UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

Exploration

RO51 - Introduction to Mobile Robotics

Zhi Yan

April 7, 2022

https://yzrobot.github.io/

www.utbm.fr

What?

- Robotic exploration refers to the exploration of a completely or partially unknown space.
- Objectives of exploration: basic such as mapping (video), complex such as search and rescue (e.g. finding survivors) and interstellar migration.
- In contrast to SLAM, there is more emphasis on the (full or partial) autonomy of the robot, i.e. answering the question of where should I go.
- Also a fundamental problem in mobile robotics, closer to the public's imagination of robots than the SLAM problem.

Why?

- There are places where the greatest possible amount of information about the environment should be obtained in advance to ensure human safety, such as caves and mines.
- There are places humans should never go, such as disaster sites (nuclear accidents, fires, etc.)
- Curiosity about the universe.
- And many more.

How?

- Related to application scenarios and exploration goals.
- Since the problem contains unknowns and a certain degree of requirements for robot autonomy, it can be seen as SLAM plus decision-making from the operational level.
- In addition, the representation of the map (such as occupancy grid map) is a key, as it is an important basis for robot decision-making.

A brief chronicle before the formal discussion

- 1985: Paper "High Resolution Maps from Wide Angle Sonar" presented by H. Moravec and A. Elfes @ ICRA. => occupancy grid map
- 1997: Paper "A Frontier-based Approach for Autonomous Exploration" presented B. Yamauchi @ CIRA. => frontier-based exploration
- 2000: Paper "Collaborative multi-robot Exploration" presented W. Burgard *et al.* @ ICRA. => pioneer work on multi-robot exploration
- 2002: Paper "Multi-robot Exploration Controlled by a Market Economy" presented R. Zlot *et al.* @ ICRA. => method developed for MER-A and MER-B
- 2010: "Sampling-based Multi-robot Exploration" my first scientific paper addresses the problem of exploration waypoint generation for multiple robots.

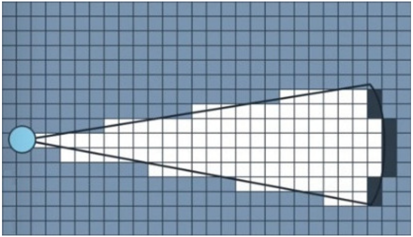
Occupancy grid map

What does it look like?

Occupancy grid map

What does it look like?

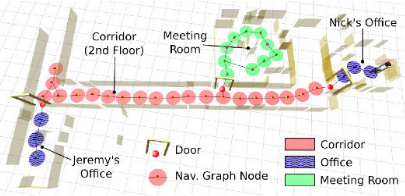
- Use (evenly spaced) grids to represent maps (like some traditional wargames).
- Each grid can have three different states: occupied (obstacle), not occupied (free space), not measured (unknown space).
- The grid state estimate is based on the posterior probability.
- It is a type of metric map (other metric maps including feature map, point cloud map, etc.).



Occupancy grid map

Different maps in mobile robotics

- **Metric map**: Has true physical dimensions, commonly used for SLAM, mapping, localization, and small-scale path planning.
- **Topological map**: Does not have the real physical size, but only represents the connectivity and distance of different locations (e.g subway network), often used for large-scale robot path planning.
- **Semantic map**: Can be seen as a labeled metric map.



Semantic map

Topological map

Occupancy grid map

Why

- **Occupancy**: A cliché: Sensors are not always accurate.
- **Grid**: Modelling continuous spaces is computationally complex and intensive. => discretize the space

Occupancy grid map

How

- Commonly used for ranging sensors (e.g. sonar, radar, lidar, etc.).
- Let's look at the algorithm first (a top-down way of thinking):

