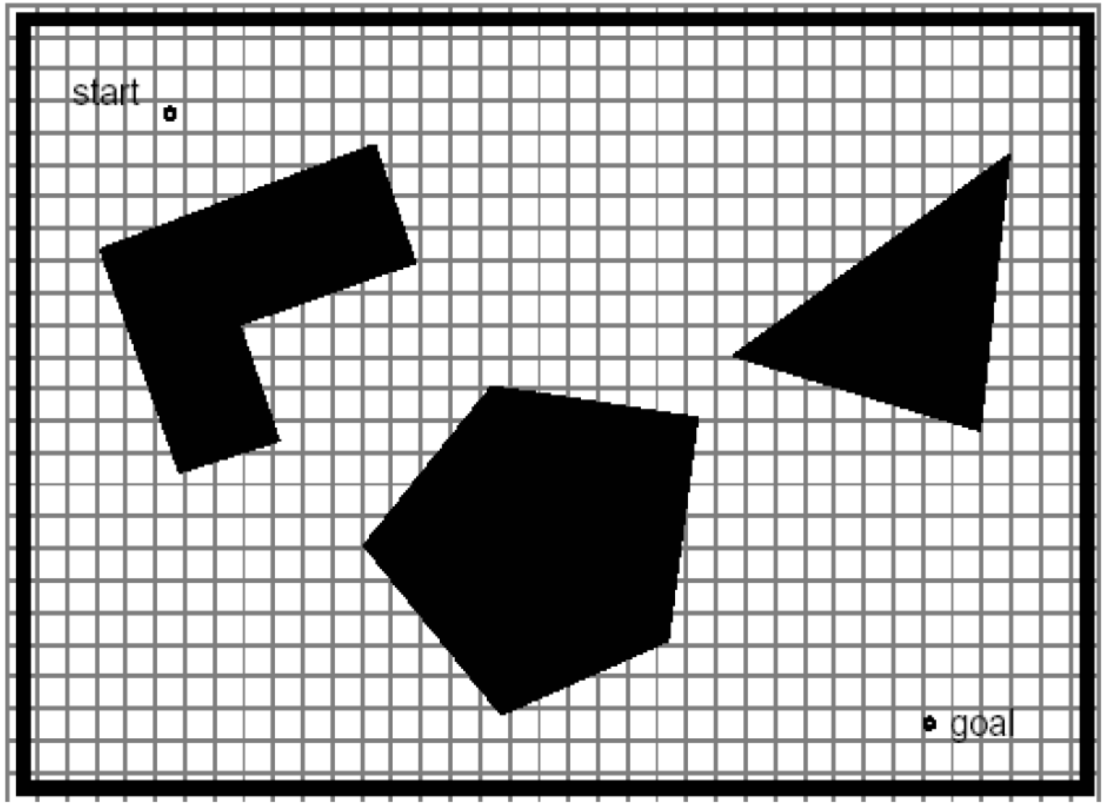


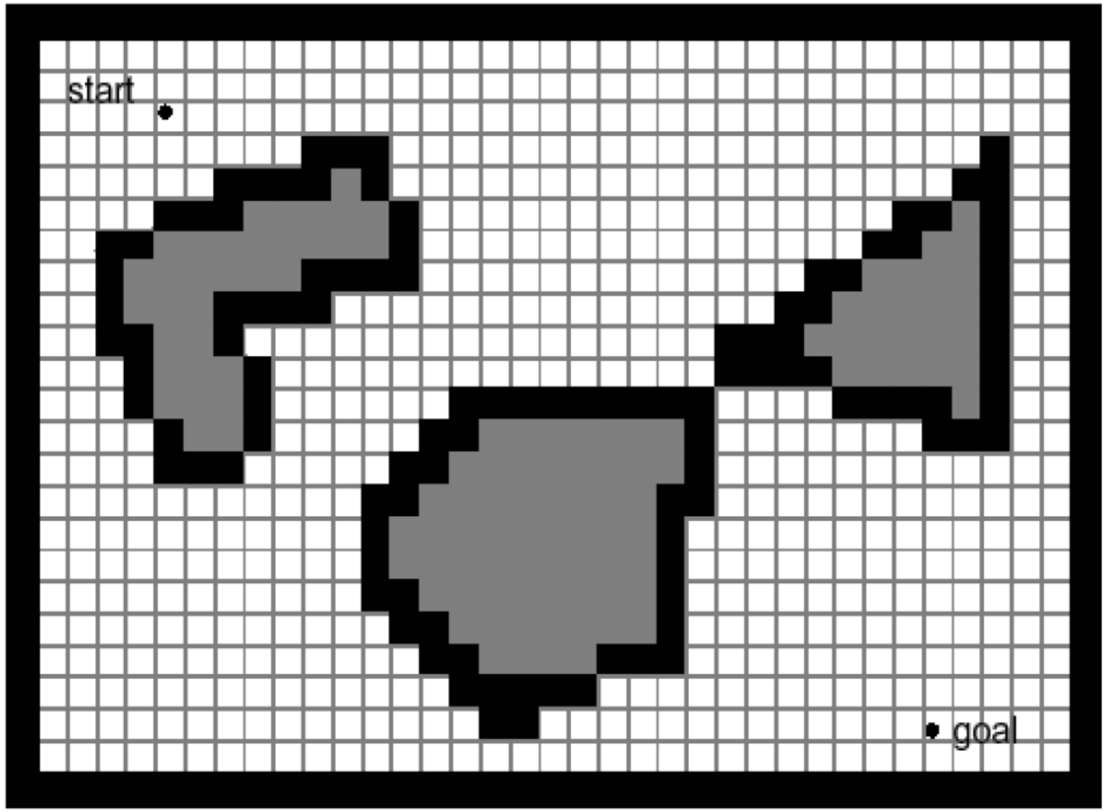
R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Map representation – Fixed cell decomposition (Occupancy grid)

- Pros:
  - Easy to implement for robots with range-based sensors

- Cons:
  - Narrow passages may disappear
  - Huge memory may be needed





R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

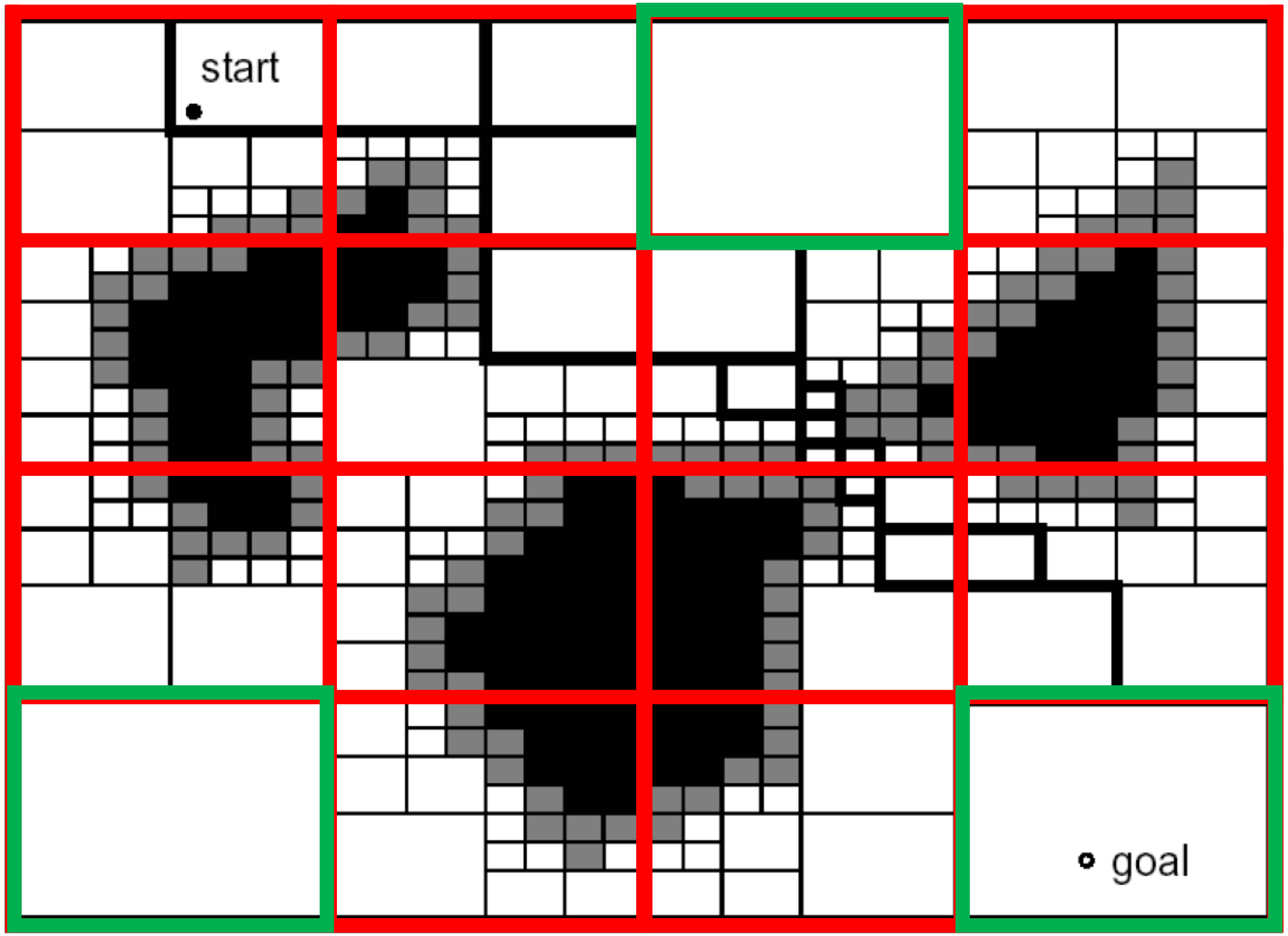
Map representation – Adaptive cell decomposition

