Sistemi Intelligenti Avanzati
Corso di Laurea in Informatica, A.A. 2020-2021
Università degli Studi di Milano

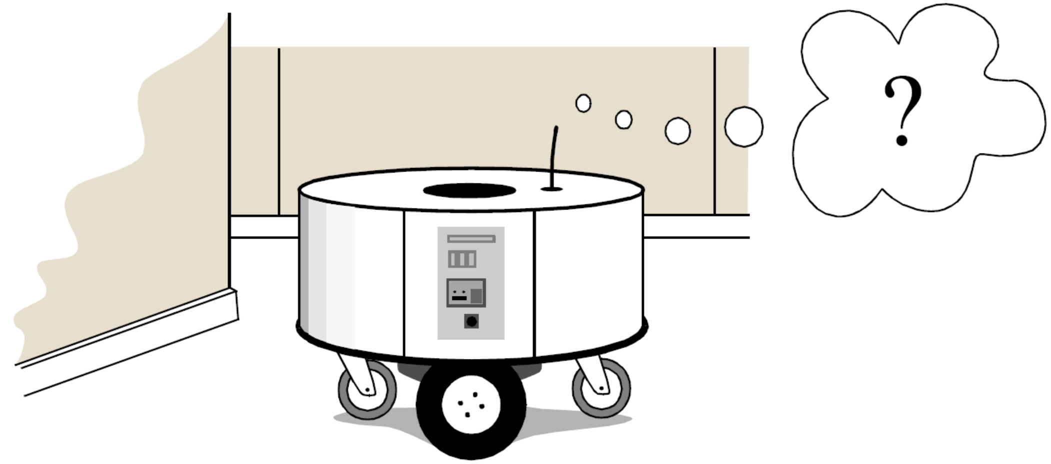
Introduction to Autonomous Mobile Robotics L2

Matteo Luperto
Dipartimento di Informatica
matteo.luperto@unimi.it

Outline

Overview of core concepts:

- Robot Motion
- Perception
- Localization and Mapping
- Navigation

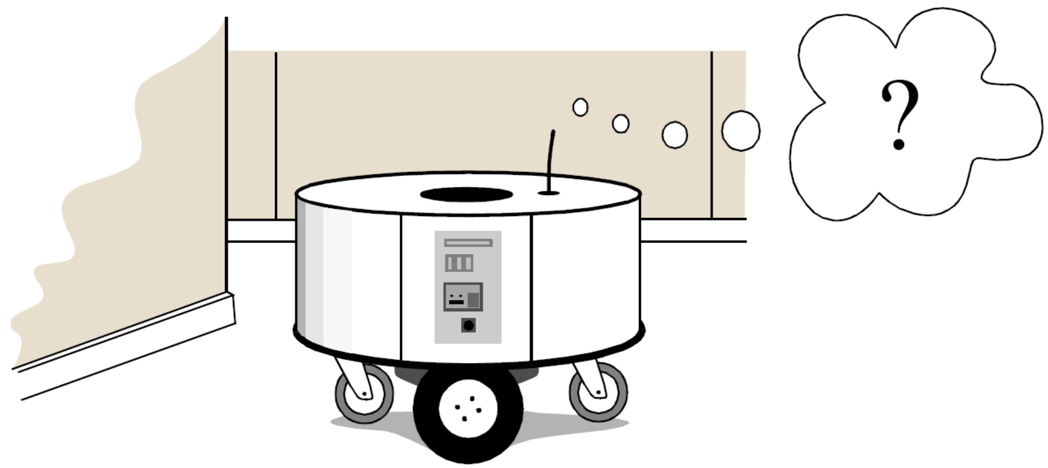


Assumption: let's talk about the simplest type of mobile robots, wheeled ground vehicles

Outline

Overview of core concepts:

- <u>Robot Motion</u>
- Perception
- Localization and Mapping
- Navigation



Assumption: let's talk about the simplest type of mobile robots, wheeled ground vehicles

Robot Motion

Locomotion

Wheels

Configuration

Kinematics

Robot Wheels

1

2

3

4

Four main types of wheels:

1. Standard wheel - 2 DOF - rotation around the wheel axle
2. Castor wheel – 2 DOF – rotation around the steering joing
3. Mecanum wheel (Swedish or Omni Wheel) – 3DOF – rotation around wheel axle, rollers, contact poing, 45° or 90°
4. Ball or Spherical Wheel

Mecanum wheel = omnidirectional

Wheel Configuration

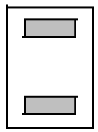
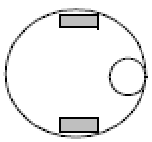
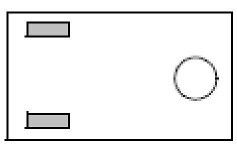
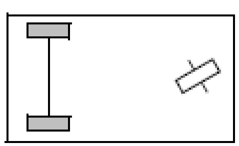
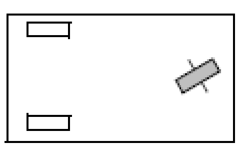
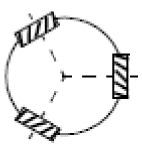
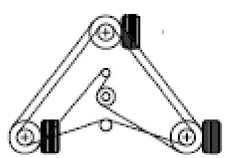
- How many wheels? 2,4,6,8?
- How many axes?
- What type of wheels?

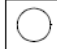
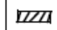
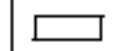
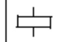
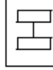
Targets:

- Stability = robot does not fall → 2 wheels minimum, 3+ for “robust” solutions
- Maneuverability = do we have motion constraints? (e.g., car in parallel parking)
- Controllability = how difficult is to control movement?

Usually, maneuverability and controllability are inversely correlated

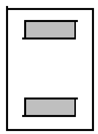
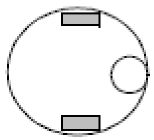
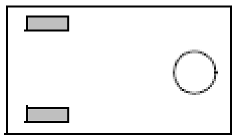
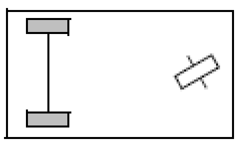
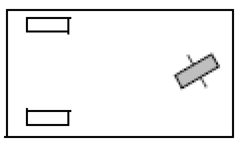
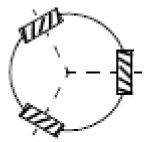
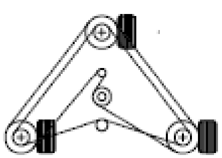
Wheel Configuration

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Three synchronously motorized and steered wheels; the orientation is not controllable	"Synchro drive" Denning MRV-2, Georgia Institute of Technology, I-Robot B24, Nomad 200

Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Wheel Configuration

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Three synchronously motorized and steered wheels; the orientation is not controllable	"Synchro drive" Denning MRV-2, Georgia Institute of Technology, I-Robot B24, Nomad 200

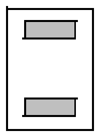
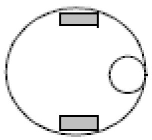
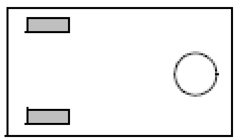
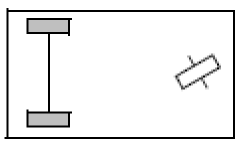
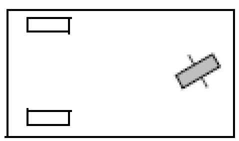
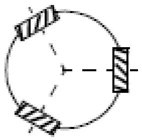
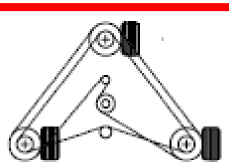
Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

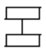
Popular configurations:

- Limited number of wheels
- Limited motors
- Simple

From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Wheel Configuration

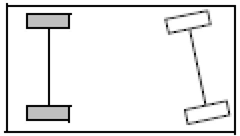
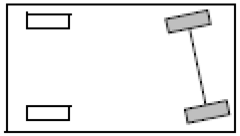
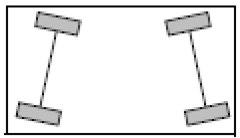
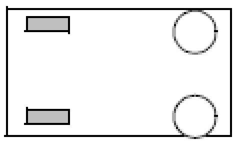
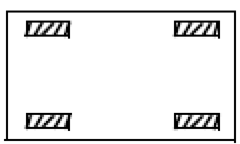
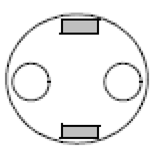
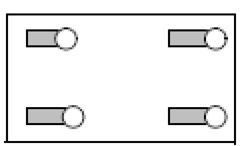
# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Three synchronously motorized and steered wheels; the orientation is not controllable	"Synchro drive" Denning MRV-2, Georgia Institute of Technology, I-Robot B24, Nomad 200

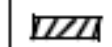
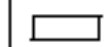
Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

Omnidirectional with 3 motors and a simple architecture

From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

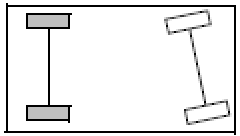
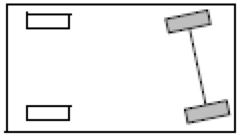
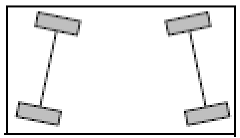
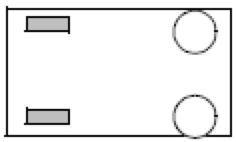
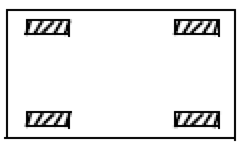
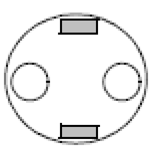
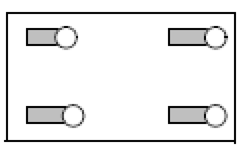
Wheel Configuration

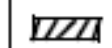
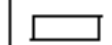
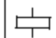
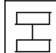
# of wheels	Arrangement	Description	Typical examples
4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

From [Siegwart, Introduction to Autonomous Mobile Robots]

Wheel Configuration

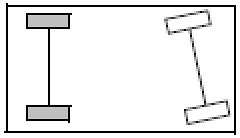
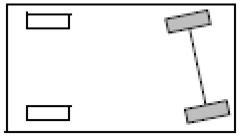
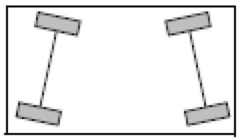
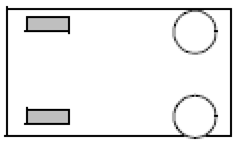
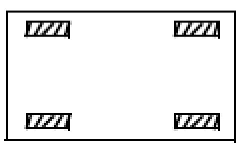
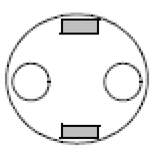
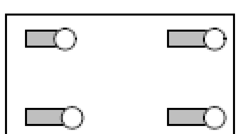
# of wheels	Arrangement	Description	Typical examples
4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

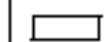
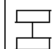
Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

Car configuration – parallel parking

From [Siegwart, Introduction to Autonomous Mobile Robots]

Wheel Configuration

# of wheels	Arrangement	Description	Typical examples
4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

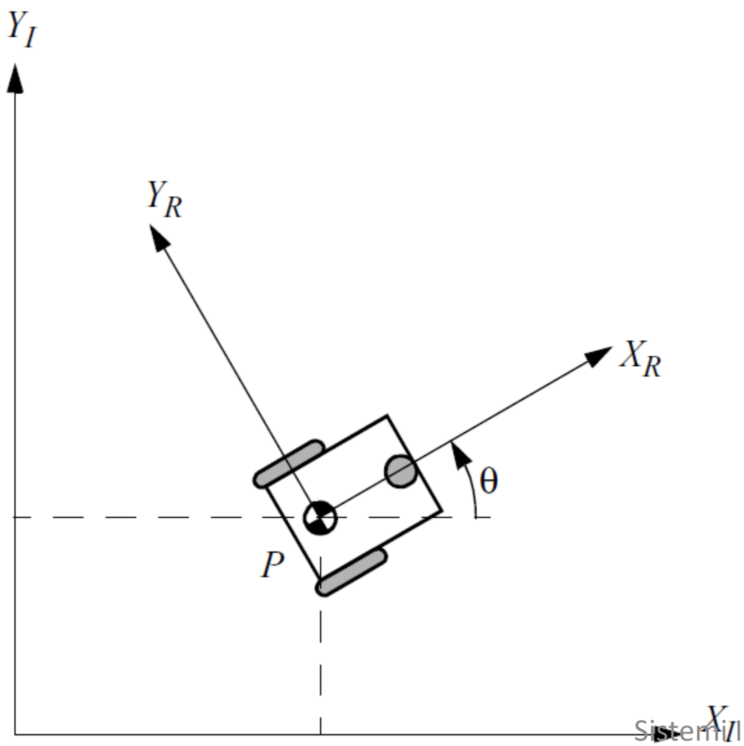
Icons for the each wheel type are as follows:	
	unpowered omnidirectional wheel (spherical, castor, Swedish);
	motorized Swedish wheel (Stanford wheel);
	unpowered standard wheel;
	motorized standard wheel;
	motorized and steered castor wheel;
	steered standard wheel;
	connected wheels.