Algorithm occupancy_grid_mapping($\{l_{t-1,i}\}, x_t, z_t$):

 for all cells m_i *do*
  if m_i *in perceptual field of* z_t *then*
   $l_{t,i} = l_{t-1,i} + \mathbf{inverse\_sensor\_model}(\mathrm{m}_i, x_t, z_t) - l_0$
  else
   $l_{t,i} = l_{t-1,i}$
  endif
 endfor
 return $\{l_{t,i}\}$

Courtesy: *Probabilistic robotics*

Occupancy grid map

How

- The goal of an occupancy mapping algorithm is to estimate the posterior probability over maps given the data:

$$p(m \mid z_{1:t}, x_{1:t})$$

  where m is the map, $z_{1:t}$ is the set of measurements from time 1 to t, and $x_{1:t}$ is the set of robot poses from time 1 to t.

- The posterior of a map is approximated by factoring it into:

$$p(m \mid z_{1:t}, x_{1:t}) = \prod_i p(m_i \mid z_{1:t}, x_{1:t})$$

  where m_i denote the grid cell with index i.

Occupancy grid map

How

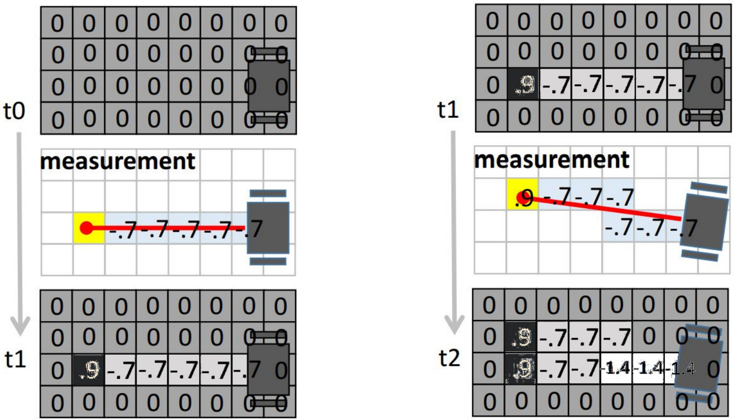
- $l_{t,i} = \log \frac{p(m_i|z_{1:t},x_{1:t})}{1-p(m_i|z_{1:t},x_{1:t})}$ (a log-odds representation, derivation will be covered in TD)
- **inverse_sensor_model**$(\cdot)$: the inversion of the posterior probability of the state through the result, i.e.

$$\textbf{inverse\_sensor\_model}(m_i, x_t, z_t) = p(m_i \mid z_t, x_t)$$

  In human language, the robot has obtained the position of the grid (through measurements), then infers the probability of that grid being occupied.
- $l_0 = \log \frac{p(m_i=1)}{p(m_i=0)} = \log \frac{p(m_i)}{1-p(m_i)} = \log \frac{0.5}{1-0.5} = 0$ (the constant l_0 is the prior of occupancy represented as a log-odds ratio)