Resolution = 1/4

Resolution = 1/16

⋮

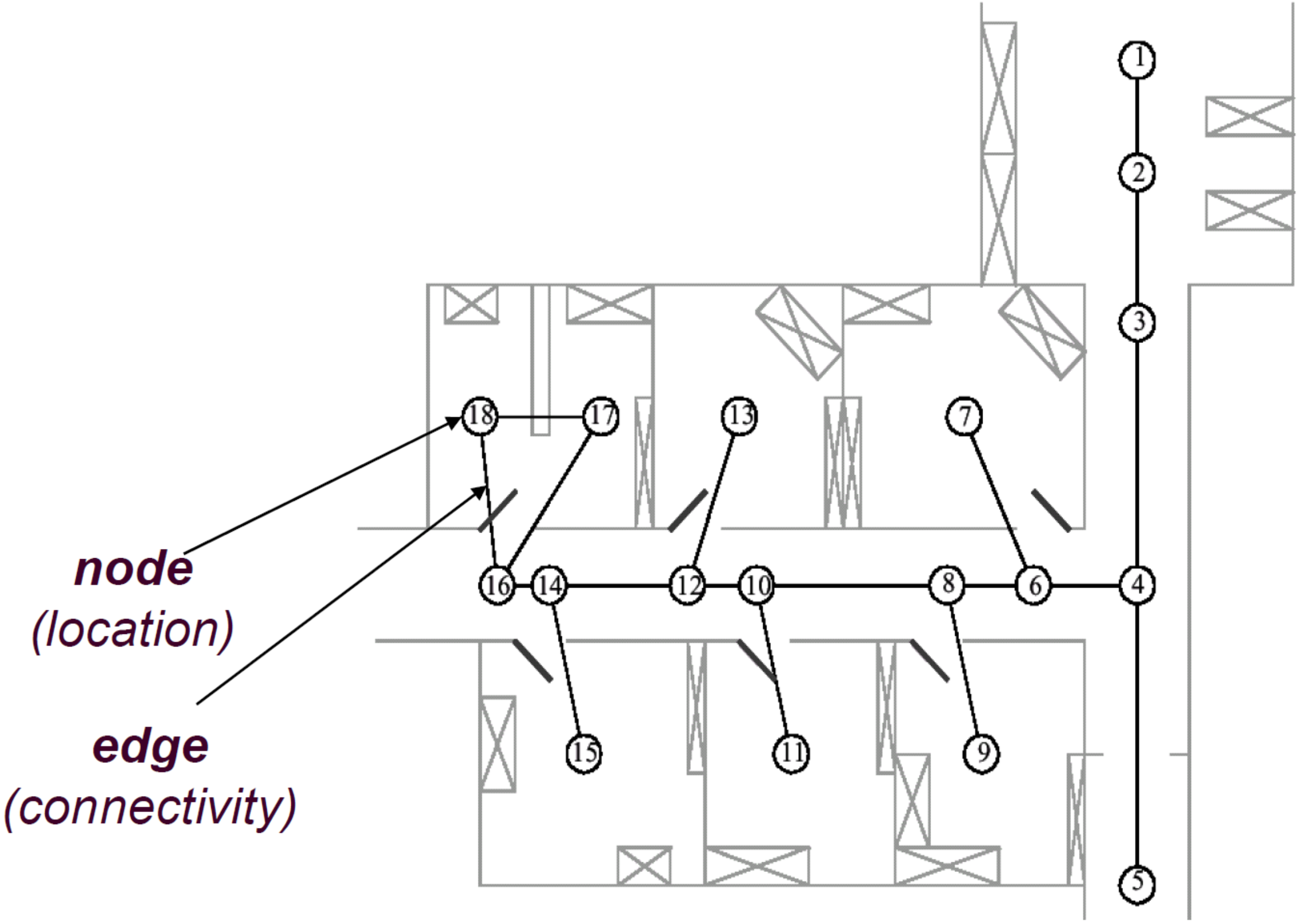
Resolution = Predefined



R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Map representation – Topological representation

- Represents environment with nodes and edges
- Lacks scale and distances
- Maintains topological relationships (connectivity)
- Adapts to geometric change



node
(location)

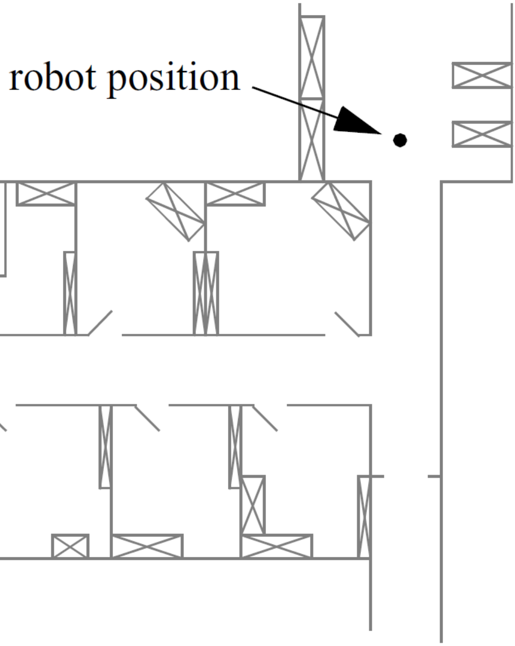
edge
(connectivity)

R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Map representation - Example

Continuous map? Occupancy grid map? Topological map?

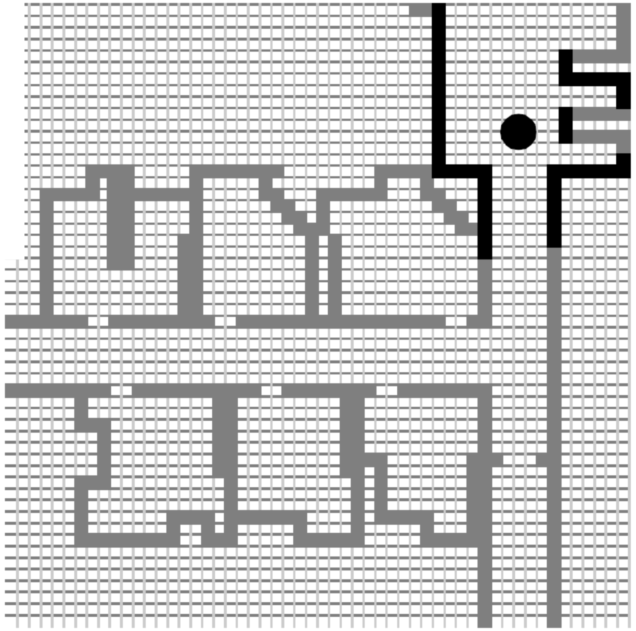
(1)



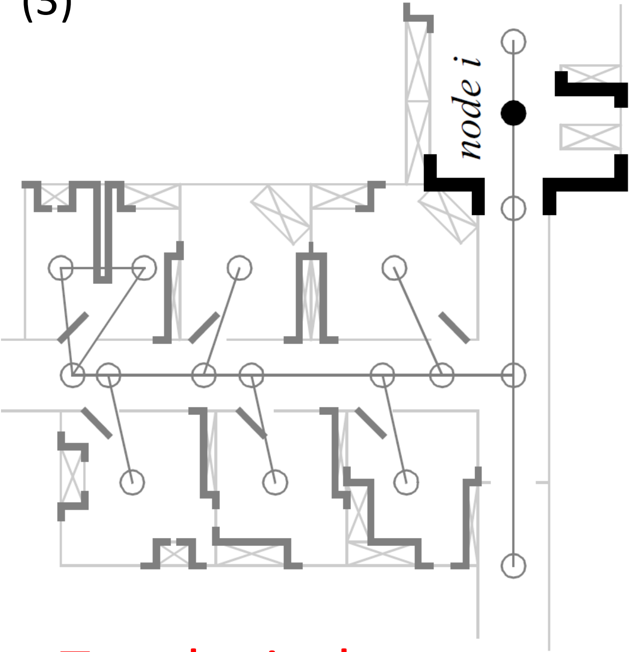
Real map

Occupancy grid map

(2)



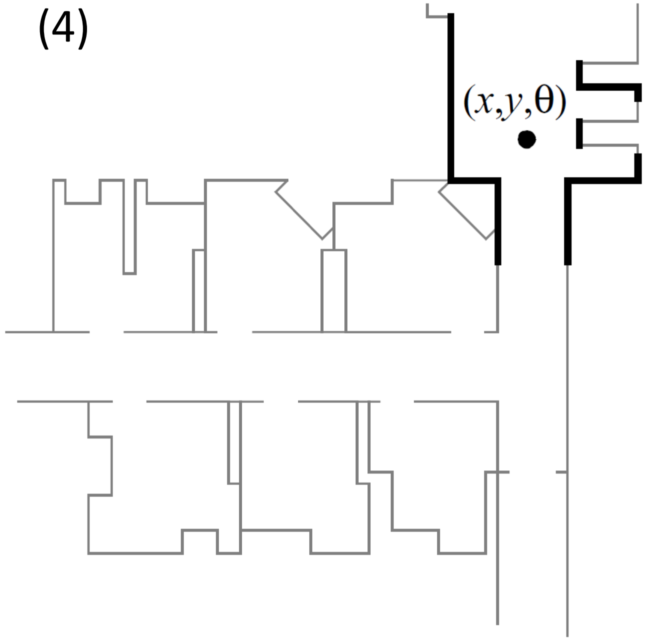
(3)



Topological map

Continuous map

(4)



R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

slido

Which map do you think is best suited to represent the maze?

ⓘ Start presenting to display the poll results on this slide.

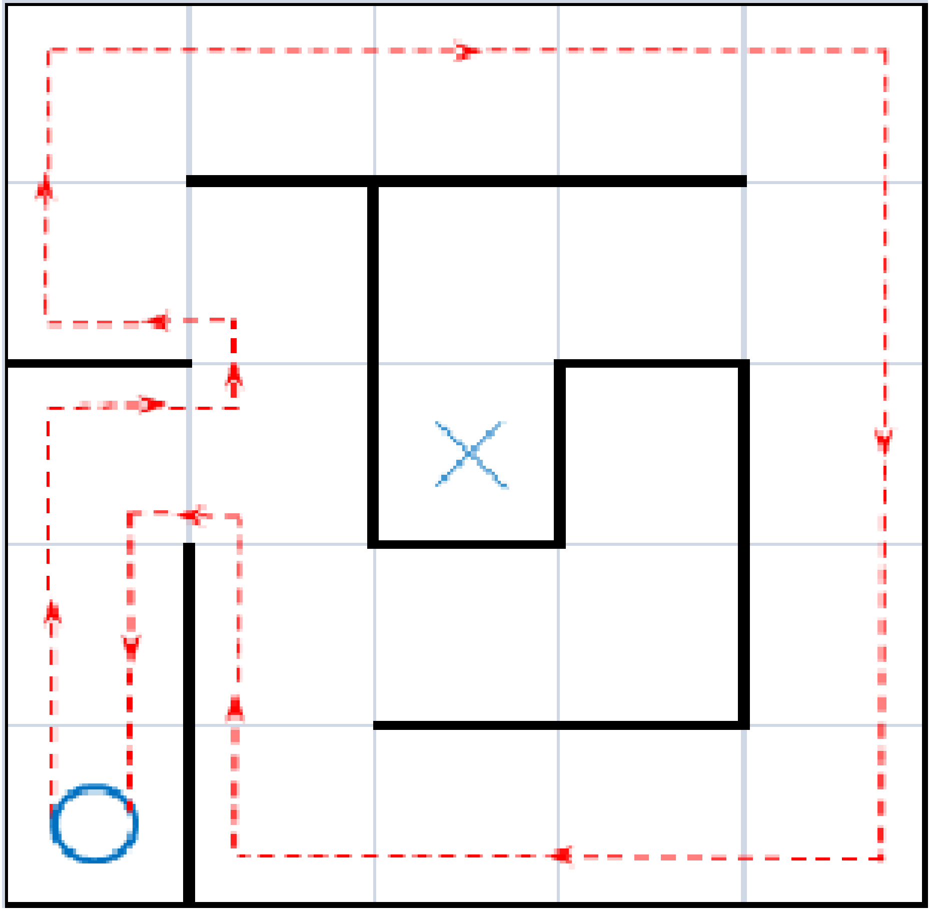
An example map for the maze

$$HorizontalWall = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}_{6\times5}$$

$$VerticalWall = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}_{5\times6}$$

$$CellValue = \begin{bmatrix} 8 & 7 & 6 & 5 & 4 \\ 9 & 10 & 1 & 2 & 3 \\ 12 & 11 & 0 & 13 & 4 \\ 13 & 10 & 11 & 12 & 5 \\ 14 & 9 & 8 & 7 & 6 \end{bmatrix}_{5\times5}$$