Omnidirectional – 4 wheels

From [Siegwart, Introduction to Autonomous Mobile Robots]

Kinematics

- Describe how a mechanical system behaves, is needed to create control software for the robot
- Kinematic Model of the robot and Constraints
  - Representing the robot position and the robot movement in a global and local reference frame



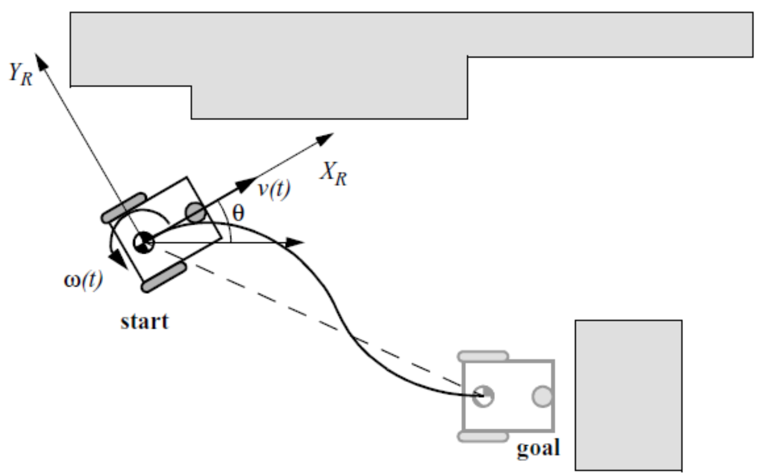
The robot pose is expressed as

$$[x, y, \theta]$$

in the global reference frame

Kinematics

- Forward Kinematics computes the robot speed in the global reference frame given the spinning speed of each wheel
- Inverse Kinematics compute the robot actuators parameters to reach a given configuration
- Each wheel configuration results into a set of constraints



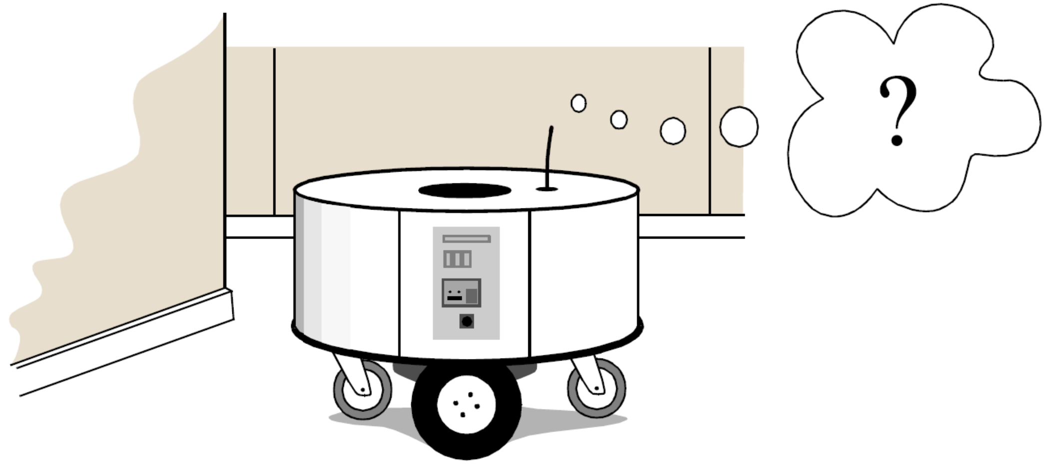
Usually, robot DDOF are considered: Differentialy Degrees of Freedom (that are equal to the degree of mobility of the robot)

$$DDOF \leq \delta_m \leq DOF$$

Outline

Overview of core concepts:

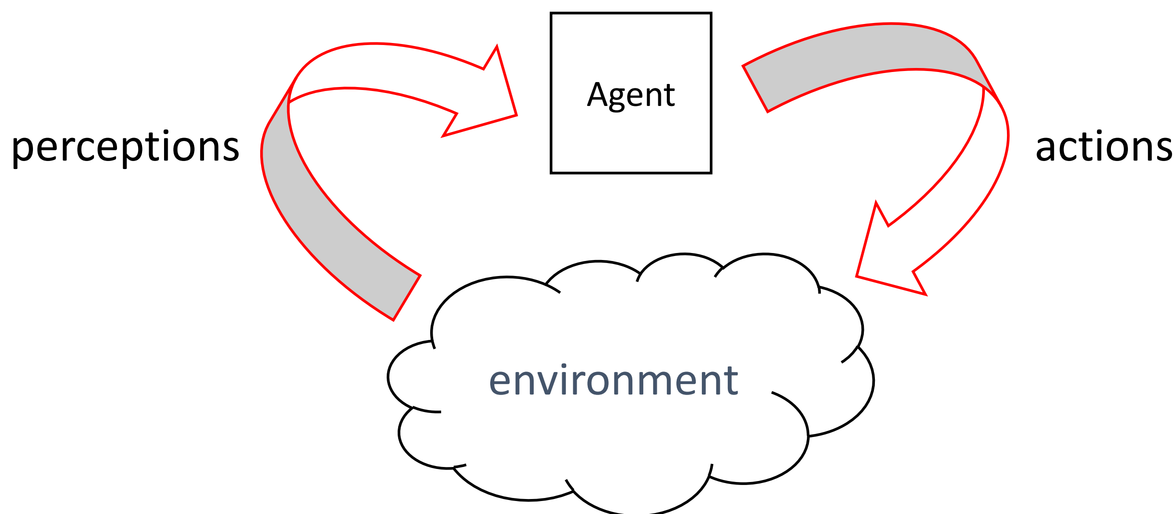
- Robot Motion
- <u>Perception</u>
- Localization and Mapping
- Navigation



Assumption: let’s talk about the simplest type of mobile robots, wheeled ground vehicles

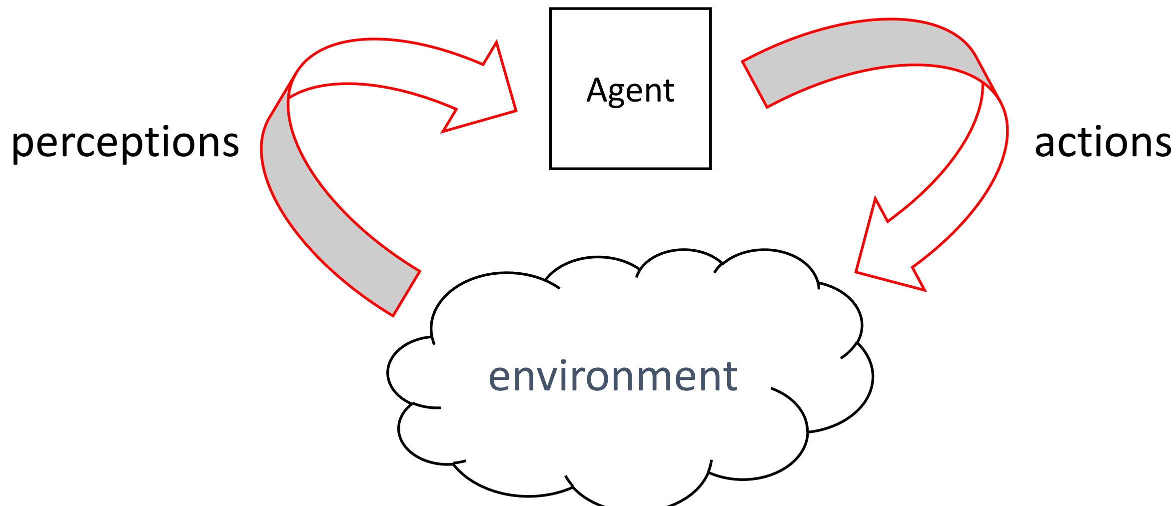
Perception: sensor types

- Proprioceptive
  - Sensor measure values internal to the system as motor speed, wheel load, robot arm joint angles, battery voltage
- Exteroceptive
  - Sensors acquire information from the robot's environment as distance measurement, light intensity, sound amplitude = meaningful environmental features



Perception: sensor types

- Passive sensors
  - Measure ambient environmental energy entering the sensors, as microphones, temperature probes, cameras
- Active sensors
  - Emit energy into the environment, then measure the environmental reaction. More control, more accuracy, but interference issues (and sometimes power)



Perception: sensor types

What to measure? What is the robot task?

- Vision
- Obstacle distance
- Position
- Environmental monitoring (ASV)
- Olfaction (e.g. inspection of chemical plants)
- Temperature (e.g. inspection of a server farm)

The most important sensors are those involved in the robots' mobility

Sensors performance characterization

- Dynamic Range
  ratio between maximum and minimum input value – usually in dB
- Resolution
  minimum difference between two values that can be detected
- Linearity
  how the sensor respond to changing inputs
- Bandwith or Frequency
  speed with which a sensor can provide a stream of readings
  number of measurements per seconds (in hz)

These specs of the sensors are usually measured in labs – controlled enviromnets;
however, often we need to identify how the sensor performs in its real-world deployment

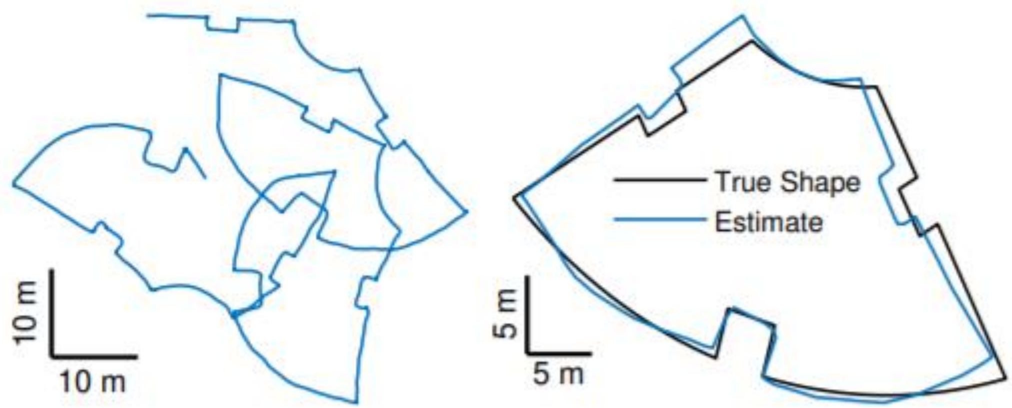
In Situ Sensors performance characterization

- Sensitivity:
  measures how incremental change in the target input changes the output signal
- Cross Sensitivity
  sensitivity to environmental external parameters that are orthogonal to the target parameter; high cross-sensitivity is task-related and unwanted
- Error
  difference between output and true value
- Accuracy
  degree of conformity between sensor’s measurement and true value (usually %)
- Systematic Error
  errors caused by factors that, theoretically, can be modeled; deterministic; example: calibration errors, slopes, …
- Random Error
  errors that cannot be predicted using a model nor can be mitigated; modeled as probabilistic process (stochastically)
- Precision
  not to be confused with accuracy; reproducibility of the results: if the phenomena is the same, the measured value should be the same (this holds if I use several different sensors of the same type: I expect the same results from all of them)

Challenges in Sensors modeling

- Blurring of systematic and random errors
  active ranging sensors tend to have failures that are triggered by specific relative position of the sensor and of the environment (e.g. glass surfaces, mirrors, …)
  During motion this happens at stochastic intervals
  Moreover, robot usually have different and concurrent sensors
  This, combined, is used to model error and to smooth their impact wrt the robot activity

(a) The courtyard of our Institute. We used the inner lawn area for testing the proposed mapping method.



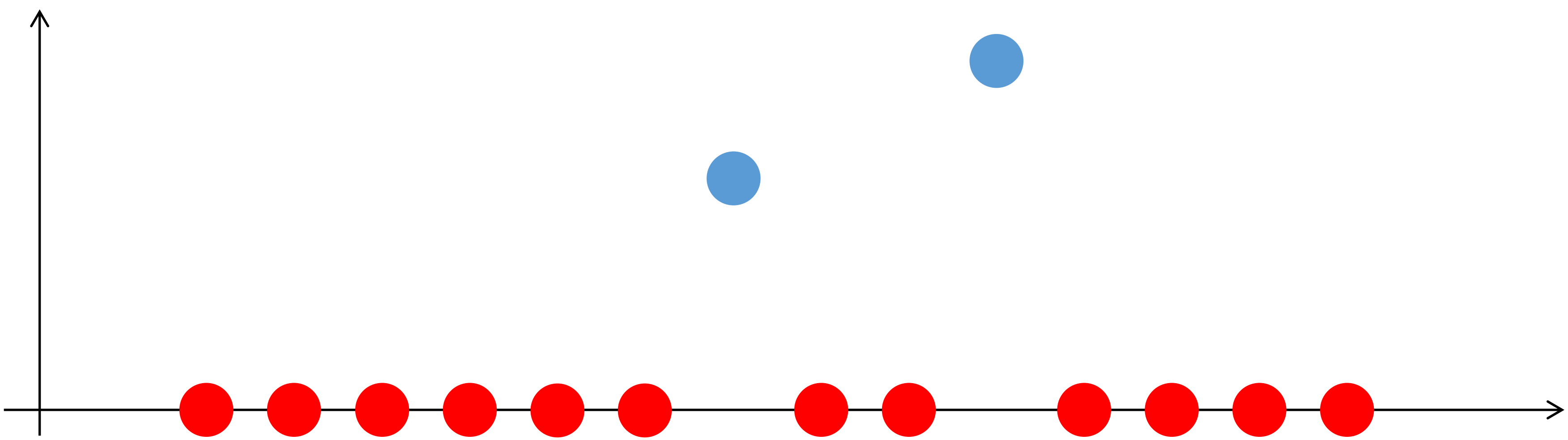
(b) The left panel shows the estimated path of the robot generated from its wheel odometry and the right panel the estimated map and the true shape of the test environment.

Fig. 9. The real courtyard depicted in (a) and the collected odometry data together with the map estimate shown in (b).

Example: [Rottman et al, ECMR 2019]

Challenges in Sensors modeling

- Multimodal error distribution
  a common choice is to characterize the behavior of a sensor's random error in terms of a probability distribution over various output values; diverging from the model can help to detect errors (measuring the correct value is most probable)

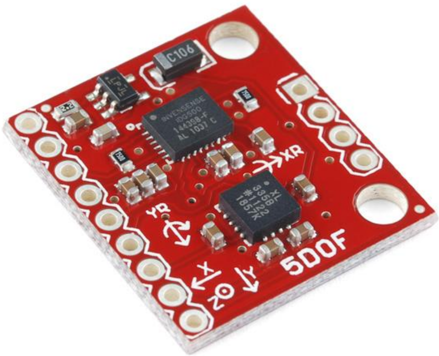


Wheel/Motor sensors

- Proprioceptive sensors used to measure the internal state and dynamics of the robot
- Optical encoders: measure the angular speed and position within a motor drive, or shaft of a wheel or steering mechanism
- Used for localization and to estimate the robot movements
- While the sensor itself could be accurate, the measure is inherently inaccurate (odometry) and needs integration (it measures the motor itself, what if a wheel slips? or if there is a slope?)