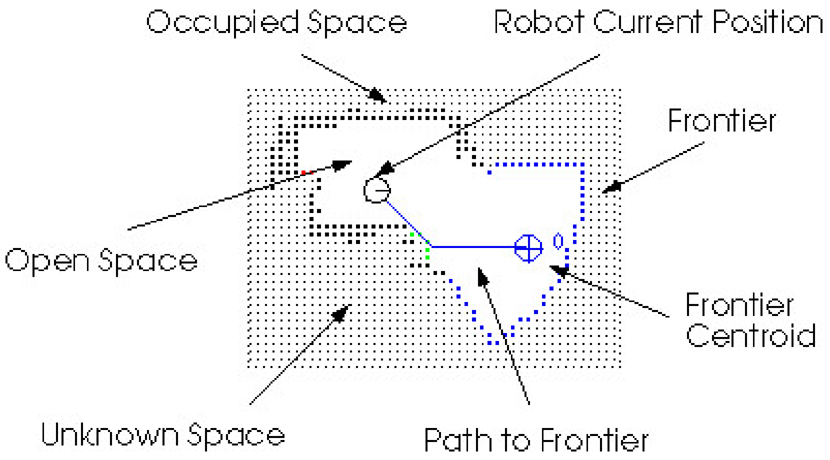
Occupancy grid map

How

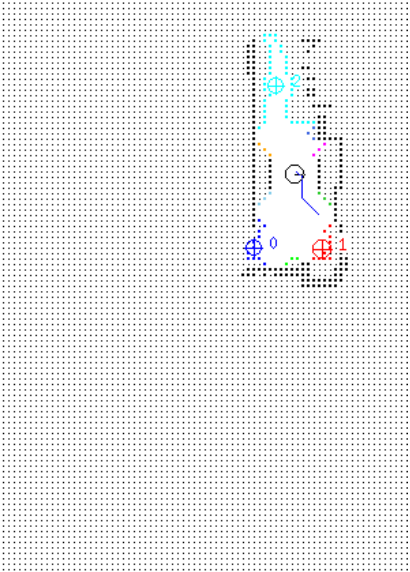


- Just an example: $l_{free} = -0.7$, $l_{occ} = 0.9$

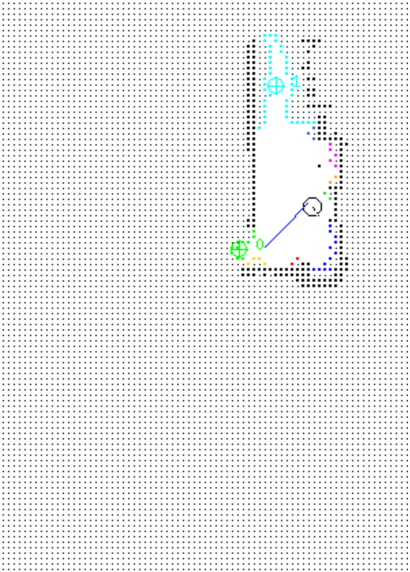
Frontier-based exploration



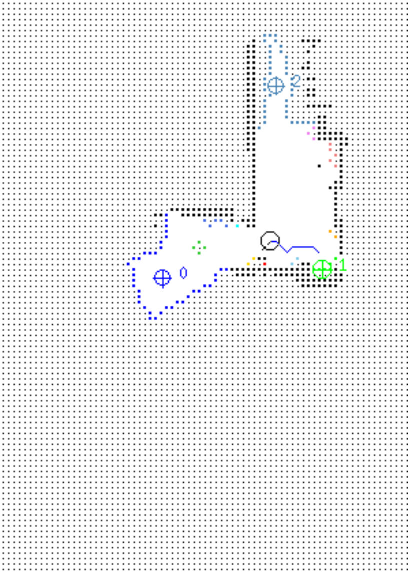
Frontier-based exploration



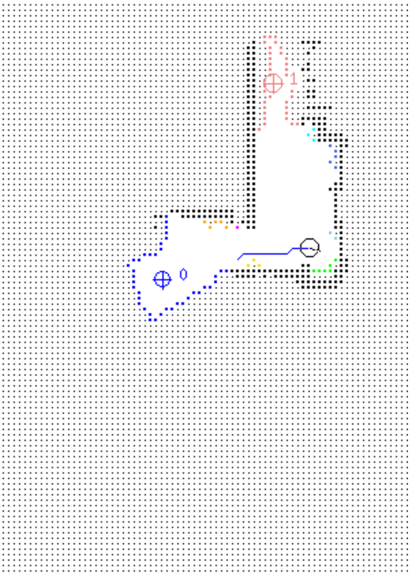
Frontier-based exploration



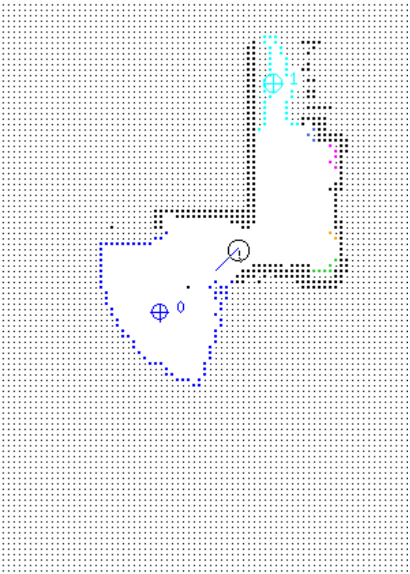
Frontier-based exploration



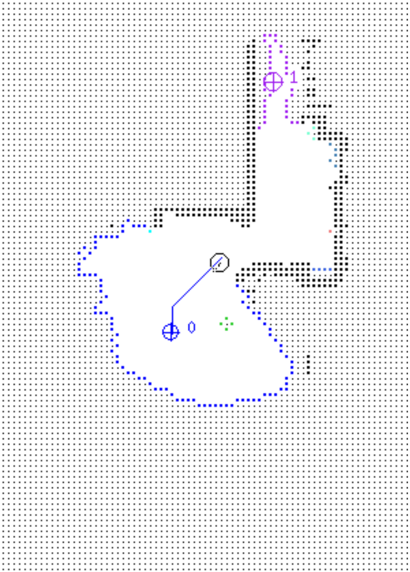
Frontier-based exploration



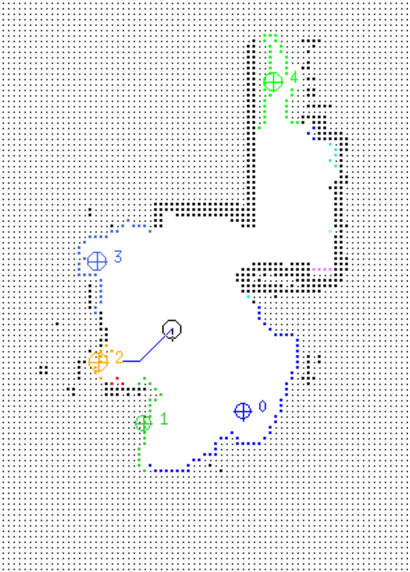
Frontier-based exploration



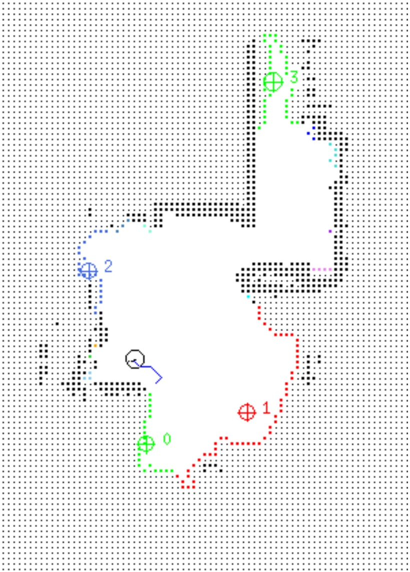
Frontier-based exploration



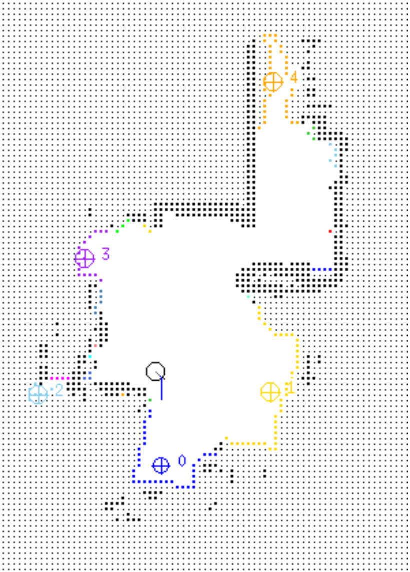
Frontier-based exploration



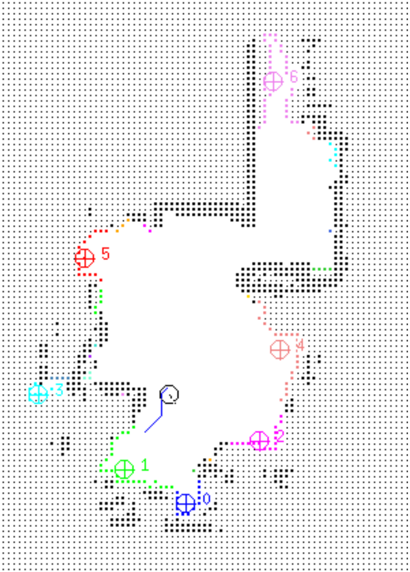
Frontier-based exploration



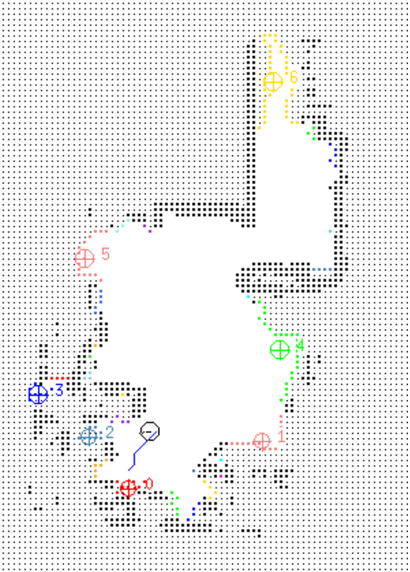
Frontier-based exploration