$RobotPos = (4{,}0)$ (assuming the top-left cell is (0,0))



Localisation Methods

Two types of localisation

- Global localisation
  - The robot is not told its initial position
  - Its position must be estimated from scratch

- Position tracking
  - A robot knows its initial position
  - It just needs to estimate the displacement relative to the initial position

Four localisation methods

- Localisation based on landmarks/artificial markers/external sensors
- Dead reckoning/Odometry
- Map based localisation – without external sensors/artificial landmarks, just use robot onboard sensors
  - Probabilistic map based localisation
- Simultaneous Localisation and Mapping (SLAM)

Four localisation methods

- Localisation based on landmarks/artificial markers/external sensors

- Dead reckoning/Odometry

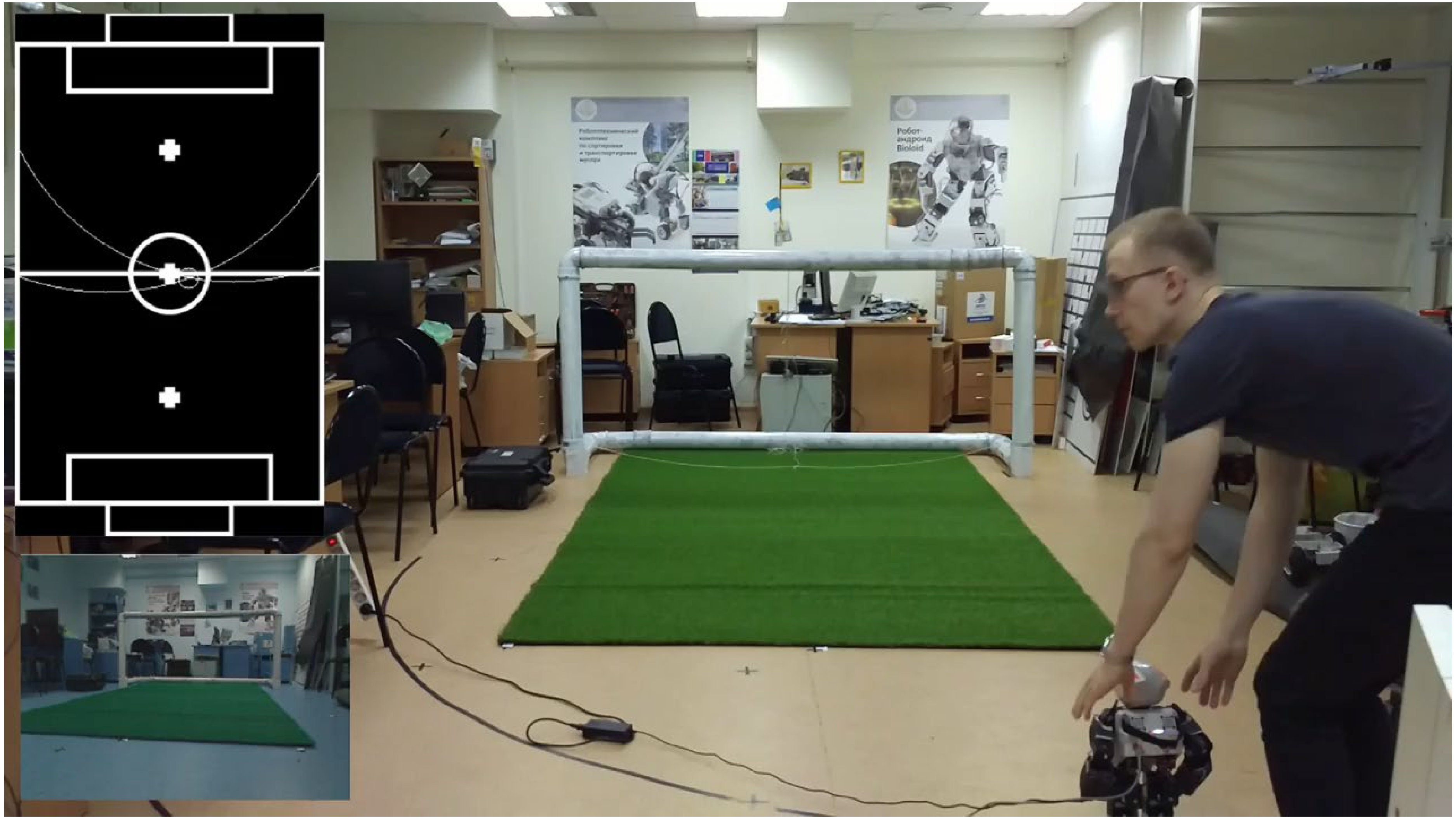
- Map based localisation – without external sensors/artificial landmarks, just use robot onboard sensors
  - Probabilistic map based localisation

- Simultaneous localisation and Mapping (SLAM)

Artificial-marker based localisation

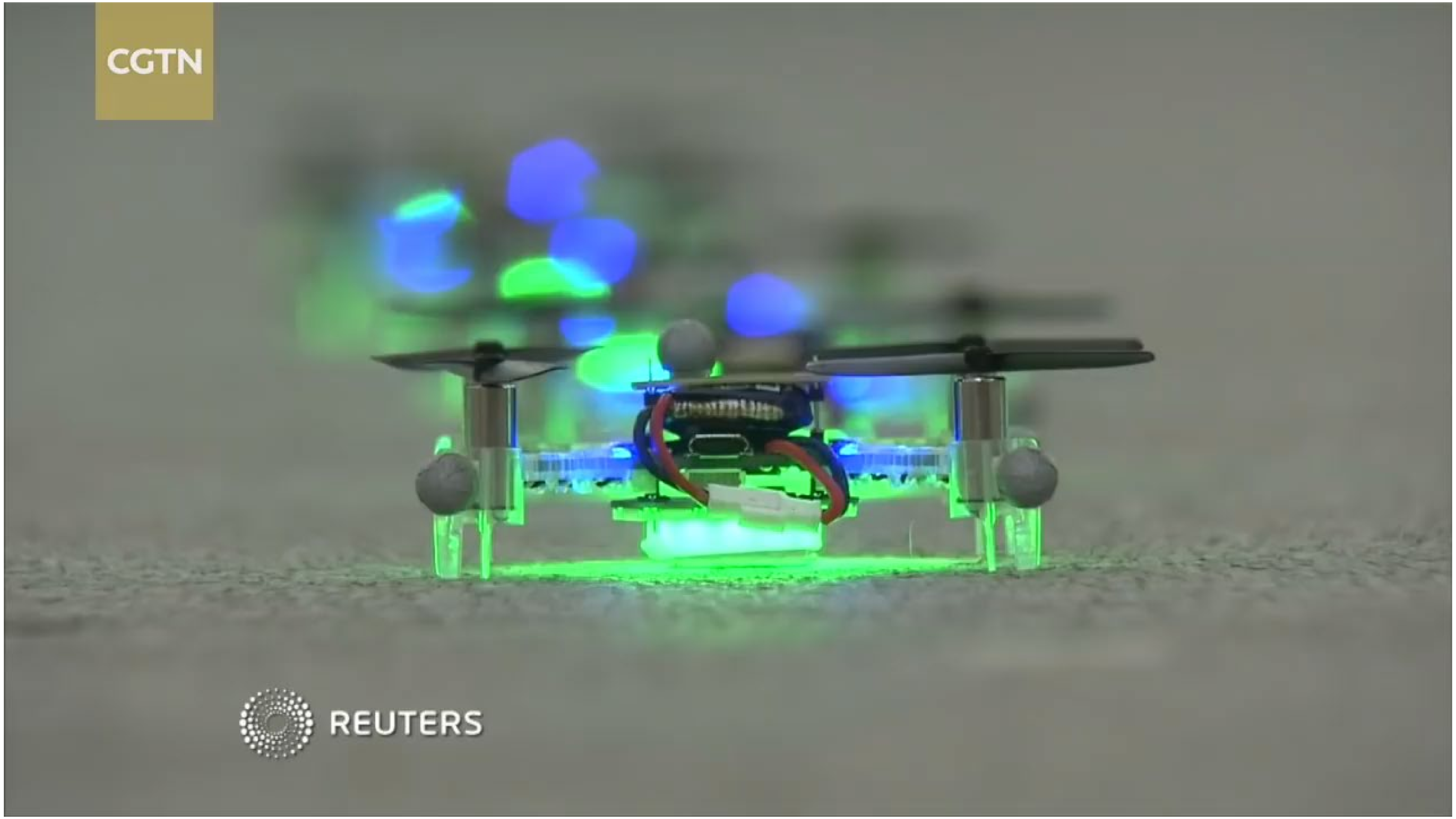
https://www.youtube.com/watch?v=I-n6fHfUHzA