Heading Sensors

- Compasses
  - outdoor
- Ground-based-beacons
  - **GPS** outdoor
  - We can have similar solutions indoor that usually requires other complex sensing capabilities (vision) or detection of NFC or RFID tags installed in the environment
- Gyroscope
  - Usually combined with accelerometers in an IMU
    Inertial Measurements Unit



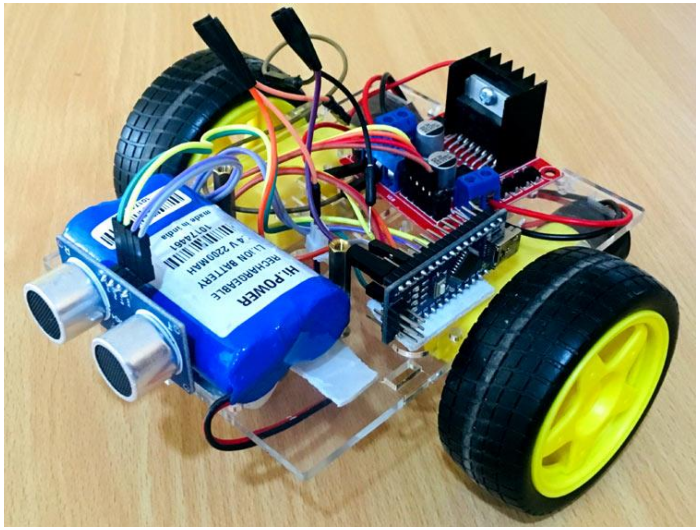
Active ranging

- Most popular sensors in mobile robotics
- Usually have a low price point and easily interpreted outputs
- Among them, *time-of-flight* sensors are those commonly used
  - $d = c \cdot t$
  - d distance travelled
  - c speed of wave propagation
  - t time of flight

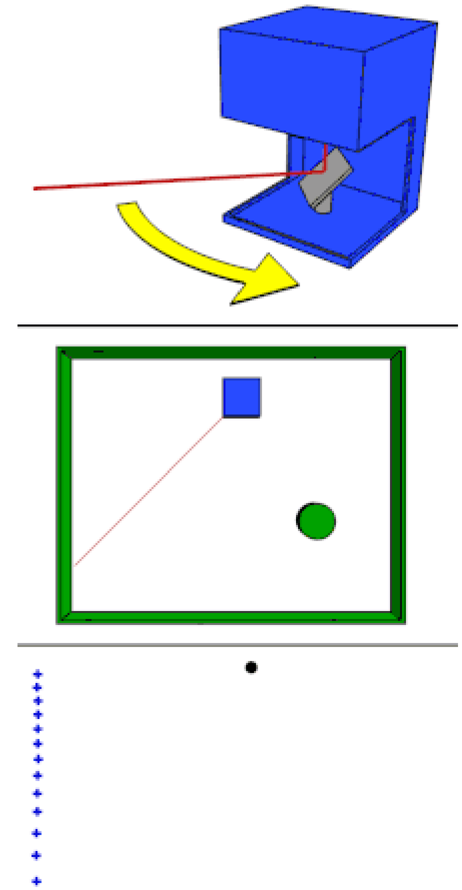
1. Sonars
2. Laser Range Finder

Ultrasonic Sensors - Sonars

- Cheap
- Not particularly accurate
- Simple and interpretable measurements
- Good for proximity – obstacle avoidance
- Low range



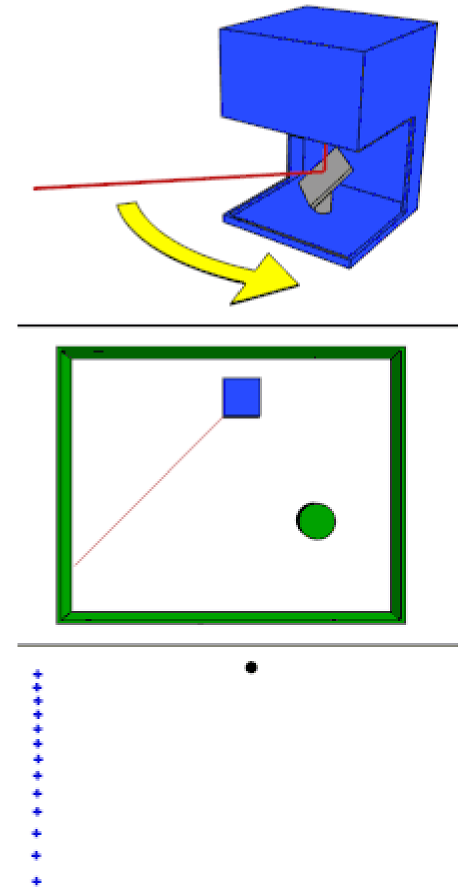
Laser range finders - Lidars



Time of Flight (ToF) sensor which is used to *scan* the surrounding of the robot:

- Range = max perceivable distance
- Field of View (FOV) = degrees of a scan, from 180° to 270°, 360°
- Angular resolution = how many points for each degree in a scan
- Frequency = how many scan per second (1hz-50hz)
- 2D or 3D

Laser range finders - Lidars



Widely used in most indoor and outdoor robot applications as it is:

- Relatively cheap
- Easy and interpretable measures
- Robust wrt environmental changes (e.g. day, night, different seasons)

Laser range scanner are the most important sensor for most autonomous mobile robots

Laser range finders - Lidars

Different tasks – different environment – different lidar types:

Indoor lidars have a range from 3-5m to 10-20m, with a FOV of 180-270°.

They are relatively cheap (250€ for unreliable entry level lidars, 1000-5000€ for reliable models).

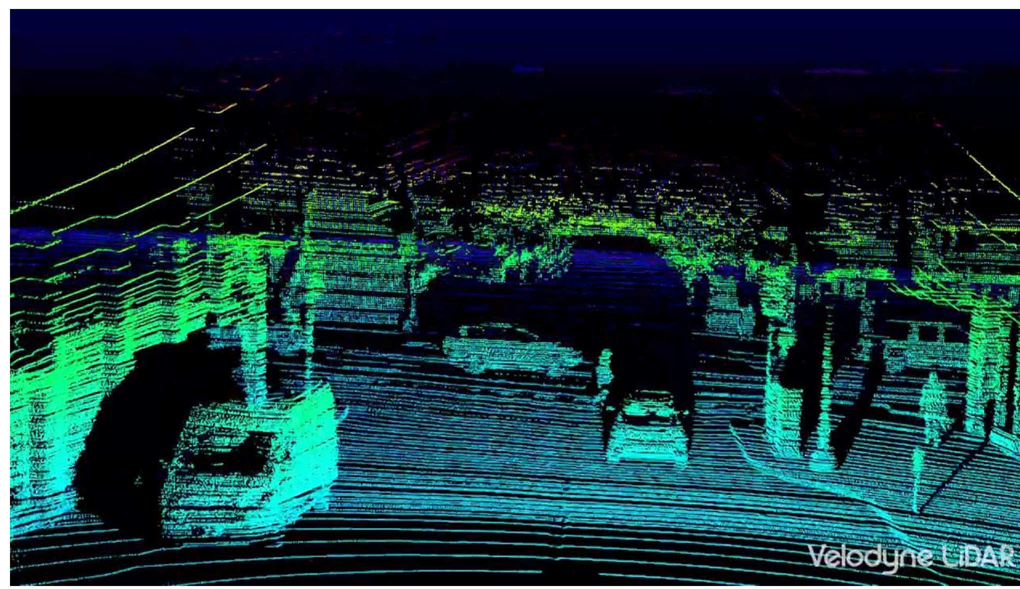
Outdoor lidars have a range from 10 to 30-50m, with a FOV of 180-360°.

Price is higher (5-15k €) but still reasonable, performance are good.

Laser range finders - Lidars

In outdoor applications (autonomous vehicles), 3D lidars are a popular choice:

- Multi-layered lidar, not really 3D (from a single source)
- Usually 360° FOV
- Usually longer ranges (up to 200m)
- Expensive (10k-100k€)
- More data but also more complex to interpret



Vision

With the rising of Deep Learning, vision has becoming more and more important in robotics

Cameras provide a lot of data, are *relatively cheap*, but their output is also much more complex to interpret than the one of LIDARS.

- Limited range
- Distortion
- *Reliability* (day-night or light changes)
- Calibration

Lidar VS Camera

Lidar

- Cheap
- Long range
- Up to 360° FOV
- Usually 2D
- Simple output
- Subject to reflections
- Measure the spatial surrounding of the robot
- Good to infer spatial occupancy, difficult to infer semantics

Camera

- Cheap
- Close range
- Limited FOV
- 3D
- Complex output
- Subject to distortions, changing light conditions, …
- Measure the appearance of the surrounding of the robot
- Could be used to infer semantic knowledge

RGBD Cameras

- camera + depth information using an active sensor
- easy to reconstruct 3D image of the environment
- good for a lot of sensing tasks (e.g. human detection)
- widely used and useful, especially indoor
- limited range - depth (useless at 3/5m, sometimes even before)
- distortion
- cheap (100€→1000€)
- usually camera is not particularly good

(b) Asus Xtion Sensor

Other sensors types

As robot can perform several different tasks, robots could be equipped with different type of sensors:

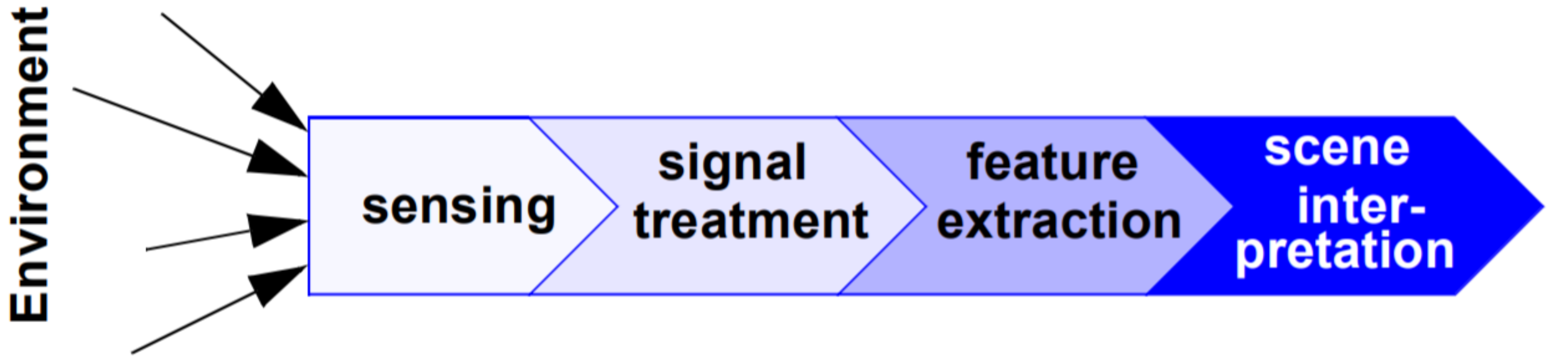
- Bumpers
- Olfactometry
- Chemicals
- Temperature sensors
- NFC readers
- RFID readers
- Radio – or other communication mechanisms…
- …

While you can expect to find one or more lidar/cameras on robot, other type of sensors are relative to the robot type/task.

Sensors wrap-up

- Robot usually have *several* sensors that are used sometimes for acquiring data related to the same subproblem, sometimes for different subproblems
- Laser range finders and cameras are usually combined
- More sensors = more data = more computational capacity required and more complexity (especially for vision)
- RGBD data are often a good compromise between data quality and complexity, but are rarely used as primary source of sensors
- There is a shift towards pure vision-based systems due also to the popularity of computer vision and deep learning, (however, this might be a trend)
- All robot data are defined by <u>errors</u> and <u>uncertainty</u> that have to be modeled (this is “easy” for lidars, but what about vision?)