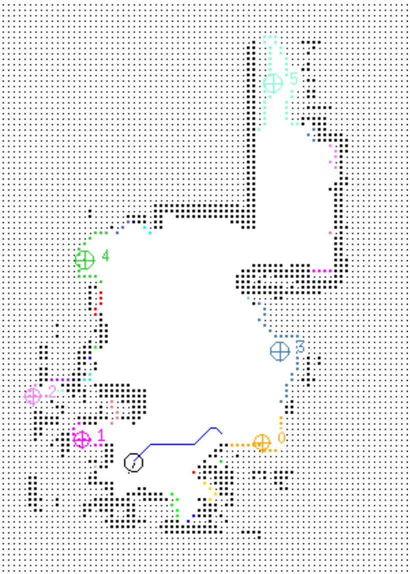
Frontier-based exploration



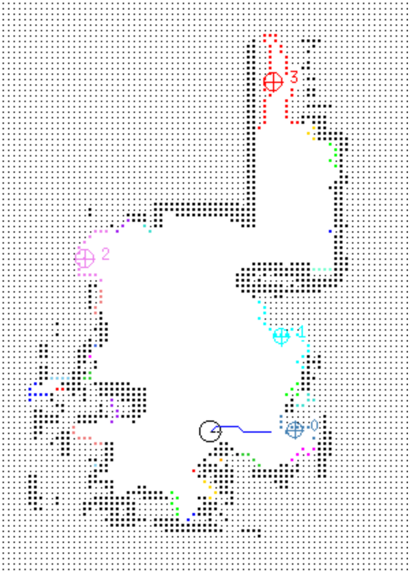
Frontier-based exploration



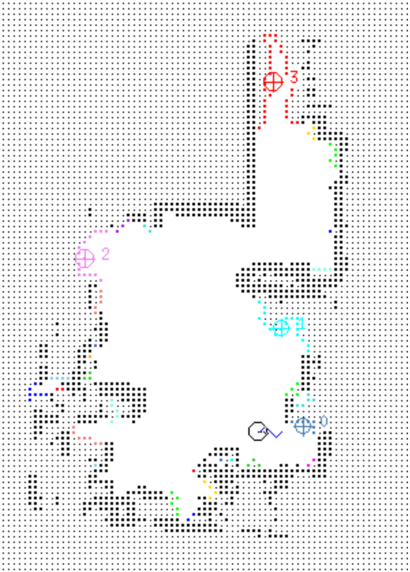
Frontier-based exploration



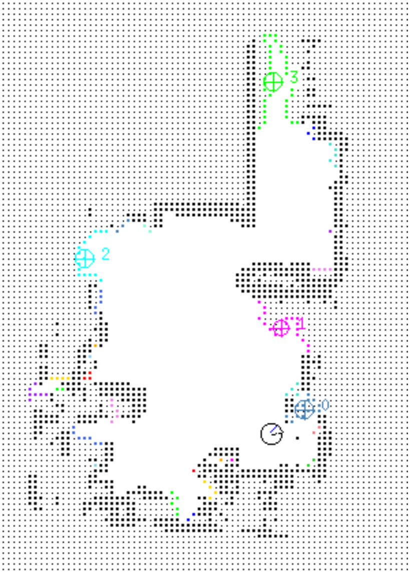
Frontier-based exploration



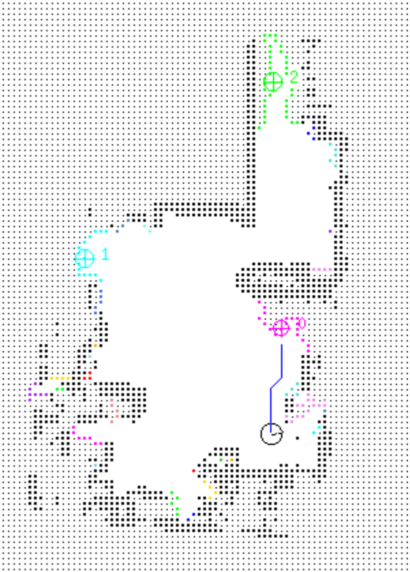
Frontier-based exploration



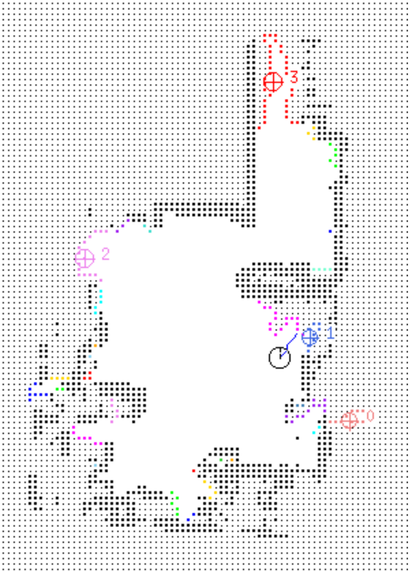
Frontier-based exploration



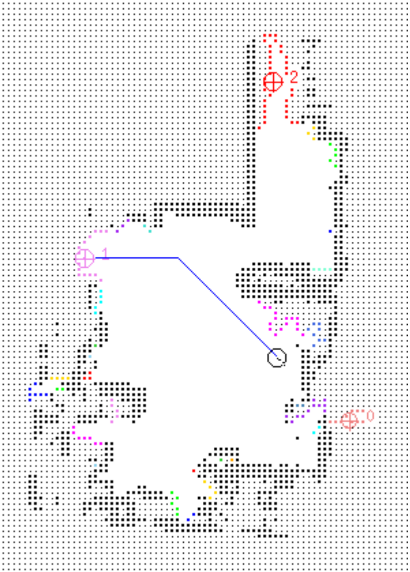
Frontier-based exploration



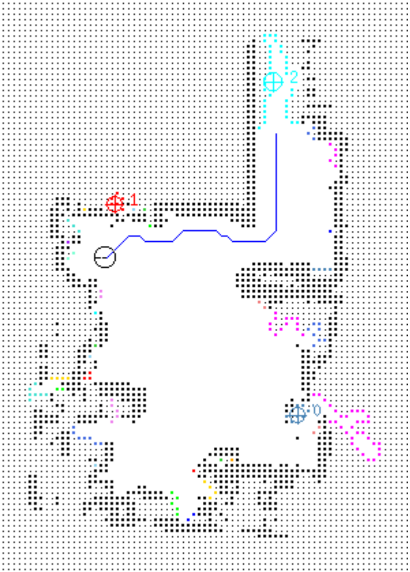
Frontier-based exploration



Frontier-based exploration



Frontier-based exploration



Frontier-based exploration

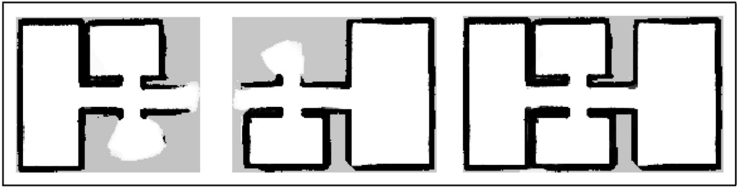
- Judge the method without the background of the times => non-objective
- Definition of the frontier: regions on the boundary between open space and unexplored space.