Landmark based localisation



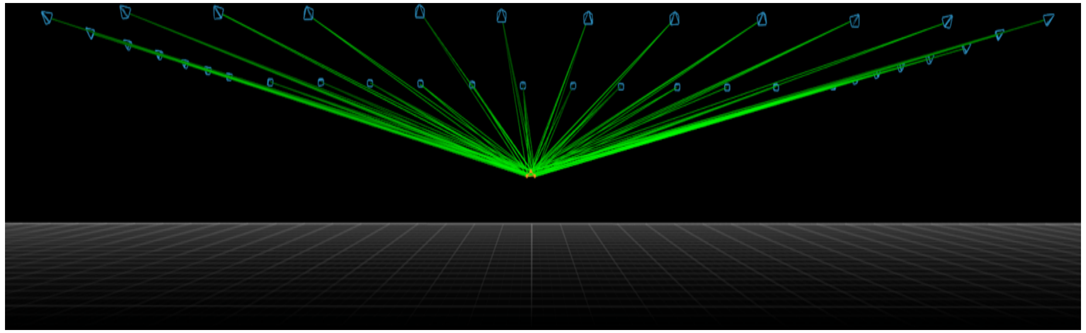
https://www.youtube.com/watch?v=xSdr16r1so8

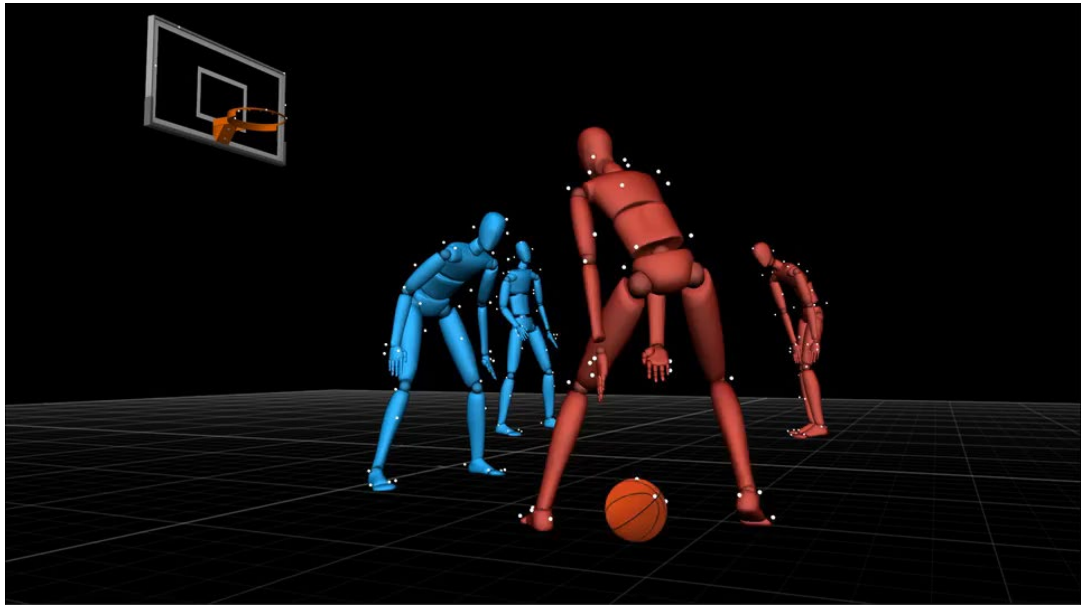
Motion-capture-system based localisation

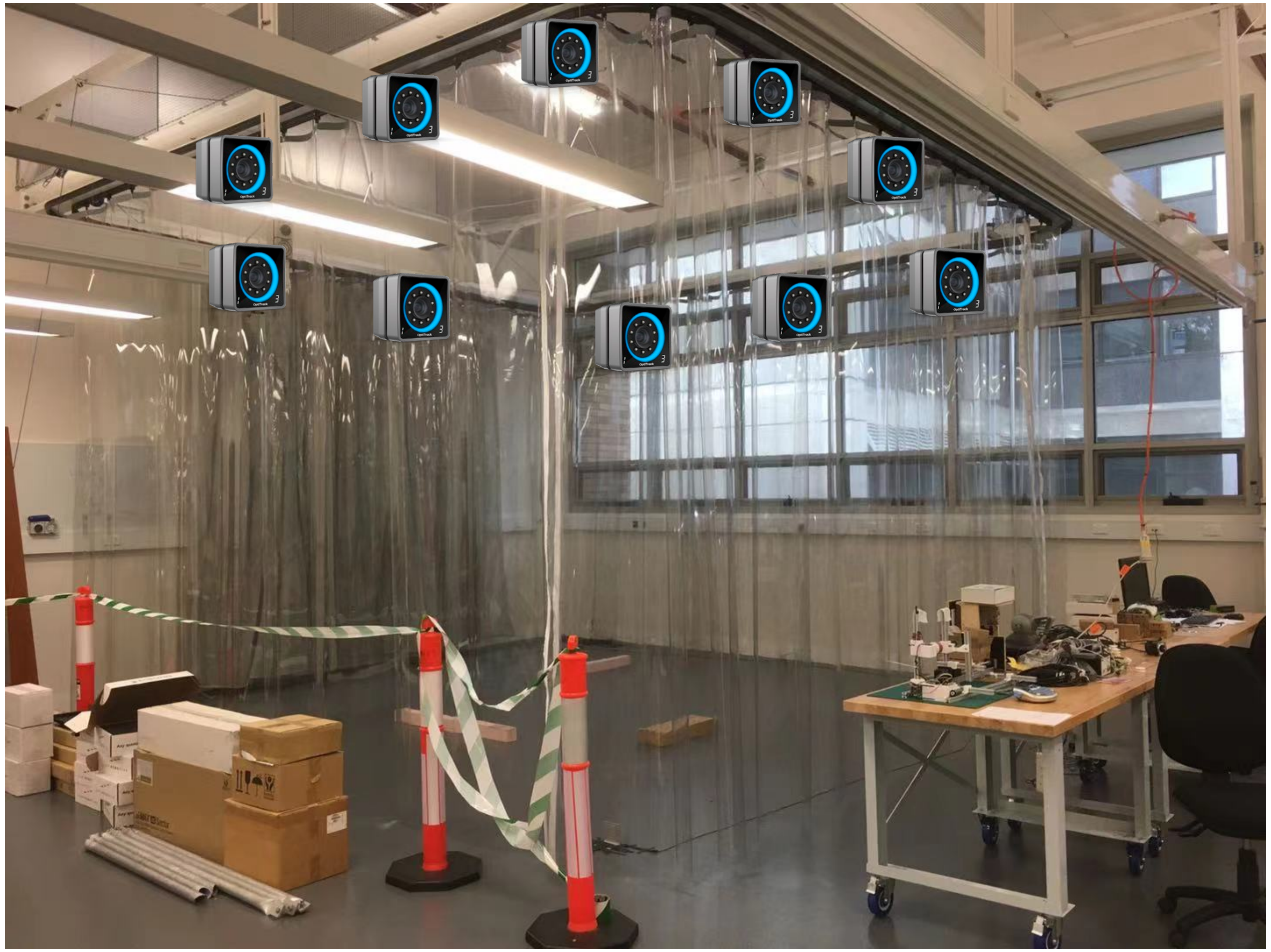


https://www.youtube.com/watch?v=MlFtHuXPbv4

Motion-capture-system at Mechatronics Research Lab, MME, UNSW







Cameras to be mounted soon! Room 204, Willis Annexe J18, UNSW

https://www.optitrack.com/

Four localisation methods

- Localisation based on landmarks/artificial markers/external sensors
  - Capable of global localisation
  - Needs modification or detailed information of the environment

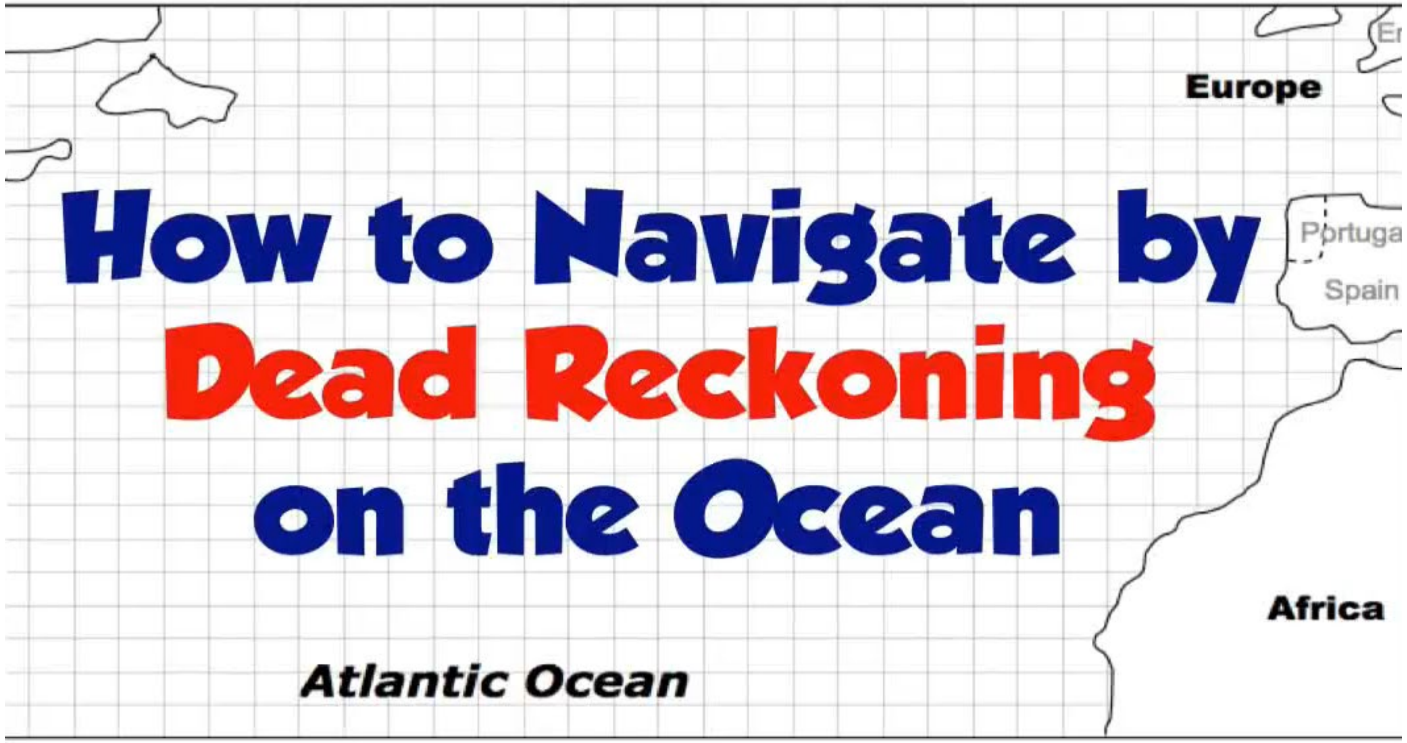
- Dead reckoning/Odometry

- Map based localisation – without external sensors/artificial landmarks, just use robot onboard sensors
  - Probabilistic map based localisation