Perception and Feature Extraction



To reduce the impact of inaccurate sensor readings, an idea is to extract features from one (or more) sensor data:

- Low level features: geometric primitives (lines)
- High level features: semantic labeling (object detection, people detection, …)

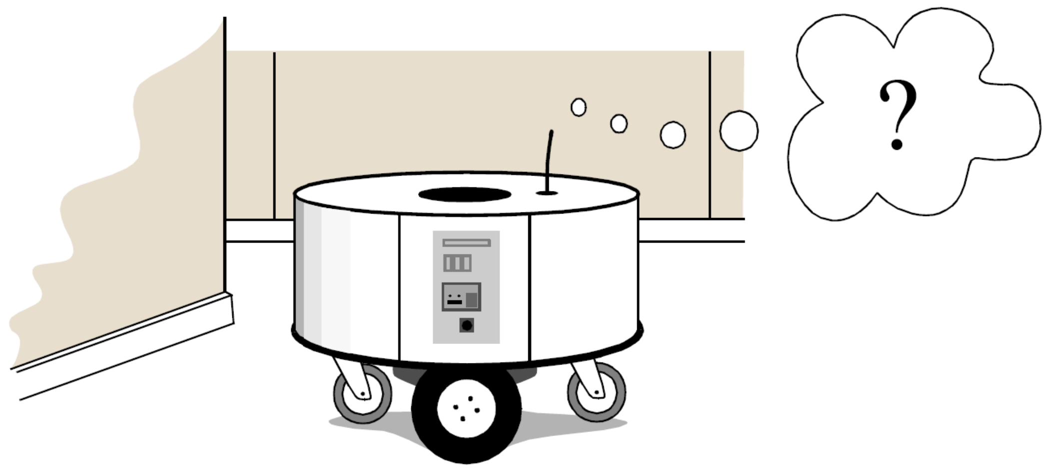
This depends on the sensors type / data quality, the environment, and the computational power required to process data

From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Outline

Overview of core concepts:

- Robot Motion
- Perception
- <u>Localization and Mapping</u>
- Navigation



From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Assumption: let’s talk about the simplest type of mobile robots, wheeled ground vehicles

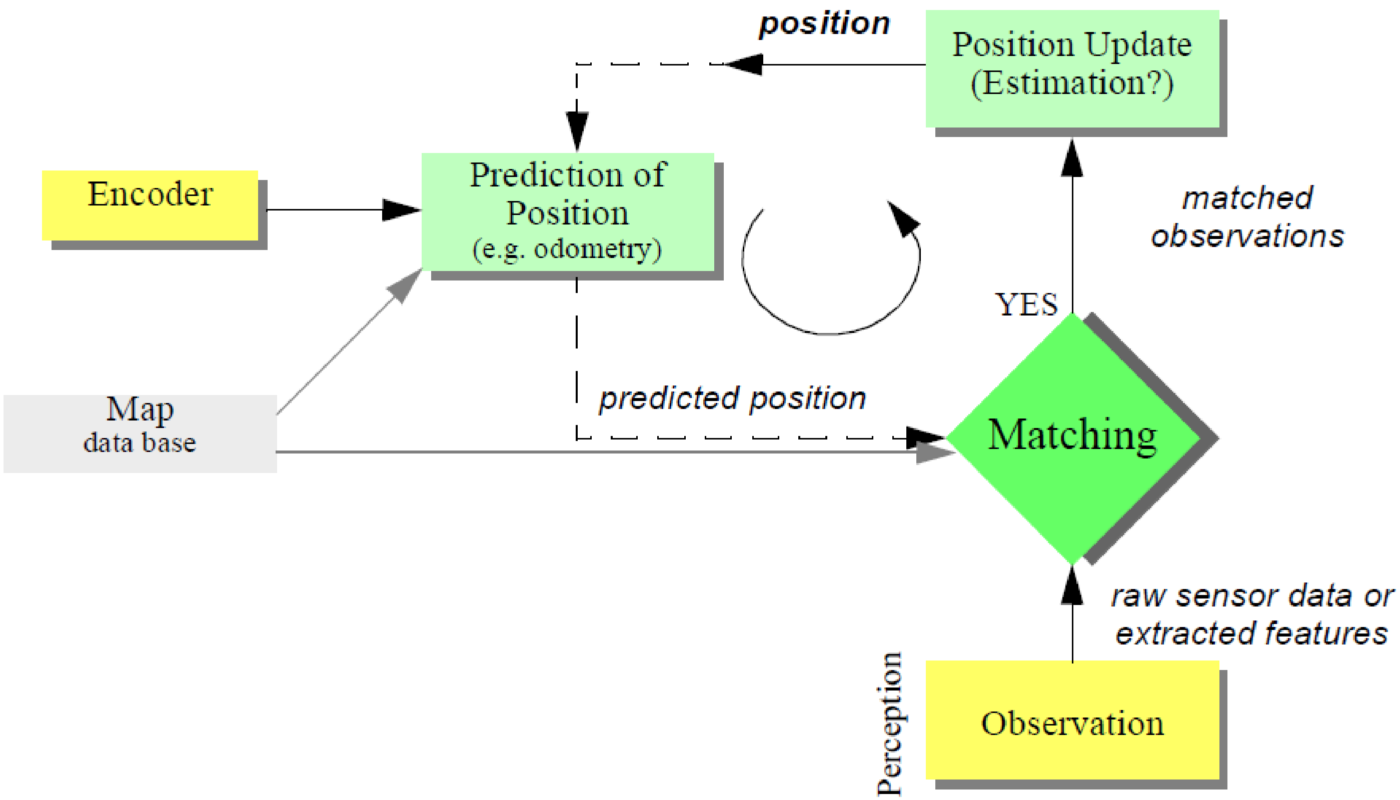
Localization and Mapping

Robot mobility requires addressing a key property: ***uncertainty***

- Environment: the world is unpredictable
- Sensors: sensors are limited, are subject to physical laws, and are subjects to noise and errors
- Robots: actuation is unpredictable, an action can not have the desired effect
- Models: we can model certain components of the robot, and also uncertainty; but models are inherenetly inaccurate, models are an abstraction
- Computation: robot are real time system, but real-time computation is often approximate

Localization

Identifying the position of the robot in a **known** environment



Localization is usually seen as an estimation problem, where we infer the robot position from available data.

Localization as estimation

Estimate the robot position from Data:

- Motion information:
  - Proprioceptive sensors, odometry
- Environmental Measurements :
  - Exteroceptive sensors as lidars, sonar, …

$$x_r = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Robot pose

$$x_{r,0:t} = \{x_{r,0}, x_{r,1}, \ldots, x_{r,t}\}$$

Robot poses from time 0 to time t

$$z_{1:t} = \{z_1, z_2, \ldots, z_t\}$$

Robot exteroceptive measurements from time 1 to time t

$$u_{0:t} = \{u_0, u_1, \ldots, u_t\}$$

Motion commands (or proprioceptive measurements) from time 0 to time t

Usually solved as using probabilistic filtering

Motion model

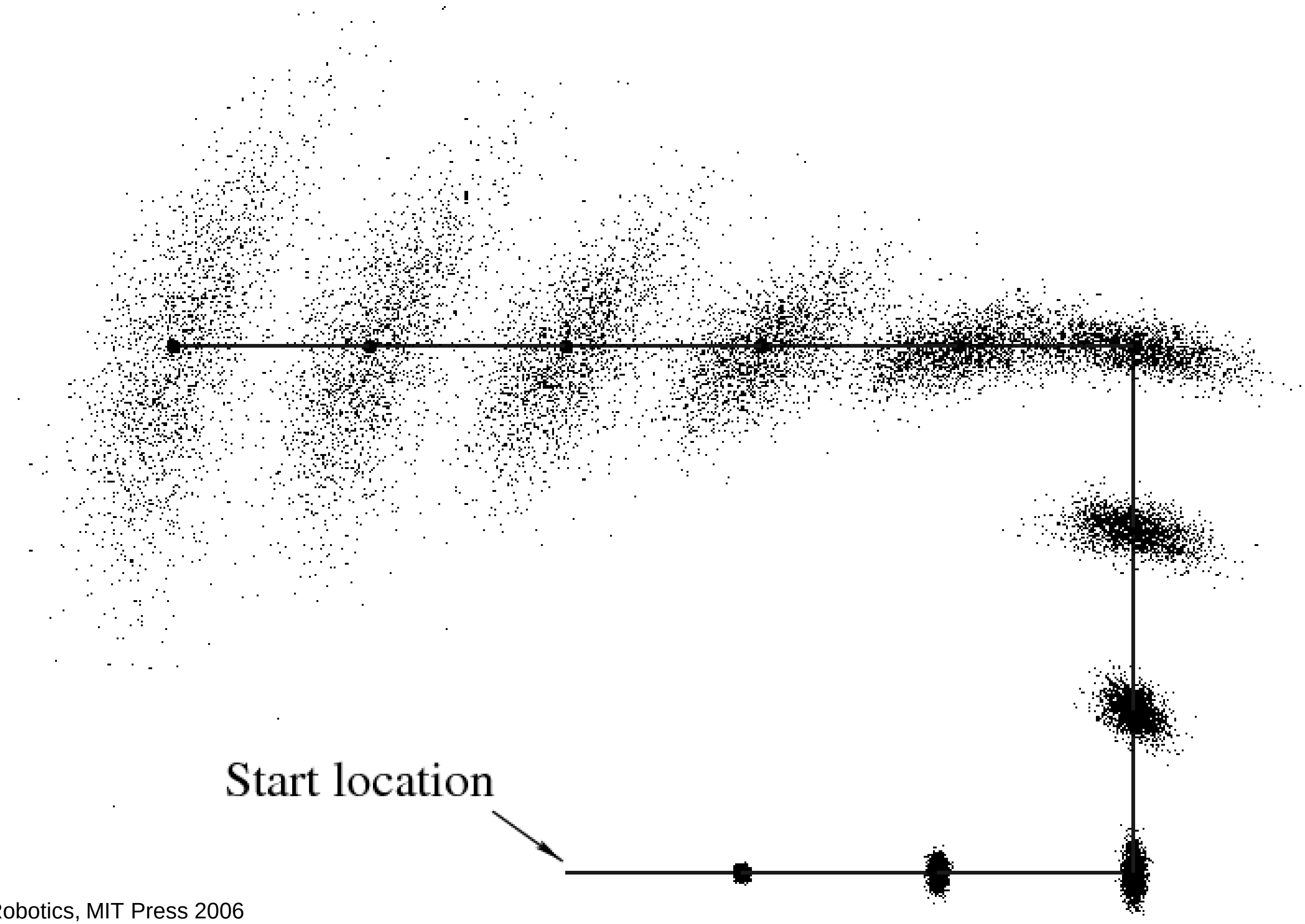
The robot motion model is the probability distribution of the robot pose at time t+1 given the robot pose and the expected robot movement, measured using motion or proprioceptive sensors:

$$p\left(x_{r,t+1} \middle| x_{r,t}, u_t\right)$$

Assuming that the robot is at $x_{r,t}$ and the control u_t is applied, we estimate the expected robot position

Motion model

Using only proprioceptive measurements, pose estimation error increases



From Thrun Burgard Fox, Probabilistic Robotics, MIT Press 2006

Measurement model

It describes the probability of a robot measurement z_t

$$p\left(z_t \middle| x_{r,t}\right)$$

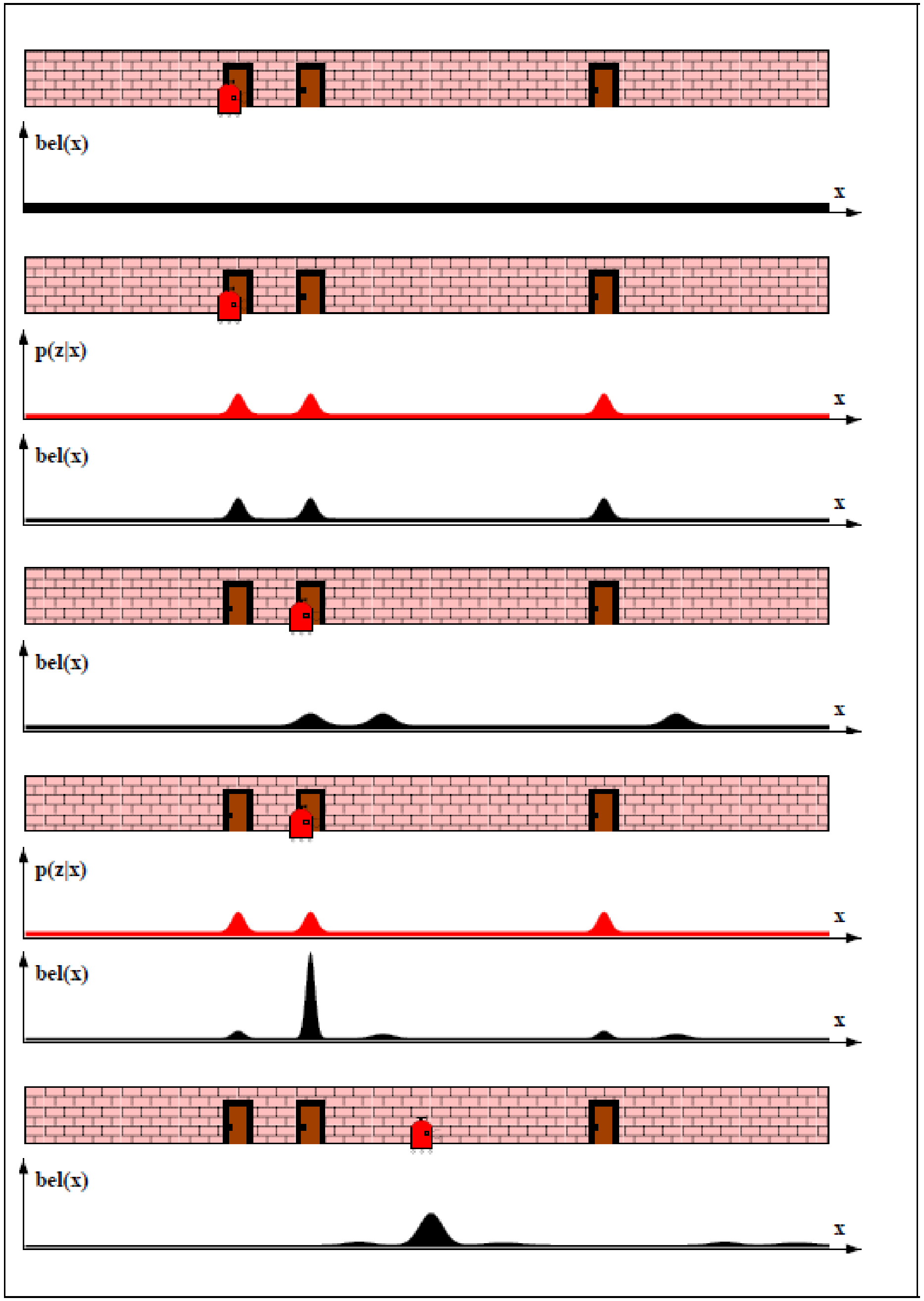
given the robot pose $x_{r,t}$ considering possible noise regarding sensors.