- Method for extracting frontier: similar to edge detection in computer vision.
- A robot does not choose frontiers that cannot be traversed (e.g. based on its physical size).
- If there are multiple frontiers, the robot chooses the closest and unvisited one.
- The robot ends exploration when all frontiers have been visited.

Multi-robot exploration

- Exploration usually has requirements on time efficiency and energy consumption, because:
  - time is money, time is life,
  - the power of the battery is limited,
- Modern exploration also requires completeness, richness and redundancy (variety) of environmental information.
- Deploying **multi-robot systems** (c.f. Lecture 9) for exploration:
  - Robot team composition (homogeneous or heterogeneous) according to mission needs
  - Cooperative strategy when performing tasks
  - Merging of maps (including communication issues between robots)
  - etc.

Multi-robot exploration

- Integration of two individual maps (from two mobile robots) into a single global map:



- Suppose there are N robots which all have an individual map m_i, and let $P(occ^i_{x,y})$ denote the probability that the location $< x, y >$ in the global coordinate frame is occupied in the map of robot i, then

$$P(occ_{x,y}) = \frac{odds_{x,y}}{1 + odds_{x,y}}$$

where $odds_{x,y} = \prod_{i=1}^{N} odds^i_{x,y}$ and $odds^i_{x,y} = \frac{P(occ^i_{x,y})}{1-P(occ^i_{x,y})}$

Multi-robot exploration

- Probabilistic fusion of multiple maps: why **odds**, but not addition or multiplication (c.f. Lecture 5)?
  - Addition: different possible outcomes of the same event (e.g. there are multiple methods to complete a task and each method can complete this task independently).
  - Multiplication: different steps of the same event (e.g. it takes n steps to complete a task).
  - Odds: provides a measure of the likelihood of a particular outcome and takes into account multi-sensor interactions (for the derivation process, see Moravec's classic paper published in 1988: *Sensor Fusion in Certainty Grids for Mobile Robots*).

Multi-robot exploration

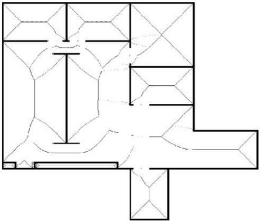
Task allocation: **Contract Net Protocol** (CNP), a task-sharing protocol introduced in 1980 by Reid G. Smith:

1. The protocol is initialized by a task manager, who sends a call-for-proposals to the contractors.
2. The contractors can send either a proposal if they are interested or a reject if they are not.
3. The manager chooses among the proposals the one that suits it best, and sends to the corresponding contractor an accept.
4. Once the contract has been accomplished, the contractor informs the manager using an inform message.