- Simultaneous localisation and Mapping (SLAM)

Dead reckoning/Odometry

- Dead reckoning (Deduced reckoning)
  - A simple mathematical procedure for determining the present location of a vessel by advancing some previous position through known course and velocity information over a given length of time.
- Odometry
  - Dead reckoning by using only wheel encoders, sometimes interchangeable with Dead reckoning



https://www.youtube.com/watch?v=TxAdmbuu93I

Dead reckoning – Differential-drive

Rotation matrix from local frame to global frame

Current pose

Local velocity

Next pose

Sample interval

$$p(t+\Delta t) \approx p(t) + R\cdot\xi\cdot\Delta t$$

$$= \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + R\cdot\begin{bmatrix} \dfrac{r\cdot\omega_L\cdot\Delta t}{2}+\dfrac{r\cdot\omega_R\cdot\Delta t}{2} \\ 0 \\ -\dfrac{r\cdot\omega_L\cdot\Delta t}{2l}+\dfrac{r\cdot\omega_R\cdot\Delta t}{2l} \end{bmatrix}$$

$\Delta\theta_L = \omega_L\cdot\Delta t$

$\Delta\theta_R = \omega_R\cdot\Delta t$

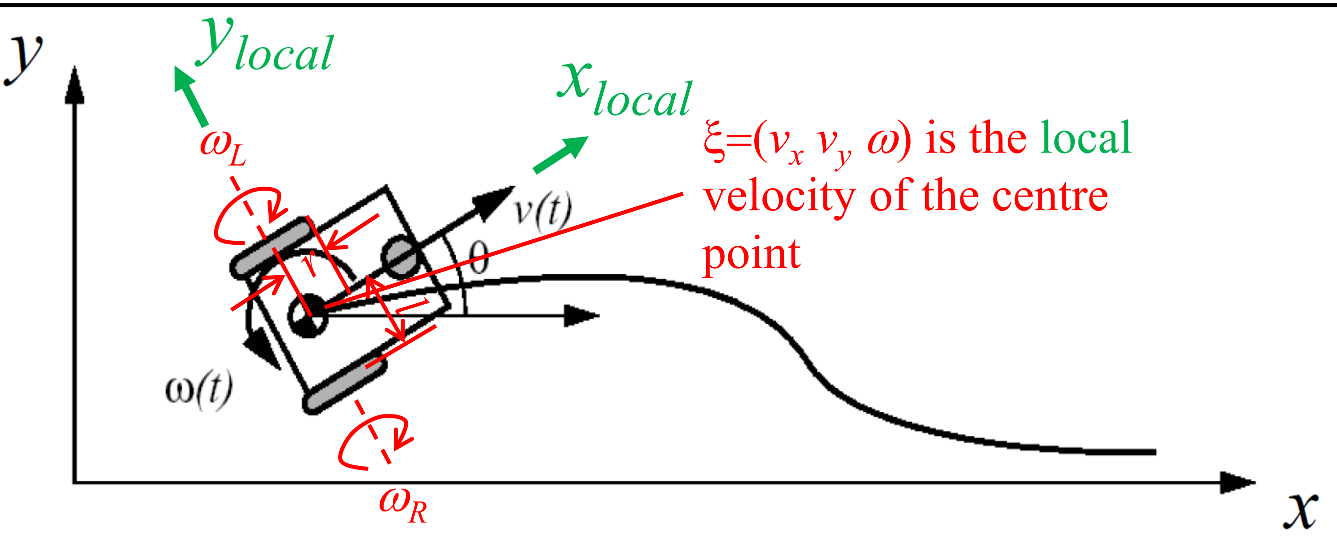
$$= \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} \dfrac{r\cdot\Delta\theta_L}{2}+\dfrac{r\cdot\Delta\theta_R}{2} \\ 0 \\ -\dfrac{r\cdot\Delta\theta_L}{2l}+\dfrac{r\cdot\Delta\theta_R}{2l} \end{bmatrix}$$

$$\Delta s \equiv \frac{r\cdot\Delta\theta_L}{2}+\frac{r\cdot\Delta\theta_R}{2}$$

$$\Delta\theta \equiv -\frac{r\cdot\Delta\theta_L}{2l}+\frac{r\cdot\Delta\theta_R}{2l}$$

$$\xi = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} \dfrac{r\cdot\omega_L}{2}+\dfrac{r\cdot\omega_R}{2} \\ 0 \\ -\dfrac{r\cdot\omega_L}{2l}+\dfrac{r\cdot\omega_R}{2l} \end{bmatrix}$$

$$= \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s\cdot\cos(\theta) \\ \Delta s\cdot\sin(\theta) \\ \Delta\theta \end{bmatrix} \approx \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta s\cdot\cos(\theta+\dfrac{\Delta\theta}{2}) \\ \Delta s\cdot\sin(\theta+\dfrac{\Delta\theta}{2}) \\ \Delta\theta \end{bmatrix}$$



$\xi=(v_x\ v_y\ \omega)$ is the local velocity of the centre point

R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Dead reckoning – Differential-drive

Increment

Current pose

Next pose

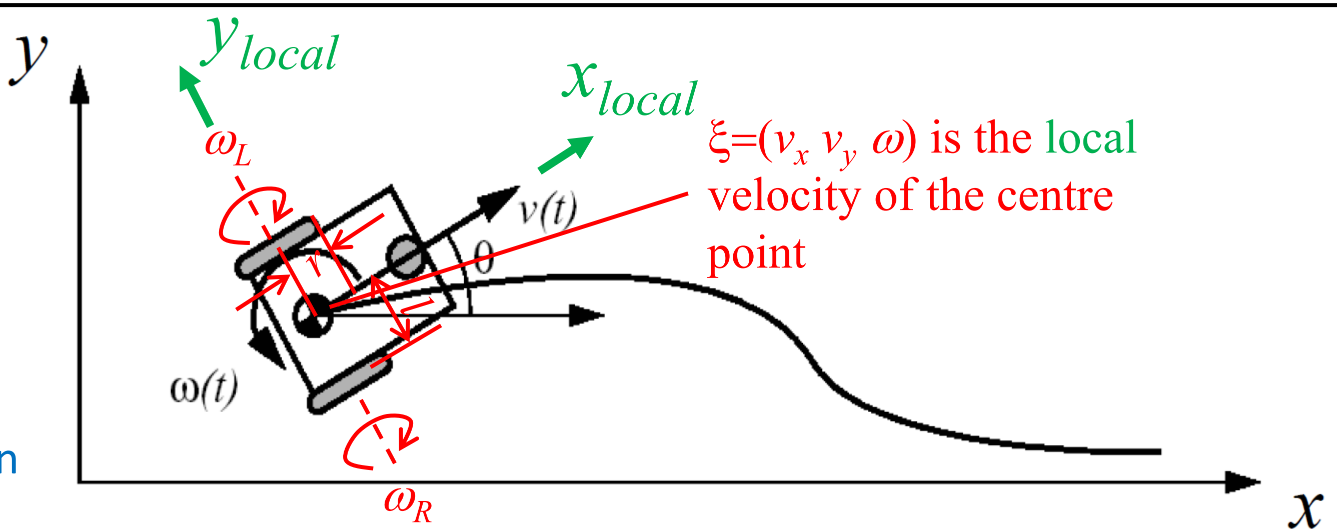
$$p(t+\Delta t) \approx p(t) + \begin{bmatrix} \Delta s \cdot \cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta s \cdot \sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{bmatrix}$$

$\Delta s \equiv \dfrac{r \cdot \Delta\theta_L}{2} + \dfrac{r \cdot \Delta\theta_R}{2}$ — Incremental linear motion

$\Delta\theta \equiv -\dfrac{r \cdot \Delta\theta_L}{2l} + \dfrac{r \cdot \Delta\theta_R}{2l}$ — Incremental rotation

$\Delta\theta_L = \omega_L \cdot \Delta t$ — Incremental rotation of left wheel

$\Delta\theta_R = \omega_R \cdot \Delta t$ — Incremental rotation of right wheel



$\xi=(v_x\ v_y\ \omega)$ is the local velocity of the centre point

Case 1: $\Delta\theta_L = \Delta\theta_R$ - Pure linear motion

Case 2: $\Delta\theta_L = -\Delta\theta_R$ - Pure rotation

R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Dead reckoning – Differential-drive: Example

Increment

Current pose

Next pose

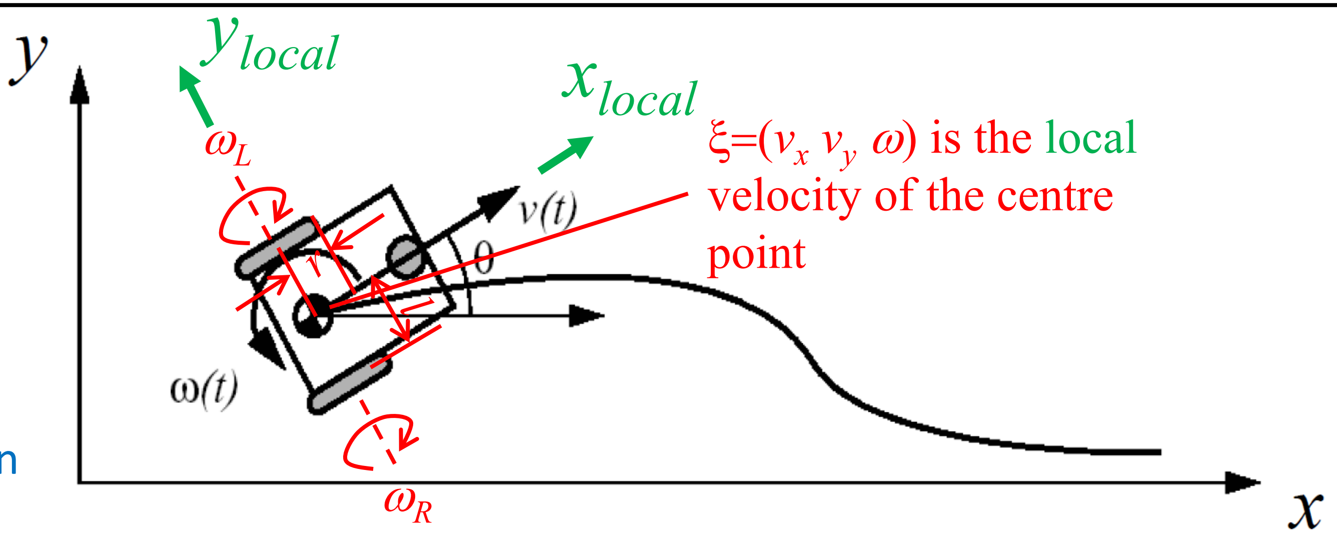
$$p(t+\Delta t) \approx p(t) + \begin{bmatrix} \Delta s \cdot \cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta s \cdot \sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{bmatrix}$$