This is used to update the robot belief at time t

$$bel_t(x_r) = p\left(x_{r,t} = x_r \middle| z_{1:t}, u_{0:t-1}\right)$$

The robot belief is a probability distribution over the space of all possible locations of the current robot pose

Localization example



From Thrun Burgard Fox, Probabilistic Robotics, MIT Press 2006

Robot localization algorithms

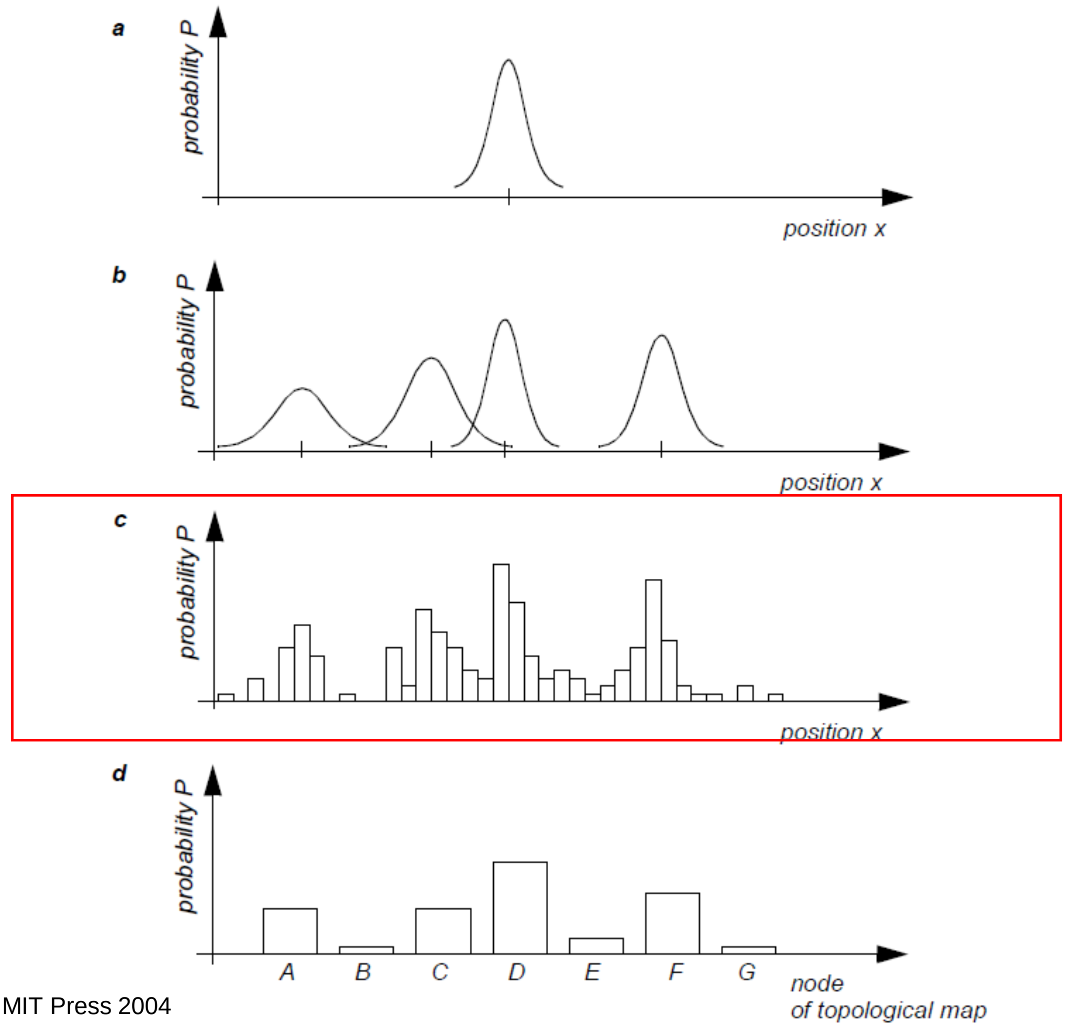
The localization model assumes so to predict the current position from movement, to observe if the measurements are coherent with the estimated position, and to close the loop by updating the robot belief.

This is done usually by exploiting probabilistic filters:

<u>Gaussian Filters</u>:	**Extended Kalman Filter (EKF),**
	Unscented Kalman Filter (UKF),
	Extended Information Filter (EIF)
<u>Non Parametric Filters</u>:	**Histogram Filter (HF),**
	Particle Filter (PF)

Belief representation

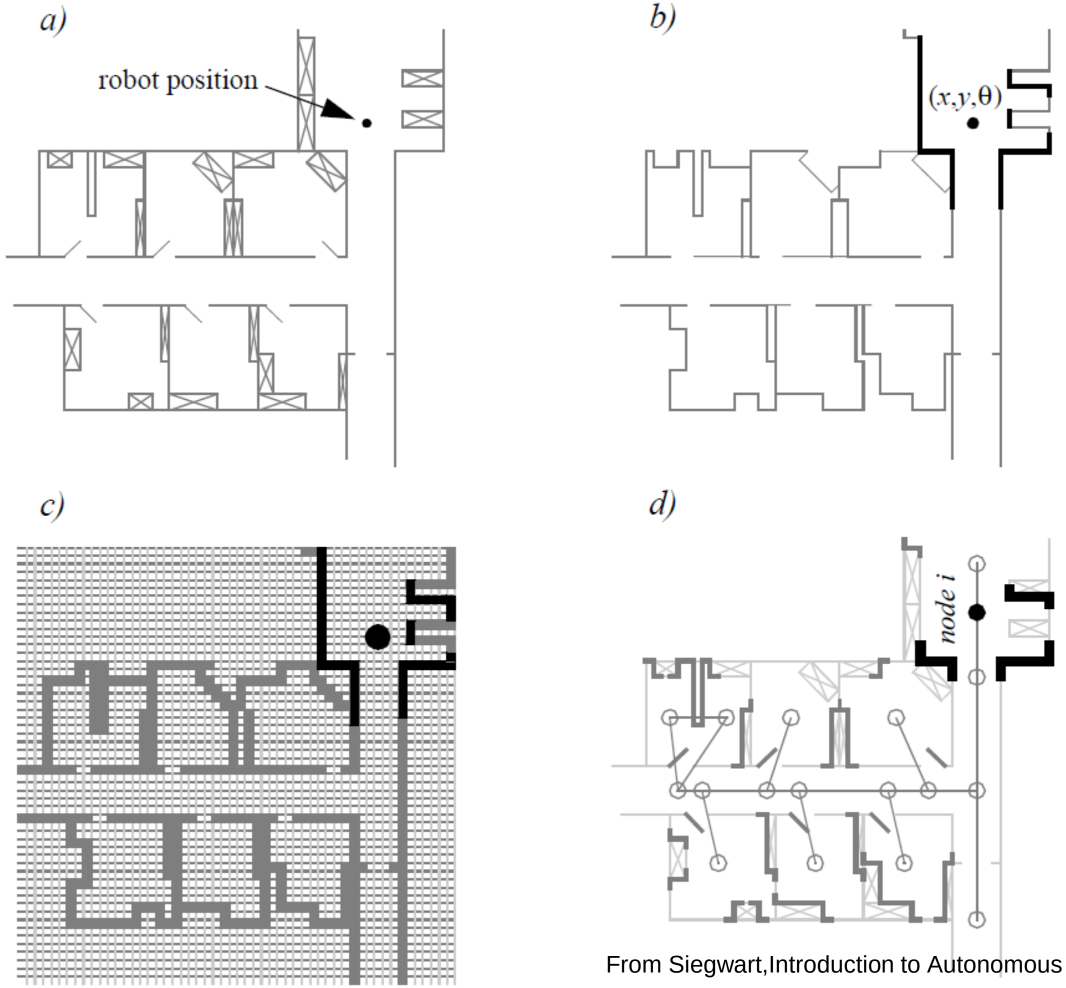
We can represent the belief as as a single hypothesis of by using <u>multiple ones</u>



From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Map representation

According to the algorithm used for localization, the type of belief distribution, we can have multiple type of map representations



From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Map representation

Usually a robot has different maps, at different level of abstraction; one of them is the one used for localization.

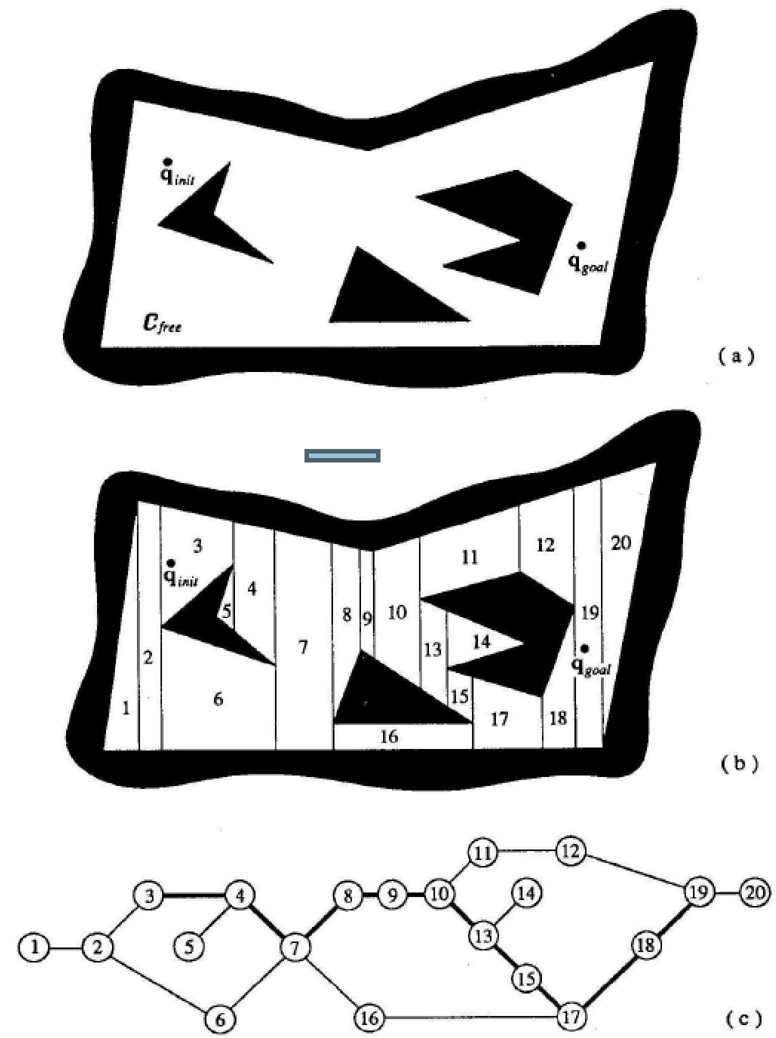
- Continuous vs discrete representation
- Occupancy vs topological maps
- Closed world assumption: only what there is in the map exists
- <u>Static</u> vs dynamic
- 2D or 3D

A map, overall, is an approximation of the environment.

Exact Cell Decomposition

This method use critical points to tesselate environment, obtaining a discrete topological map from a continuous one.

Assumption: the particular position of the robot in one of the area belonging to one node of the map does not matter, that matter is the ability of the robot to move from area to area.



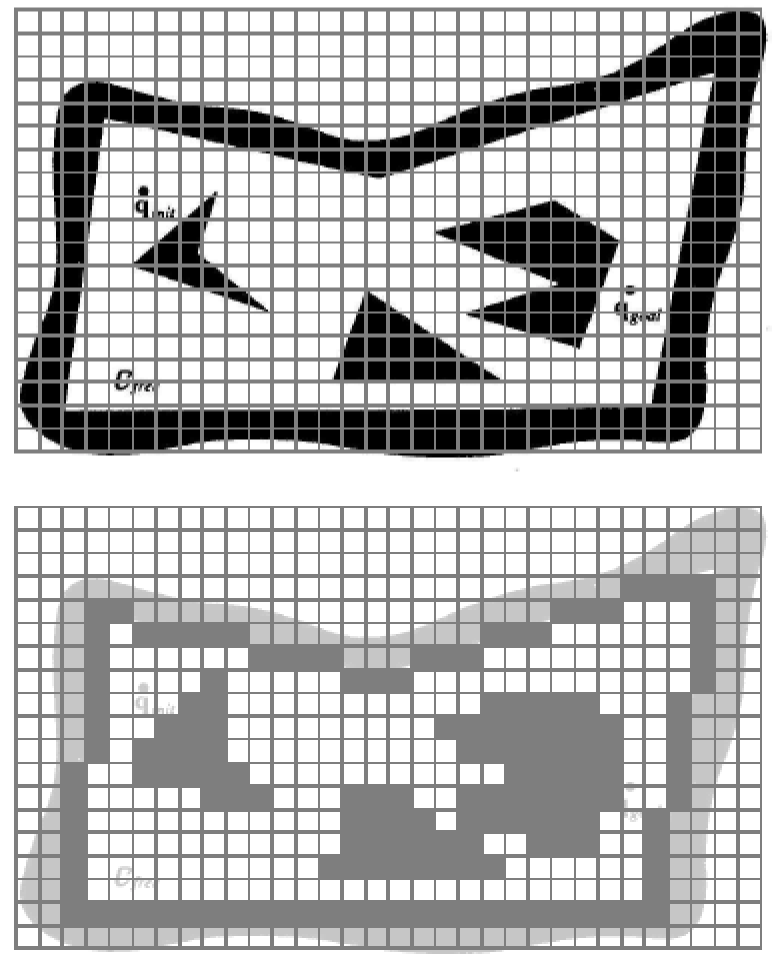
From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Fixed decomposition maps

Discretization of the map into cells of the same size, representing occupancy.

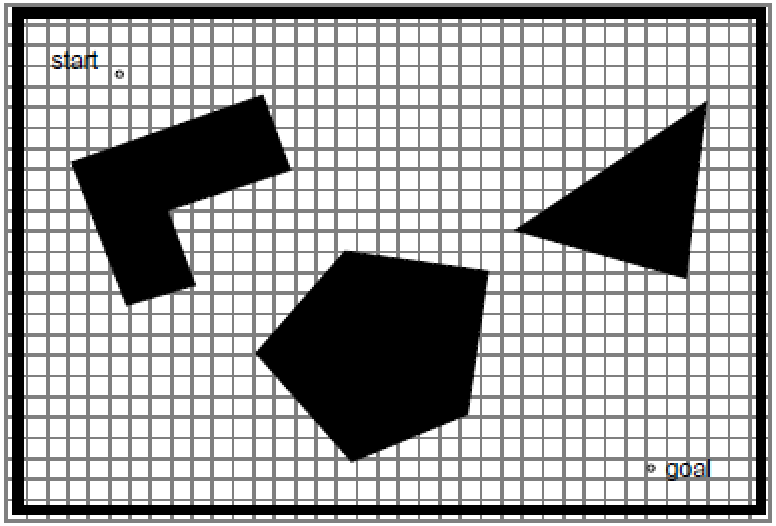
Narrow passages disappear, but each cell has the same representation.