$\Delta s \equiv \frac{r \cdot \Delta\theta_L}{2} + \frac{r \cdot \Delta\theta_R}{2}$ — Incremental linear motion

$\Delta\theta \equiv -\frac{r \cdot \Delta\theta_L}{2l} + \frac{r \cdot \Delta\theta_R}{2l}$ — Incremental rotation

$\Delta\theta_L = \omega_L \cdot \Delta t$ — Incremental rotation of left wheel

$\Delta\theta_R = \omega_R \cdot \Delta t$ — Incremental rotation of right wheel



$\xi=(v_x\ v_y\ \omega)$ is the local velocity of the centre point

Q: Suppose a differential-drive robot is running at a constant speed. The wheels have diameter 40mm and spaced at 100mm. The encoders of two wheels are read twice. The differences from the first to the second reading are 30deg and 60deg for the left and right wheels, respectively. Assume at the first reading, the robot's pos is (0mm, 0mm, 0deg). What is the robot's pose at the second reading? ($\pi = 3.14$)

R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Dead reckoning – Differential-drive: Example

Increment

Current pose

Next pose

$$p(t+\Delta t) \approx p(t) + \begin{bmatrix} \Delta s \cdot \cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta s \cdot \sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{bmatrix}$$

$$\Delta s \equiv \frac{r \cdot \Delta\theta_L}{2} + \frac{r \cdot \Delta\theta_R}{2}$$

$$\Delta\theta \equiv -\frac{r \cdot \Delta\theta_L}{2l} + \frac{r \cdot \Delta\theta_R}{2l}$$

$$\Delta\theta_L = \omega_L \cdot \Delta t$$

$$\Delta\theta_R = \omega_R \cdot \Delta t$$

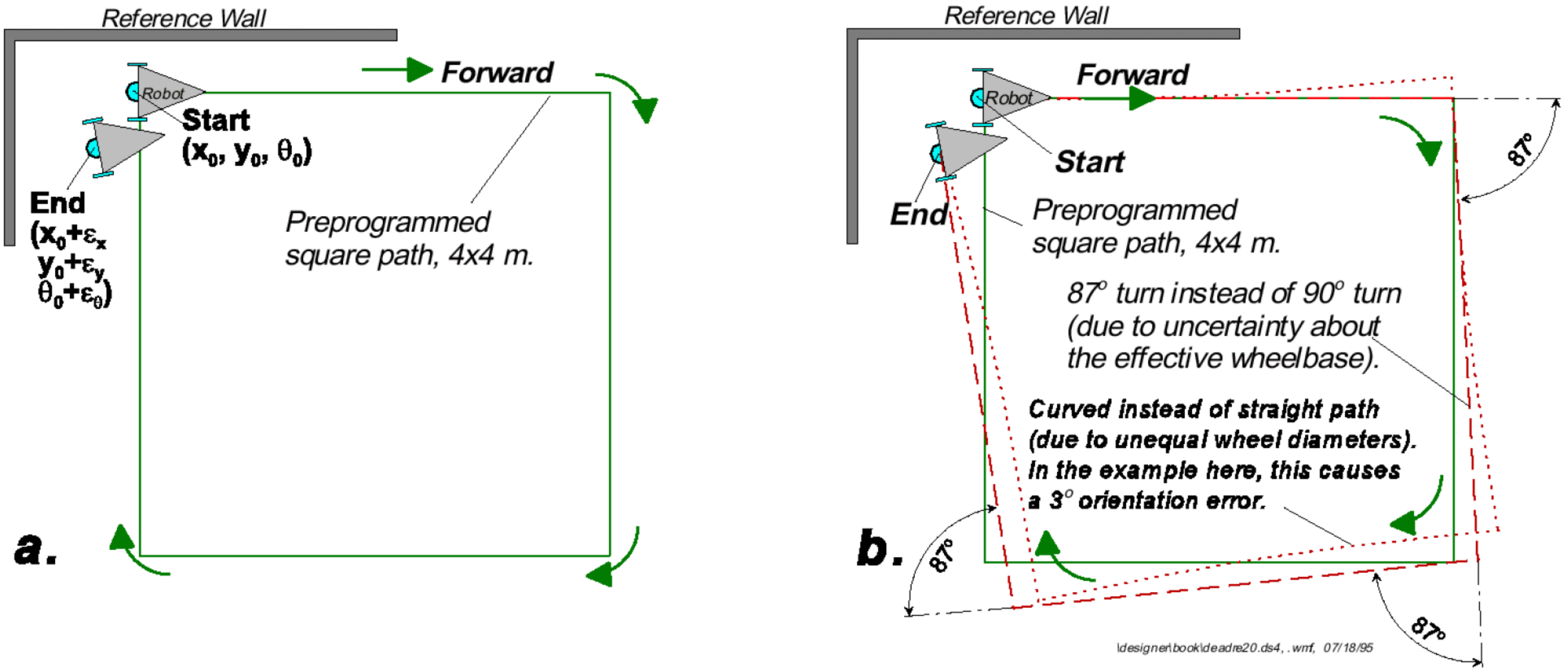
Q: Suppose a differential-drive robot is running at a constant speed. The wheels have diameter 40mm and spaced at 100mm. The encoders of two wheels are read twice. The differences from the first to the second reading are 30deg and 60deg for the left and right wheels, respectively. Assume at the first reading, the robot's pos is (0mm, 0mm, 0deg). What is the robot's pose at the second reading? ($\pi = 3.14$)

Homework:

Solve this problem and write a program in MATLAB (or any other language) for the calculation.

Hints: The solution is: (15.7mm, 0.83mm, 0.105rad).

Dead reckoning – Square path experiment



87° turn instead of 90° turn (due to uncertainty about the effective wheelbase).

Curved instead of straight path (due to unequal wheel diameters). In the example here, this causes a 3° orientation error.

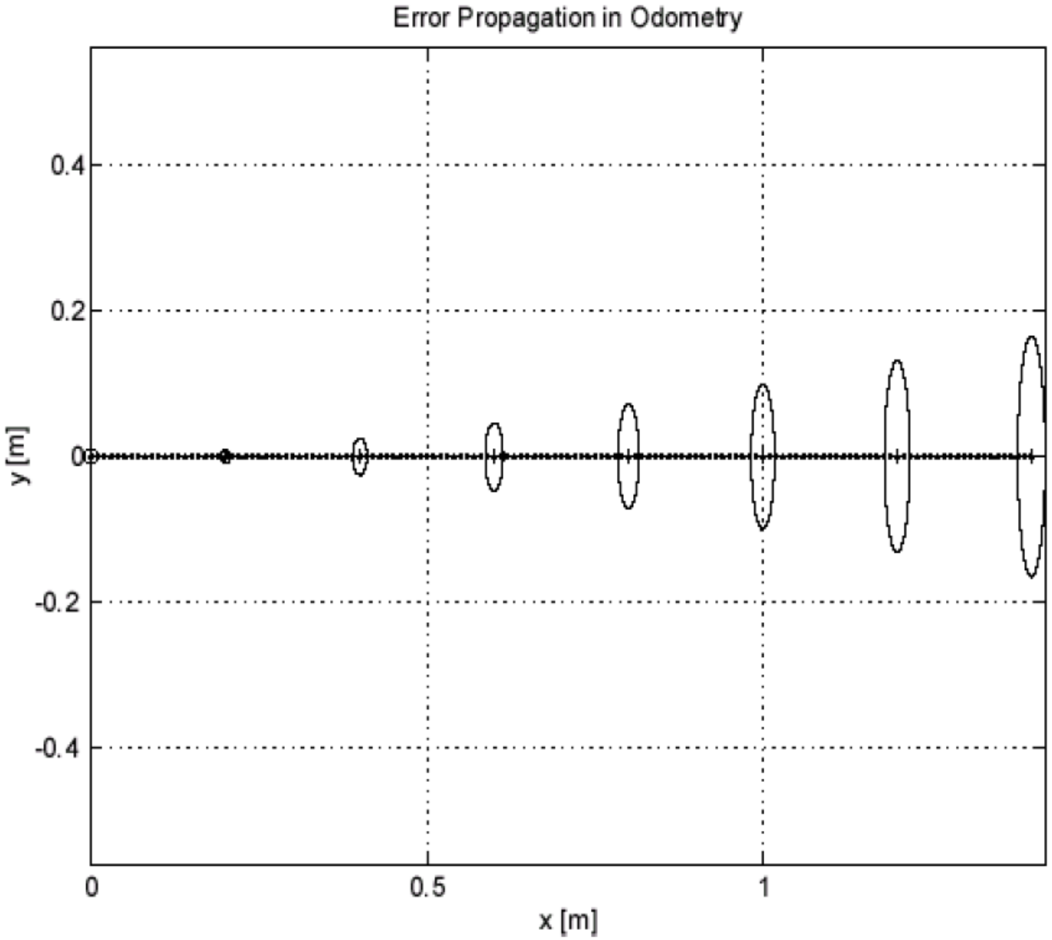
R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Dead reckoning – Error sources

Deterministic (Systematic)	Non-Deterministic (Non-Systematic)
• Can be reduced/eliminated by proper calibration of the system	• Are random errors, have to be described by error models, and will always lead to uncertain position estimate
• Examples • Misalignment of the wheels • Unequal wheel diameter	• Examples • Variation in the contact point of the wheel • Unequal floor contact (slippage, non-planar, etc.)

Dead reckoning – Growth of pose uncertainty for straight line movement

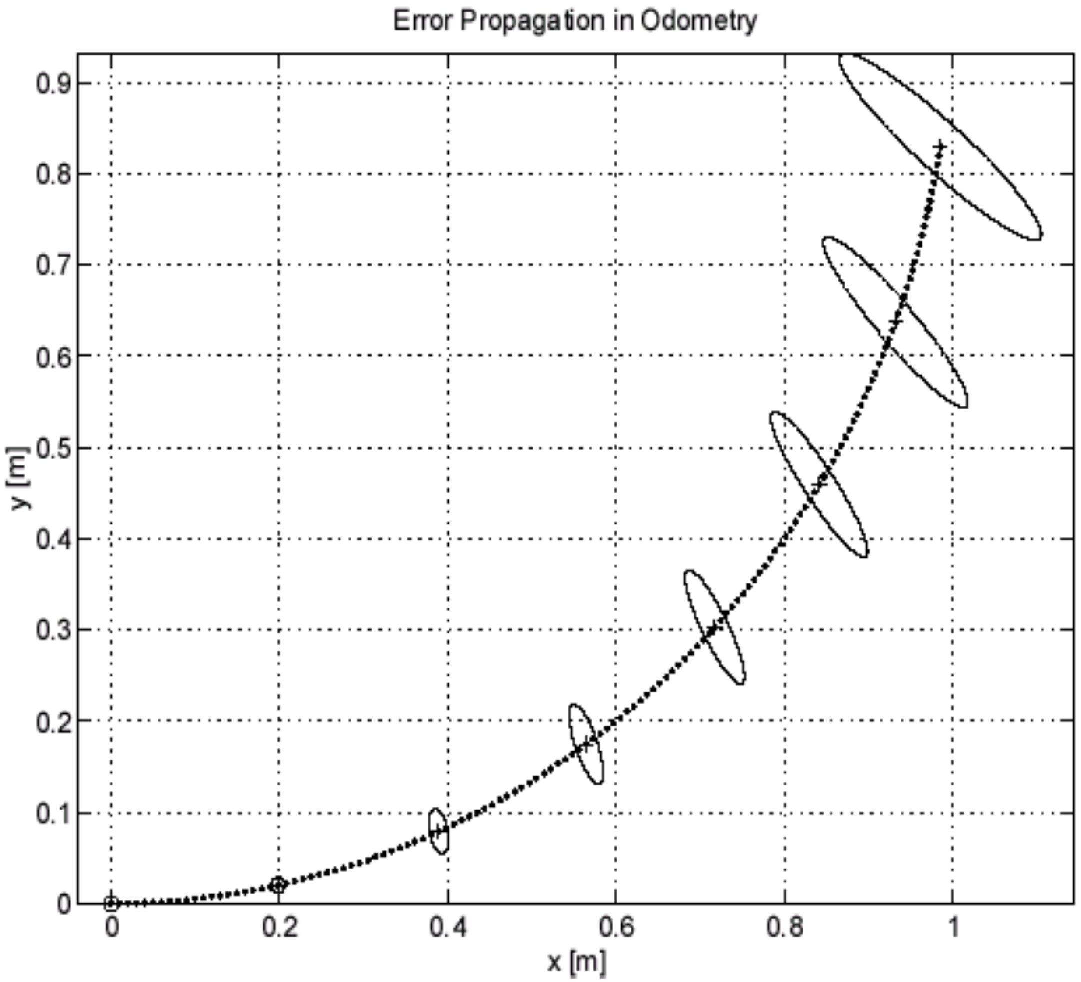
- Errors perpendicular to the direction of movement are growing much faster!



R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Dead reckoning – Growth of pose uncertainty for a circular movement

- Errors ellipse does NOT remain perpendicular to the direction of movement



R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.

Calibration of the robot parameters

Increment

Current pose

Next pose

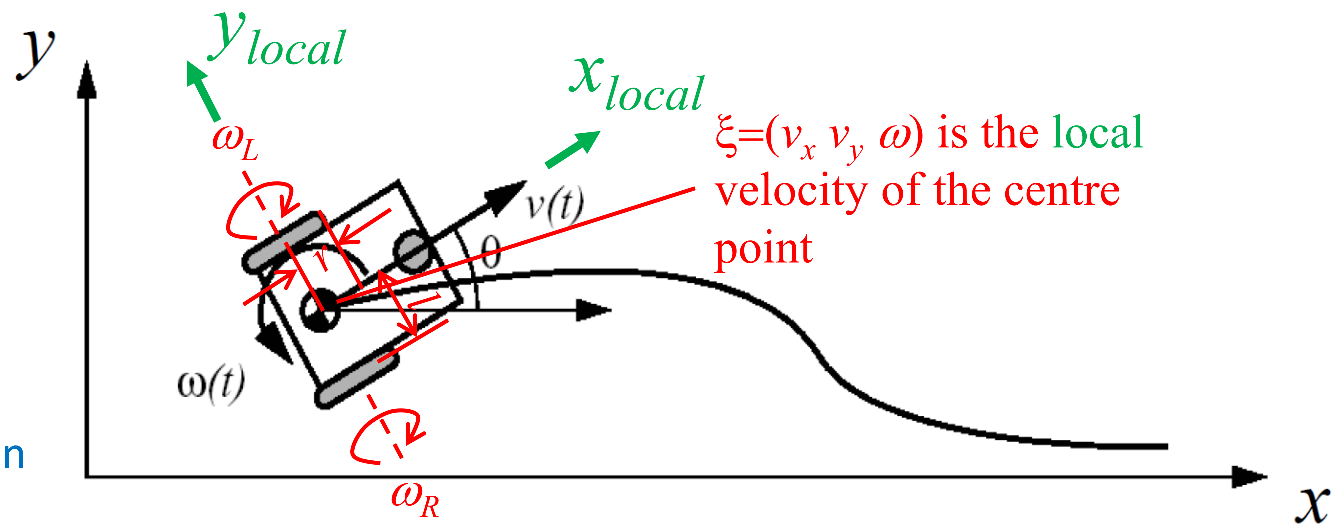
$$p(t+\Delta t) \approx p(t) + \begin{bmatrix} \Delta s \cdot \cos(\theta + \frac{\Delta\theta}{2}) \\ \Delta s \cdot \sin(\theta + \frac{\Delta\theta}{2}) \\ \Delta\theta \end{bmatrix}$$

$\Delta s \equiv \frac{r \cdot \Delta\theta_L}{2} + \frac{r \cdot \Delta\theta_R}{2}$ — Incremental linear motion

$\Delta\theta \equiv -\frac{r \cdot \Delta\theta_L}{2l} + \frac{r \cdot \Delta\theta_R}{2l}$ — Incremental rotation

$\Delta\theta_L = \omega_L \cdot \Delta t$ — Incremental rotation of left wheel

$\Delta\theta_R = \omega_R \cdot \Delta t$ — Incremental rotation of right wheel

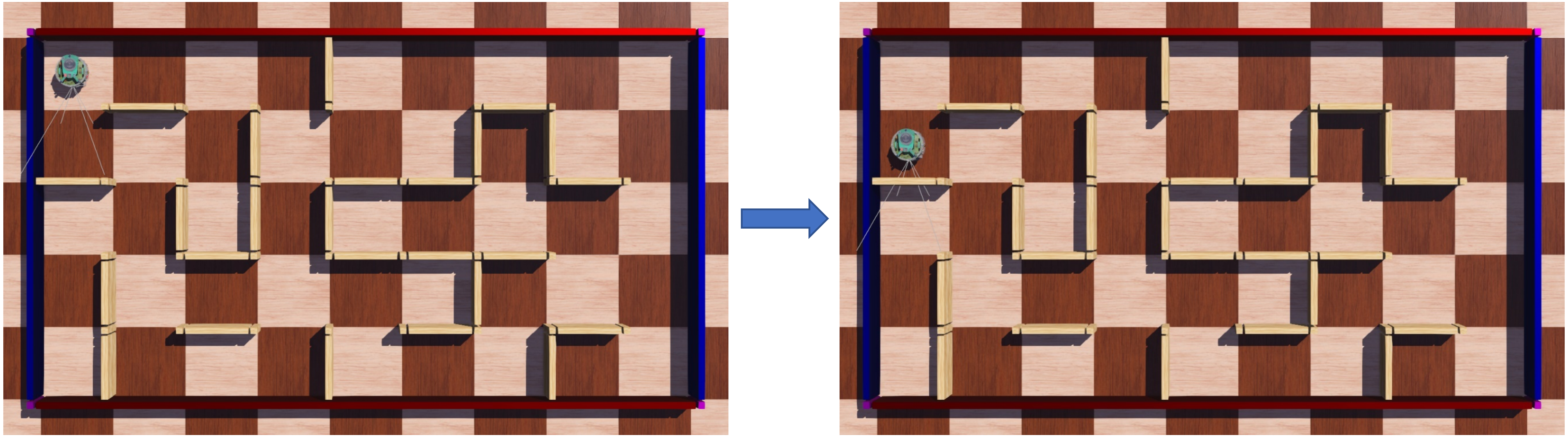


ξ=(v_x v_y ω) is the local velocity of the centre point

Case 1: $\Delta\theta_L = \Delta\theta_R$ - Pure linear motion

Case 2: $\Delta\theta_L = -\Delta\theta_R$ - Pure rotation

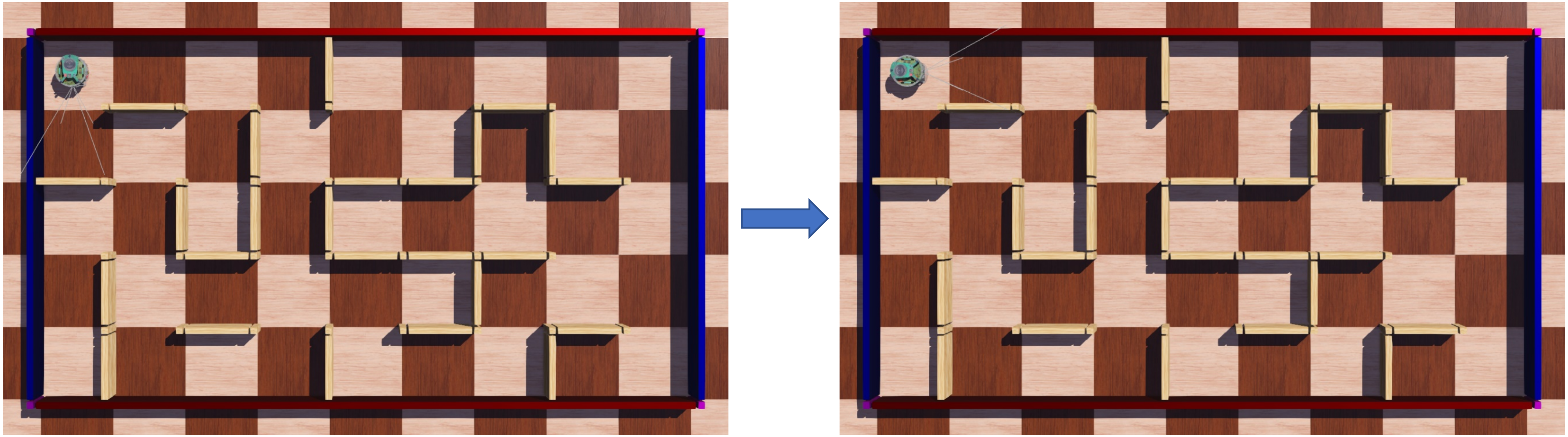
Calibration of the robot parameters – Wheel radius