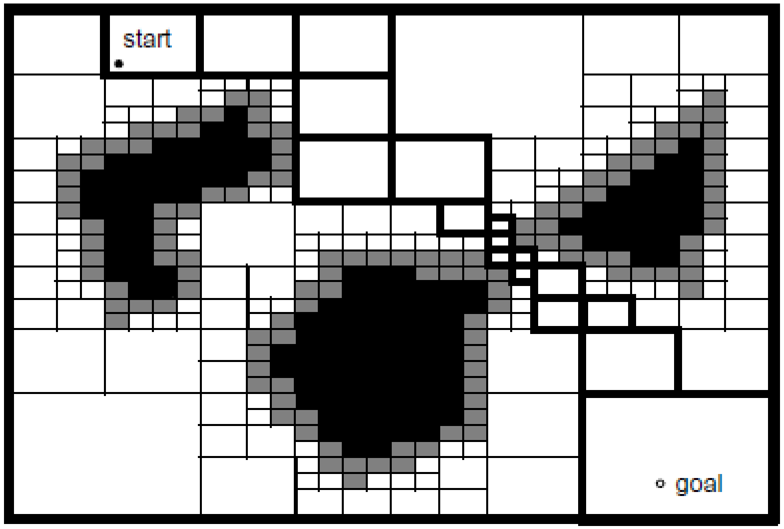
We obtain a grid map.
We can also assign different type of values to each cell (instead of 1-0, e.g. occupancy probability)

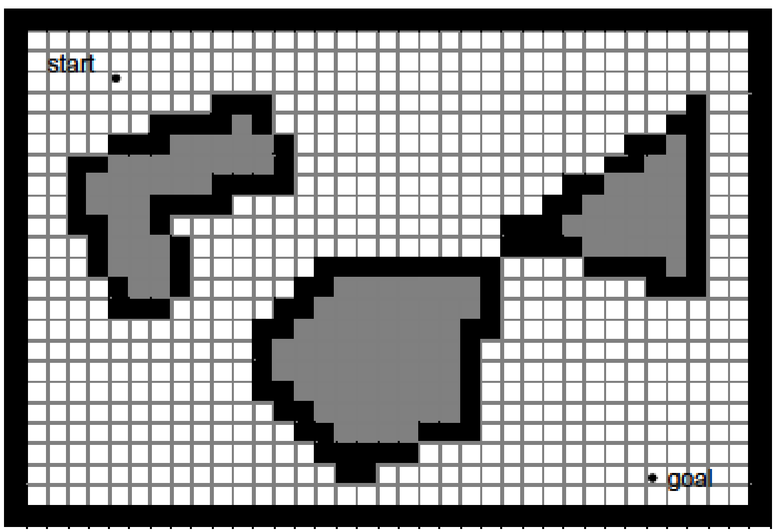


From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Grid maps





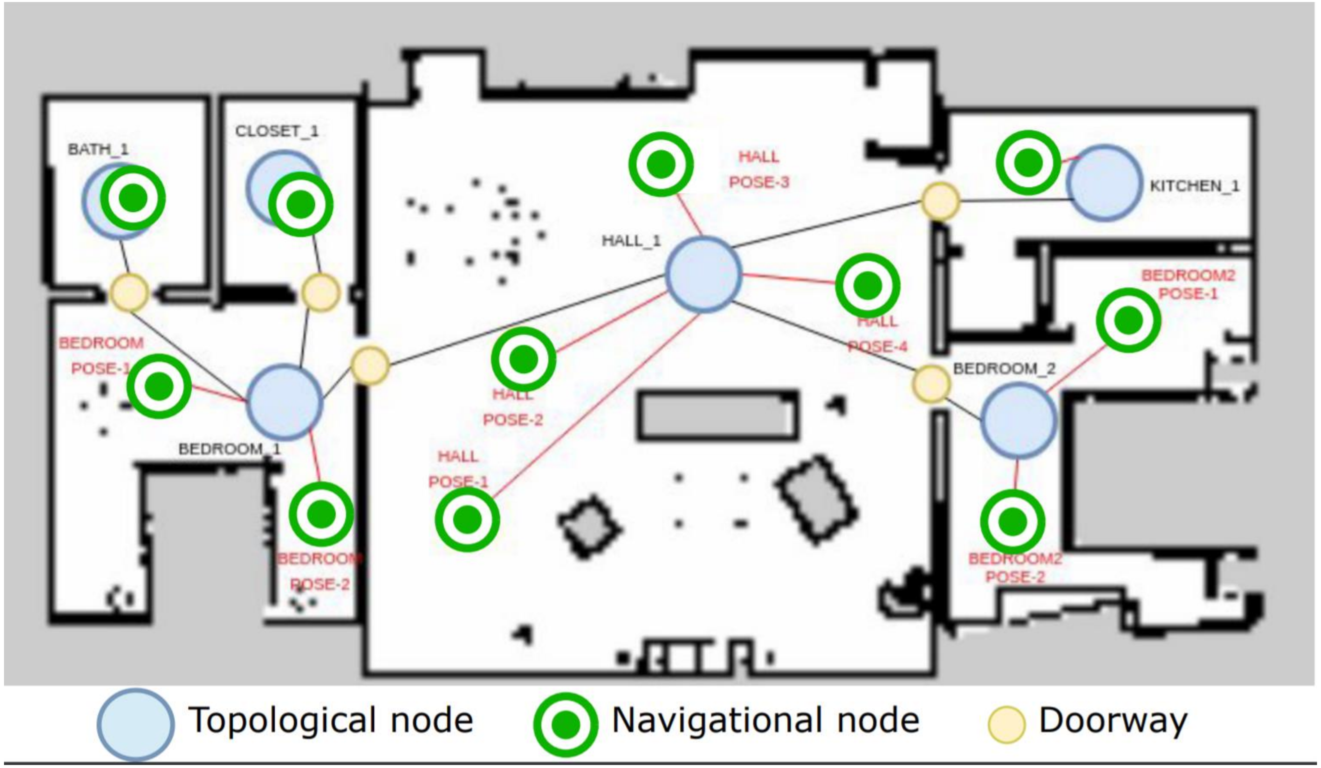


Grid map are a popular approach widely adopted.
For saving space (without losing information) we can

From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Hybrid maps

We can have different layered maps, as topological and grid maps, combined, to allow the robot to do different tasks.



SLAM

In all the previous examples, we have considered the map as known. However, what if the robot is placed in an unknown environment?

How the map is done in the first place?

The robot needs at the same time to:

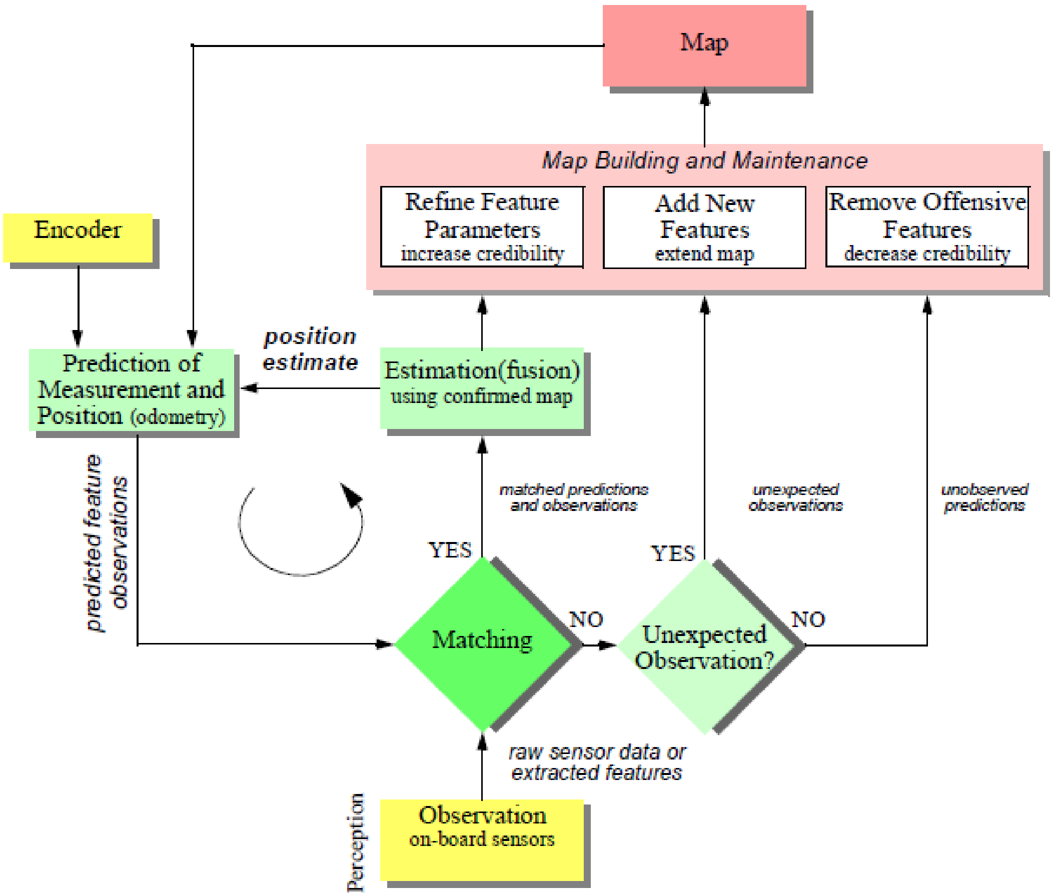
1. Map incrementally the environment integrating new observations
2. Localize itself its in the map

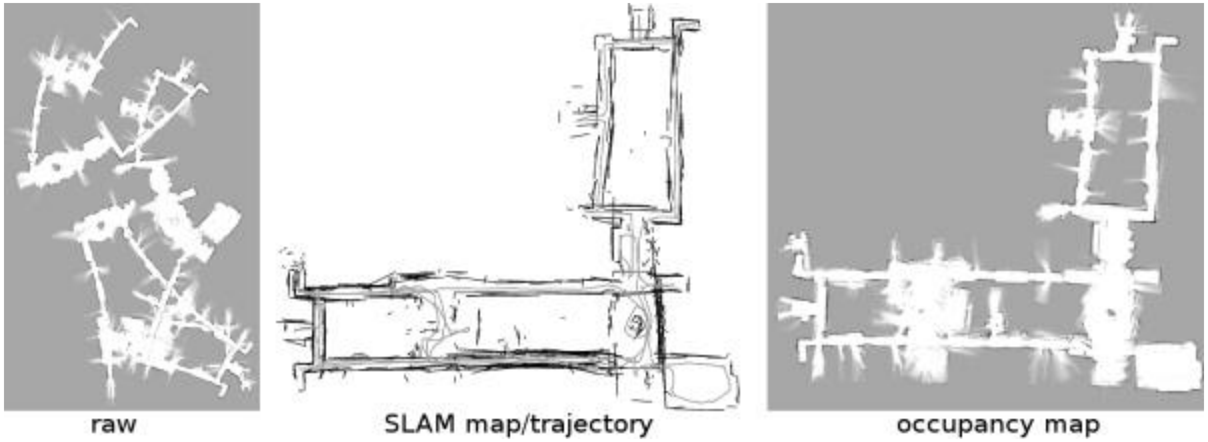
This is called Simultaneous Localization and Mapping (SLAM), a joint estimate of both the environment map and the robot pose.

SLAM 101

During SLAM the robot integrates sensorial input by correcting odometry and sensing error to provide an estimate of the environment.

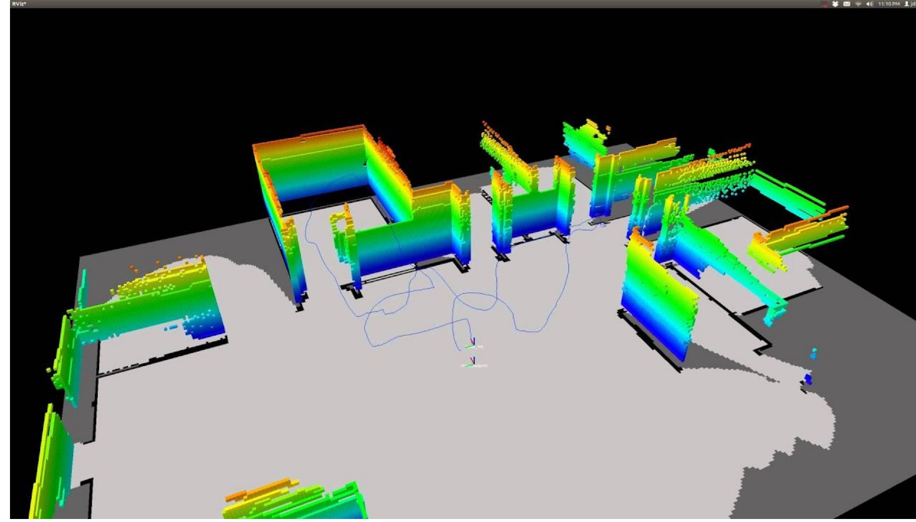
At the same time it estimates its pose in it.

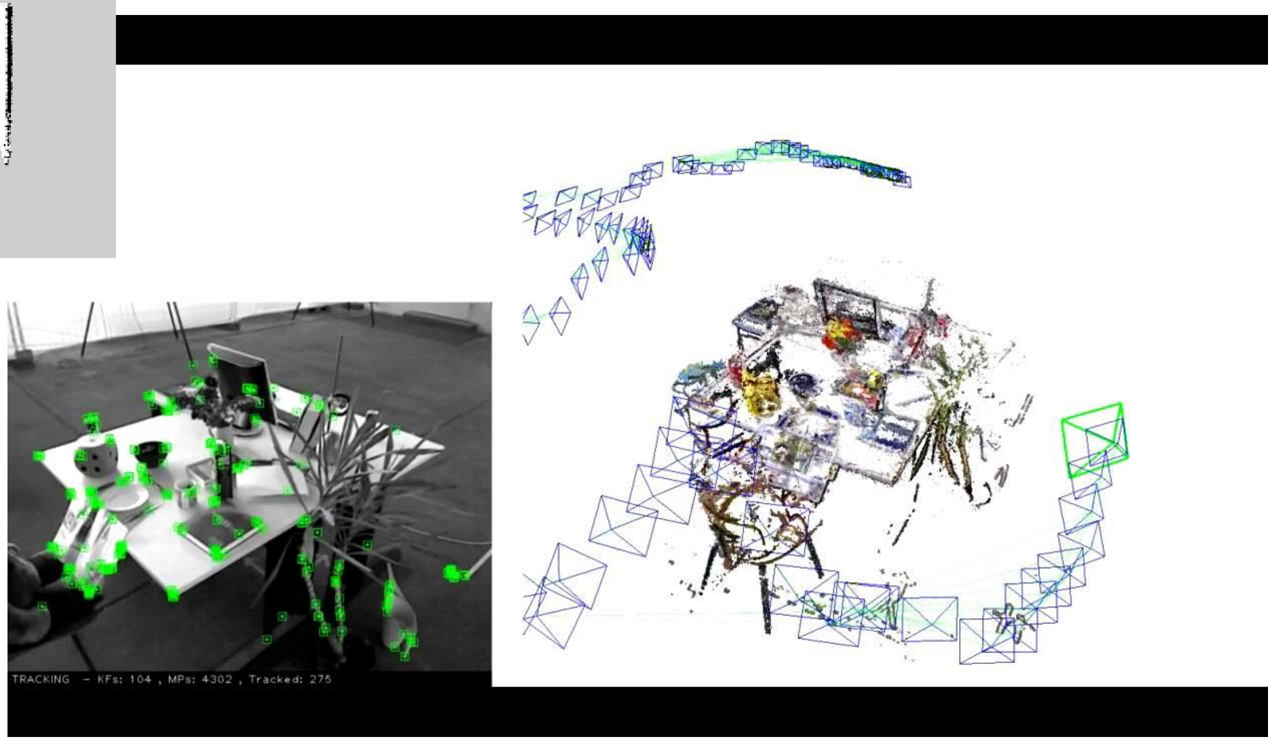




From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

SLAM

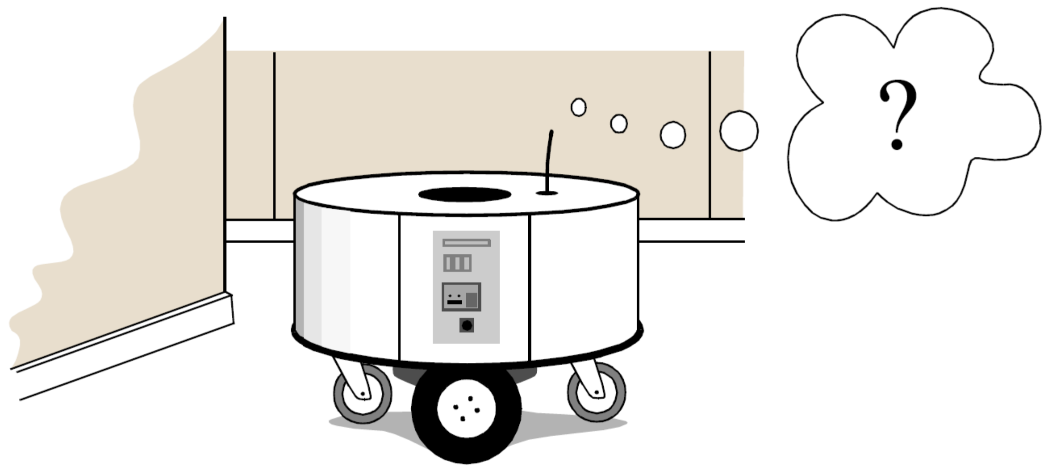




Outline

Overview of core concepts:

- Robot Motion
- Perception
- Localization and Mapping
- Navigation



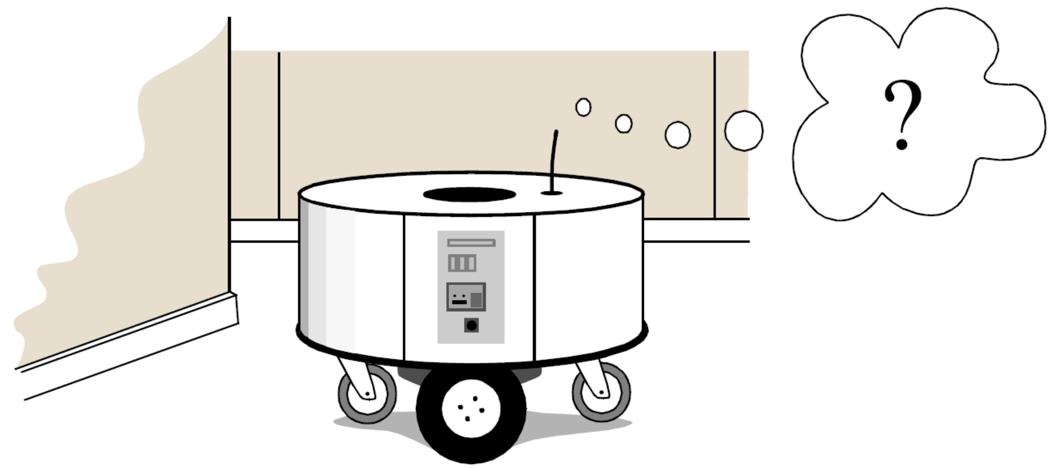
From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Assumption: let’s talk about the simplest type of mobile robots, wheeled ground vehicles

Navigation

After we have a map, and the robot position, how to go from A to B?

- Path Planning
- Obstacle Avoidance
- Navigation Architecture



Path Planning approaches

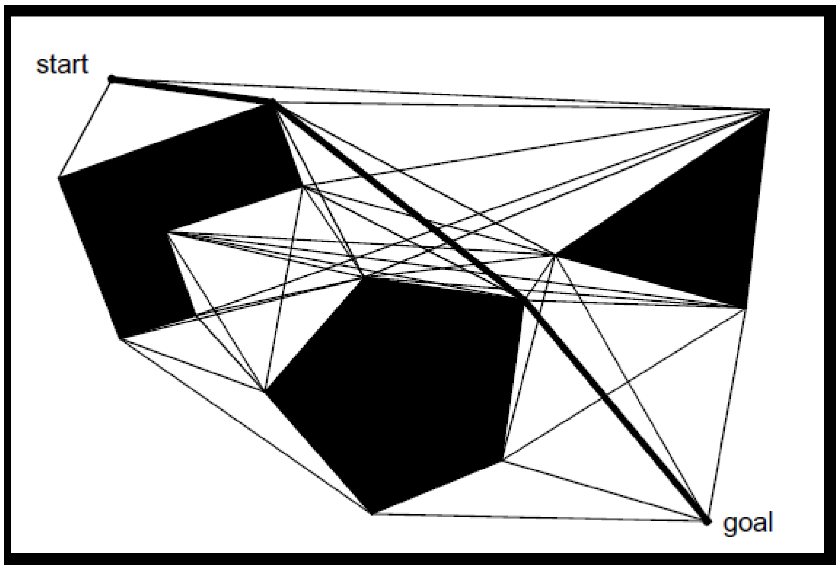
Once we have the map, we have to compute a set of states for finding the path that the robot can execute.

However, as we've seen, we have to provide a proper formulation for this problem:

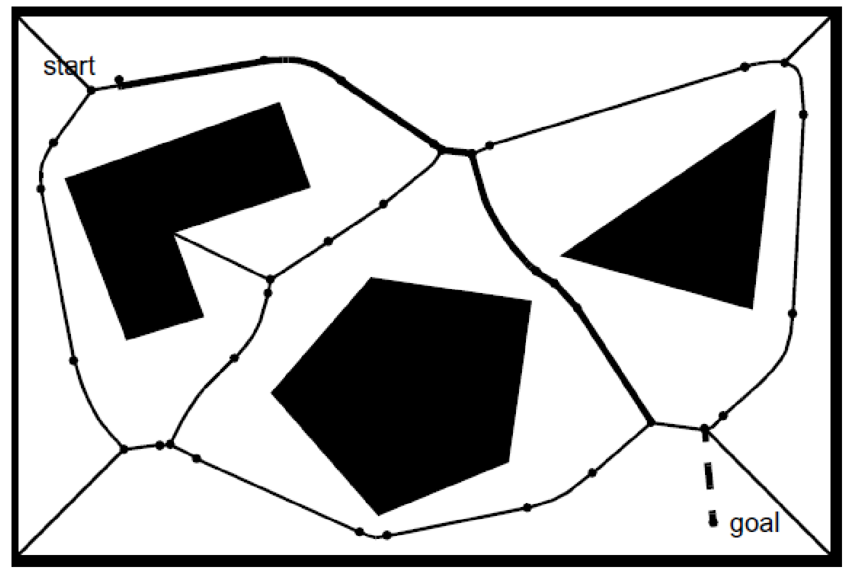
- Road map: identify a set of routes within the free space
- Potential field: impose a mathematical function over the space
- Cell decomposition: discriminate between free and occupied cells

Road map path planning

Idea: develop a network of roads / paths along the environment using a decomposition of the robot traversable free space.



Visibility Graph

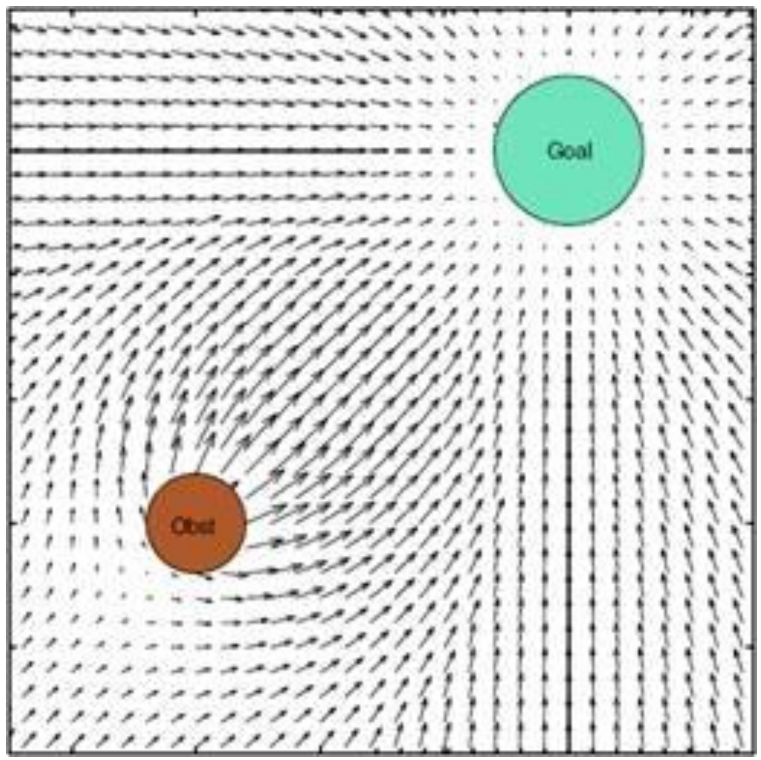


Voronoi Decomposition

From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

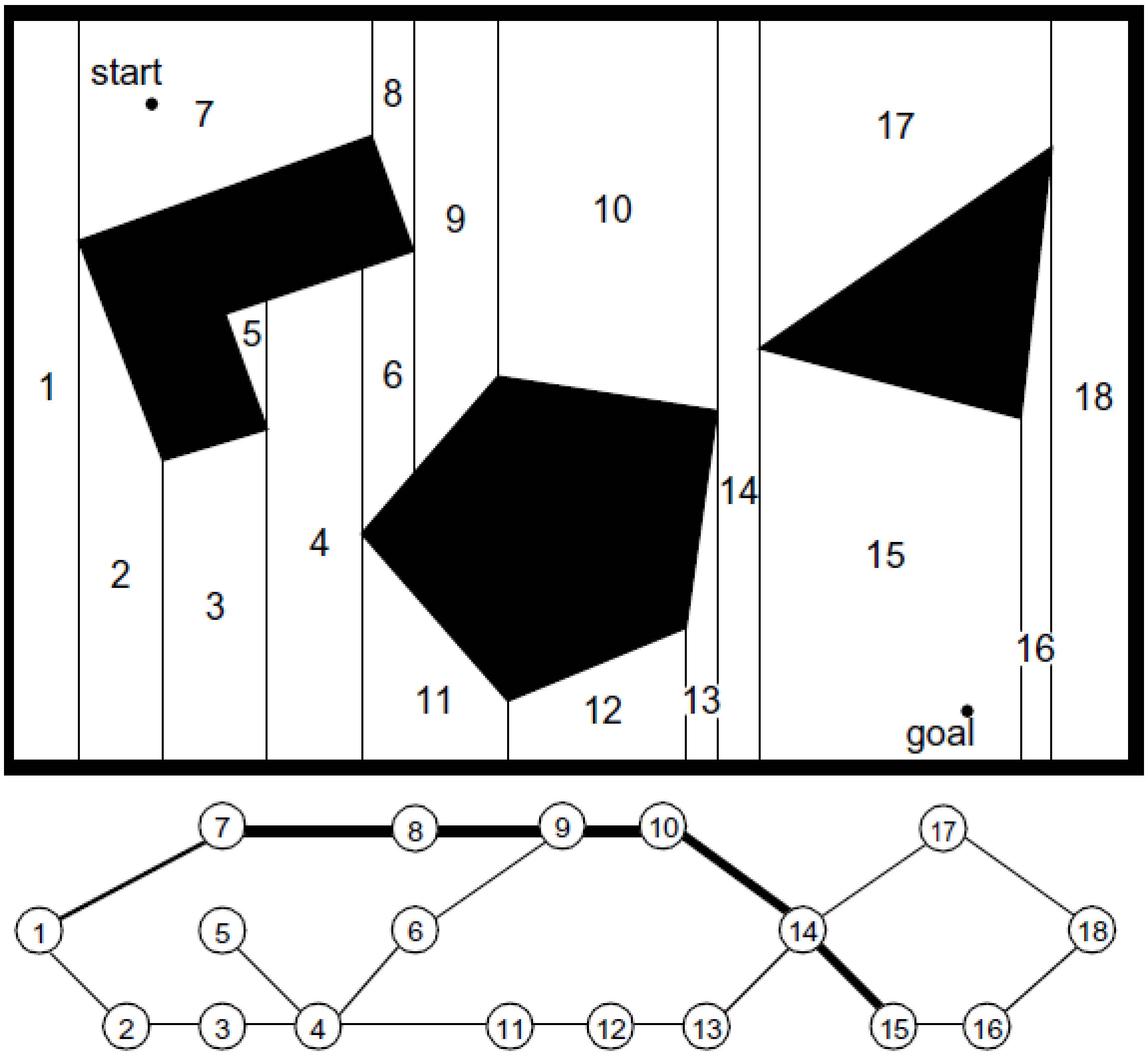
Potential field path planning

Idea: put an attractive artificial potential field on the goal, a repulsive one on obstacles, let the robot follow these simulated forces