1. Make $\Delta\theta_L = \Delta\theta_R = \emptyset$: pure translation
2. Tune this value $\emptyset$, i.e., the rotation angles of both motors until the robot moves to the centre of next cell
3. Calculate r from $\Delta s \equiv \frac{r \cdot \Delta\theta_L}{2} + \frac{r \cdot \Delta\theta_R}{2}$

- Note 1 – It may make the calibration more accurate by moving the robot for a longer distance, e.g., 10 cells (you can put the robot outside the maze to avoid wall collision)
- Note 2 – The recommended corrections to the parameters were obtained from this calibration process; you can also calibrate the robot to get your own corrections if interested

Calibration of the robot parameters – Axle length



1. Make $\Delta\theta_L = -\Delta\theta_R = \emptyset$: pure rotation
2. Tune this value $\emptyset$, i.e., the rotation angles of both motors until the robot rotates to a certain angle, e.g., 360deg
3. Calculate l from $\Delta\theta \equiv -\frac{r\cdot\Delta\theta_L}{2l} + \frac{r\cdot\Delta\theta_R}{2l}$ and the calibrated r

- Note 1 – It may make the calibration more accurate by rotating the robot more turns, e.g., 10 turns
- Note 2 – The recommended corrections to the parameters are obtained from this calibration process; you can also calibrate the robot to get your own corrections if interested

Four localisation methods

- Localisation based on landmarks/artificial markers/external sensors
  - Capable for global localisation
  - Needs modification or detailed information of the environment

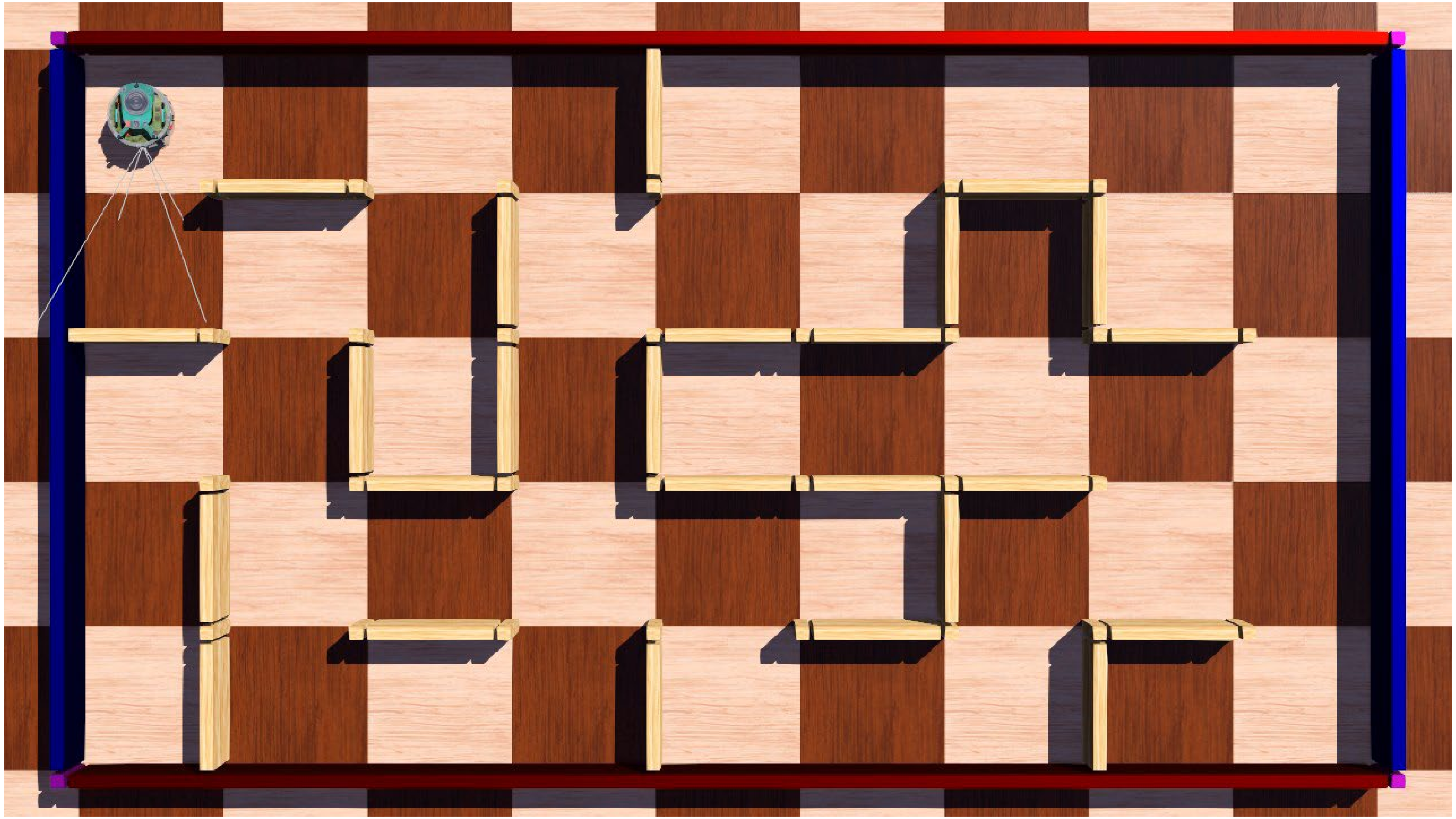
- Dead reckoning/Odometry
  - Only suitable for position tracking
  - Subject to deterministic and non-deterministic errors
  - Error may accumulate over time
  - Heading sensors (e.g. gyroscope) may help reduce the accumulated errors
    - $\Delta\theta$ measured by heading sensors instead of estimated by odometry

- Map based localisation – without external sensors/artificial landmarks, just use robot onboard sensors
  - Probabilistic map based localisation

- Simultaneous localisation and Mapping (SLAM)

What methods/sensors can be used here for localisation?

https://www.sli.do/
#4110



slido

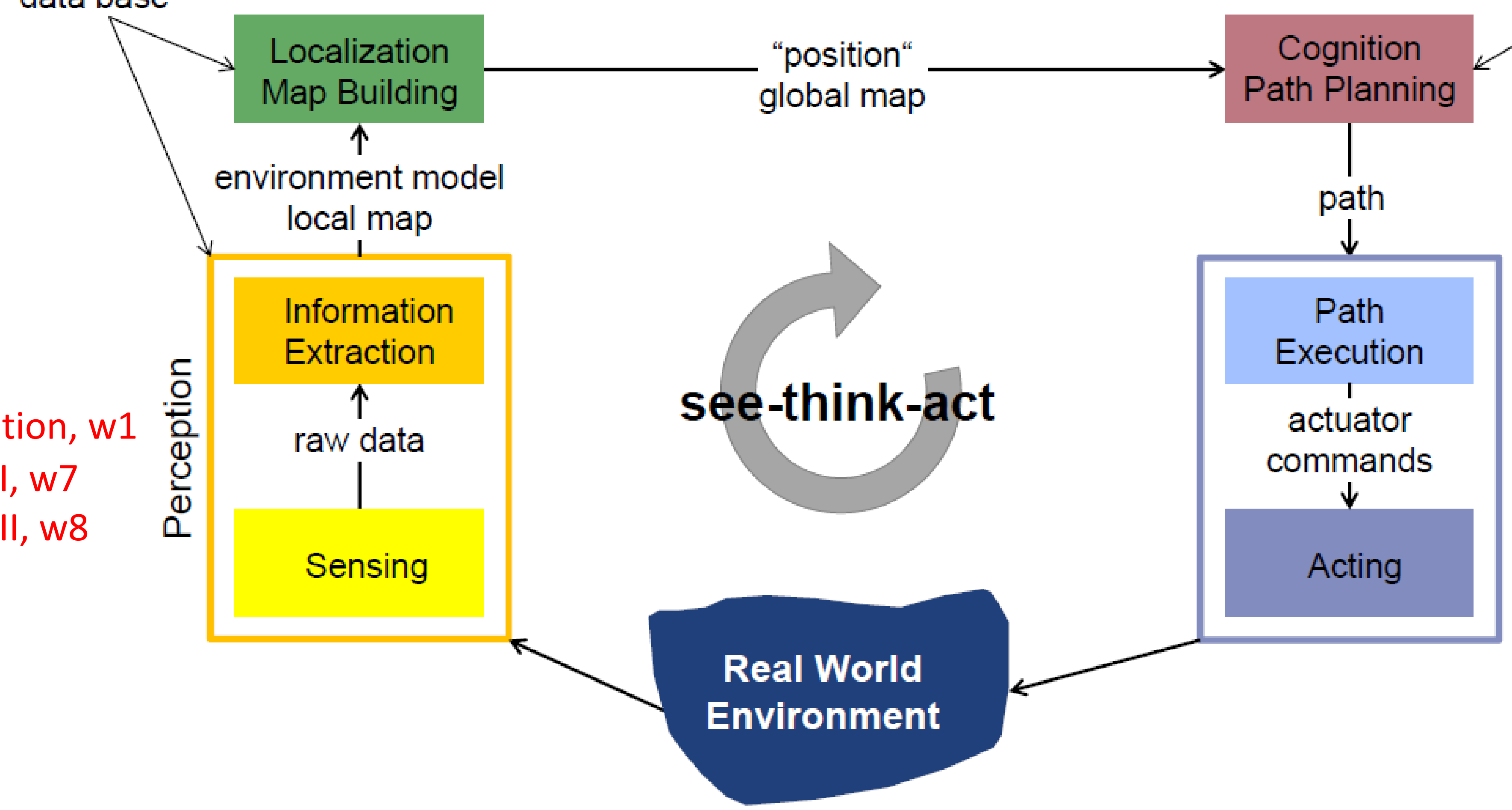
What methods/sensors can be used for localisation in the course project?

ⓘ Start presenting to display the poll results on this slide.

What we have learnt today

- Behaviour-based navigation vs. Map-based navigation
- Five different map representations
- Two different localisation methods
  - Global localisation
  - Dead reckoning/Odometry
- Error sources and calibration for dead reckoning

Next week: Kinematics



Localisation I, w2
Localisation II, w9

Planning I, w4
Planning II, w5

knowledge, data base

Localization Map Building

"position" global map

Cognition Path Planning

mission commands

environment model local map

path

Perception, w1
Vision I, w7
Vision II, w8

Perception

Information Extraction

raw data

Sensing

see-think-act

Path Execution

actuator commands

Acting

Motion Control

Locomotion, w1

Kinematics, w3

Real World Environment

R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to autonomous mobile robots. The MIT Press. Second edition. 2011.