from https://www.cs.mcgill.ca/~hsafad/robotics/

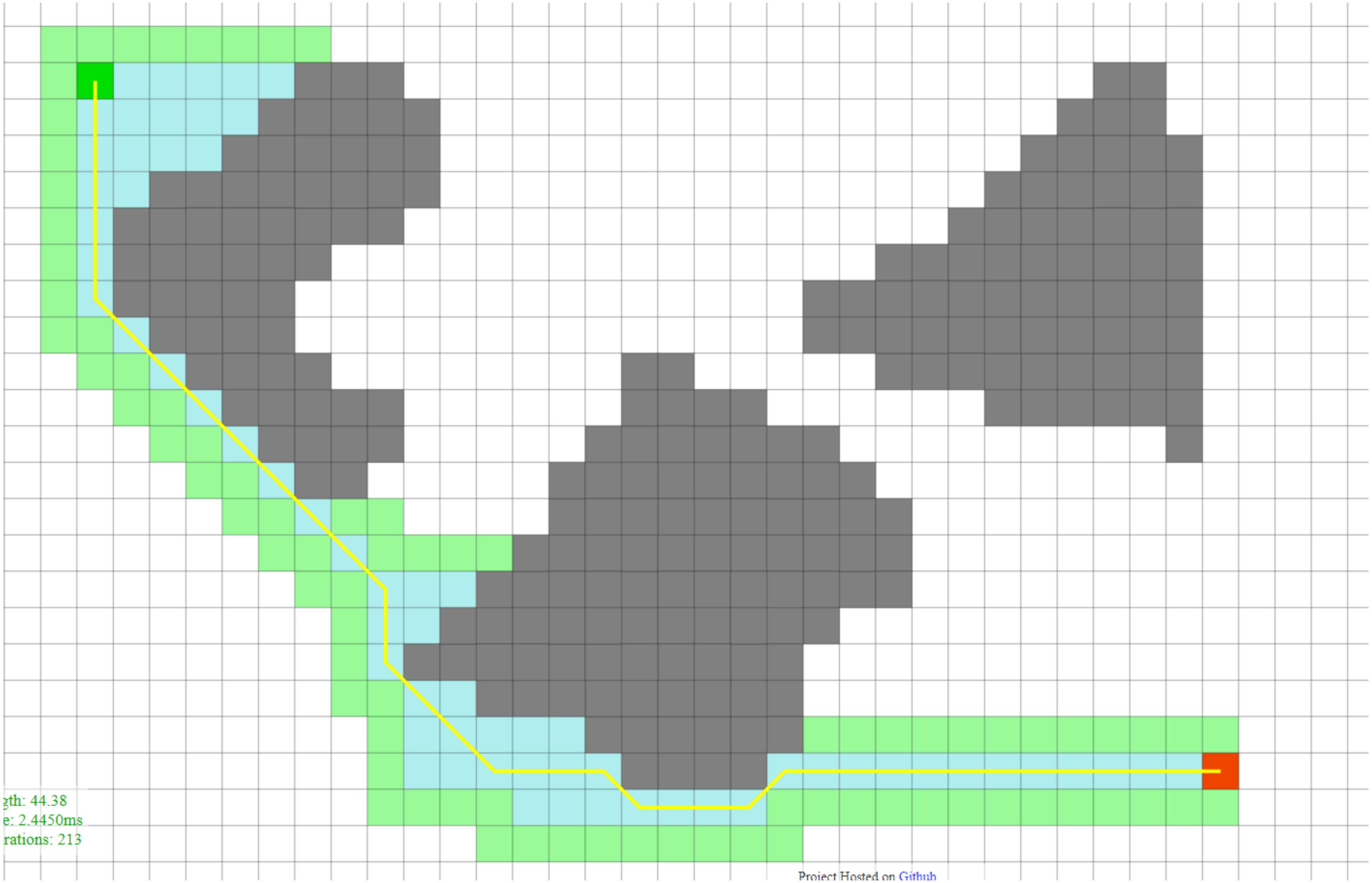
Cell decomposition path planning



From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

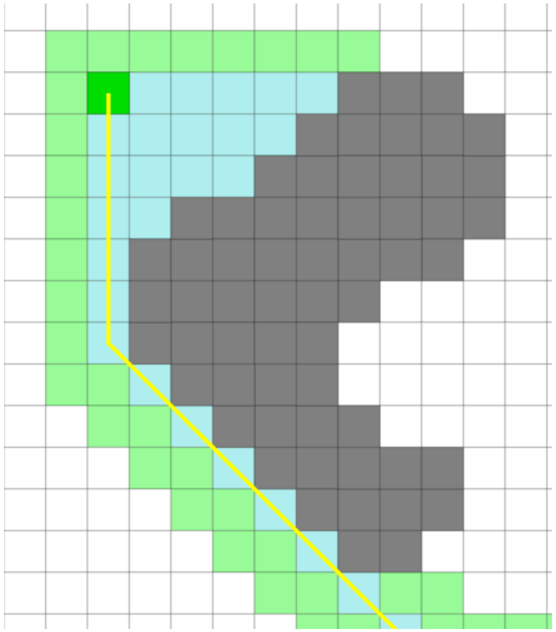
Approximate cell decomposition

This is what is usually done: path planning on a grid map



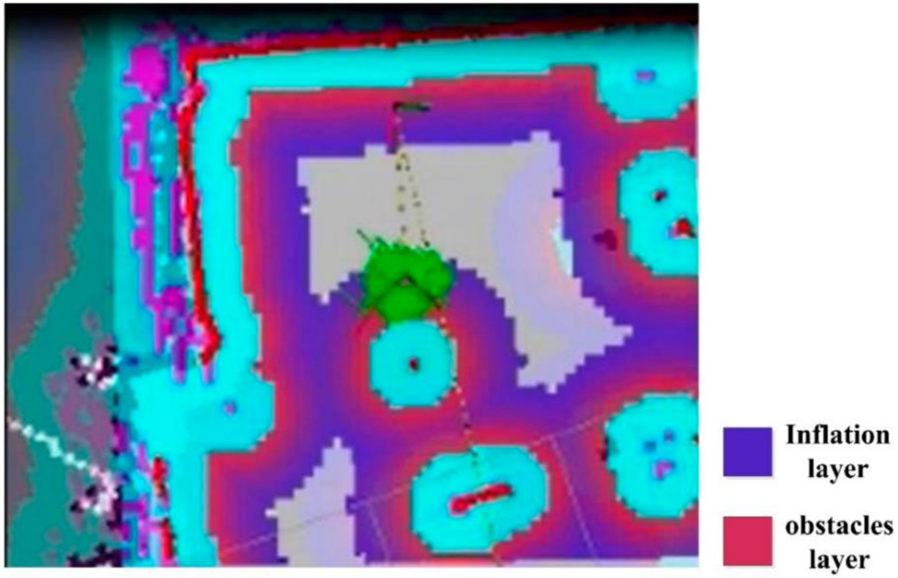
Here: A* for solving the search problem, Manhattan distance as h()

Obstacle avoidance



What happens if the robot is bigger than 1 cell (e.g., a 2x2 cell)? Shall we allow trajectories that are that close to the obstacles?

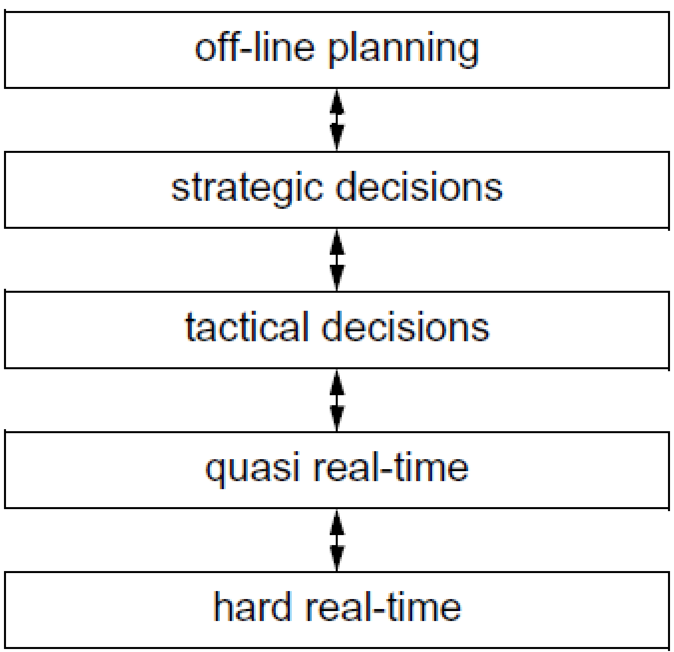
Several techniques are used for performing obstacle avoidance. Examples: inflating either the obstacle (considering the robot as a point) or the robot (allowing the robot to plan trajectories that goes across the obstacle.

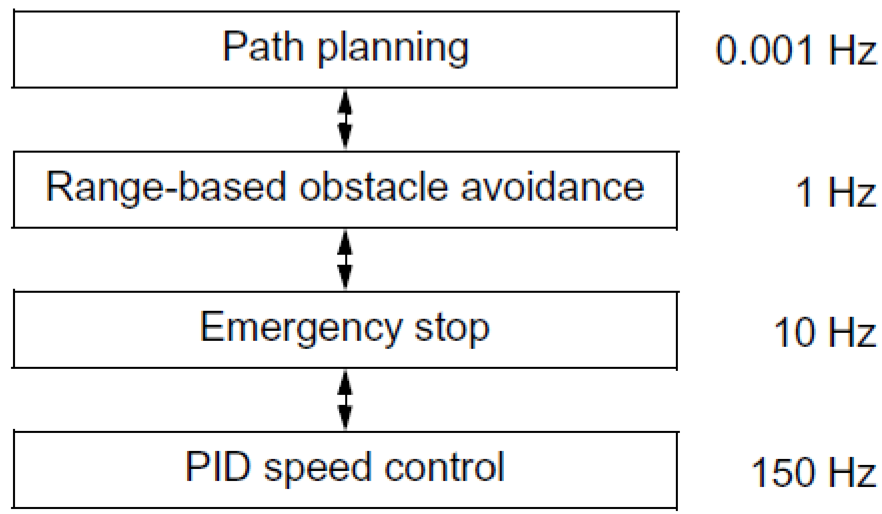


Navigation Architecture

Navigation is a task that requires both hi-level planning and low-level control, reacting to changes in the environment.

We can organize modules of the robot according to different hierarchies, as performing a temporal decomposition:

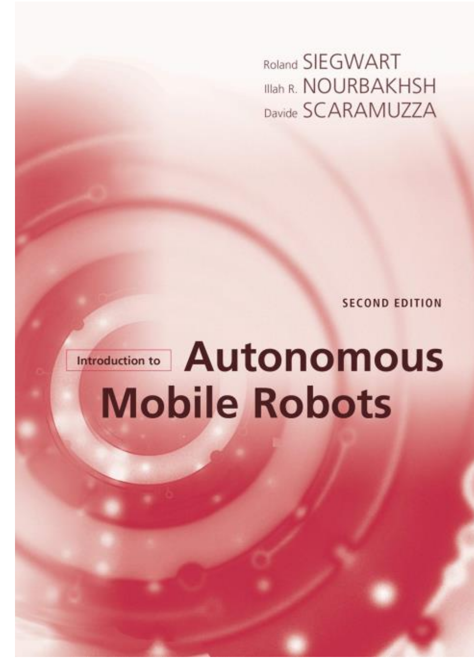




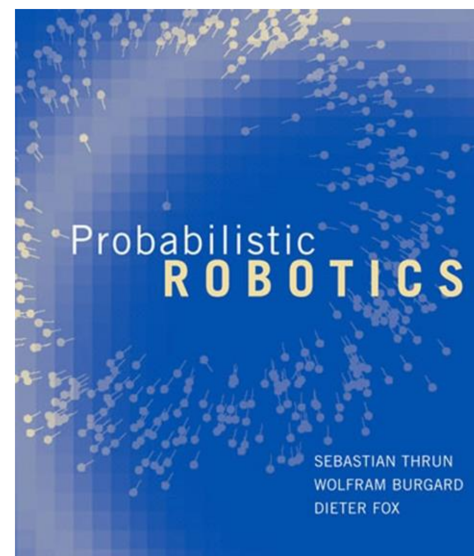
From Siegwart,Introduction to Autonomous Mobile Robots, MIT Press 2004

Sources

Roland Siegwart, Illah Nourbakhsh,
Davide Scaramuzza
Introduction to Autonomous Mobile Robots, II ED,
MIT Press, 2011



Sebastian Thrun, Wolfram Burgard, Dieter Fox,
Probabilistic Robotics,
MIT Press, 2006



Sistemi Intelligenti Avanzati
Corso di Laurea in Informatica, A.A. 2020-2021
Università degli Studi di Milan

Matteo Luperto
Dipartimento di Informatica
matteo.luperto@unimi.it