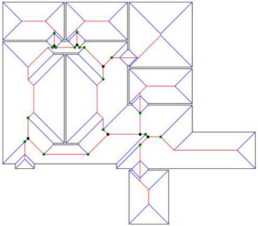
	Market-based approach	**Auction-based approach**	**Trade-based approach**
Negotiation (communication) model	publish / subscribe	publish / subscribe	apply / allocate
Task allocation algorithm	greedy algorithm (first-price auction)	greedy algorithm (first-price auction)	greedy algorithm (first-price auction)
Task allocation ability per iteration (trading round)	single task	single task	multiple tasks
Role determination of the robots	voluntary	voluntary	negotiation
Utility consideration	cost and benefit	cost	cost
Task reassignment	allowed	not allowed	allowed
Communication complexity	$O(1)$ / bidder, $O(n)$ / auctioneer	$O(1)$ / bidder, $O(n)$ / auctioneer	$O(1)$ / buyer, $O(n)$ / seller
Computation complexity	$O(n)$	$O(n)$	$O(n)$

Completely vs. partially unknown spaces

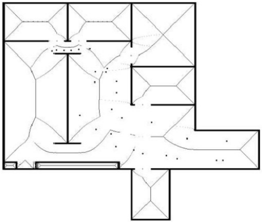
- **Completely unknown space**: the robot (not human) does not have any information (a priori) about the target environment.
- **Partially unknown space**: the robot has partial information about the target environment, such as a floor plan.
  - This is often very useful for search and rescue missions.



Voronoi Diagram



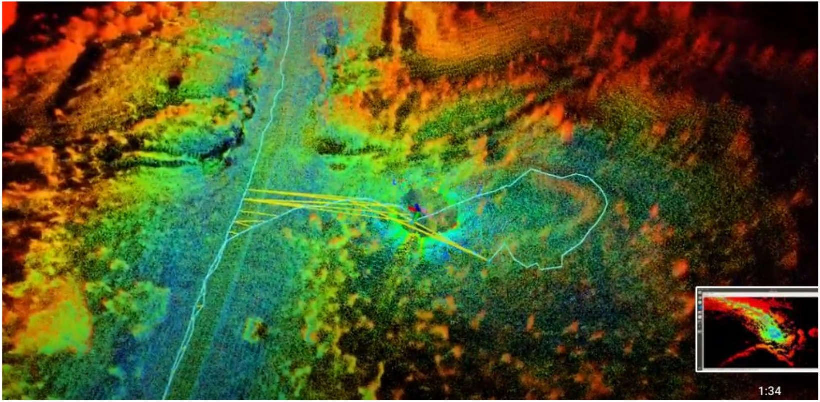
Straight Skeleton



Sampling-Based

Structured vs. unstructured environments

- **Structured environment**: building interior
- **Unstructured environment**: fields, mines, Mars, etc.



Lidar mapping in field test at Longshaw National Trust

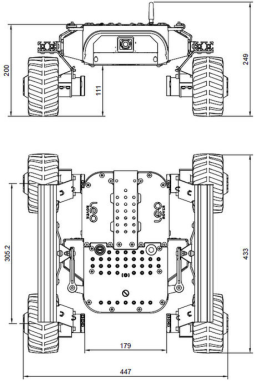
Rover@UTBM project

In a Mars-like environment:

- The road situation is very complex. => challenging the stability of the robot locomotion (c.f. Lecture 2)
- It is difficult to rely solely on ranging sensors (e.g. lidar, radar, sonar) to explore. => lack of landmarks and sensor readings

Rover@UTBM project

Size and performance



Size

Weight: 6,5 kg
Dimensions: 447x433x249 mm
Payload capacity: ca. 5 kg

Upper platform mounting dimensions

Dimensions: 299 x 183 mm
Hole grid: 18 x 15 mm
Holes: 40 x Φ 7mm + 22 x Φ 5,5mm

Performance

Estimated maximum obstacle size: 70 mm
Protection rating: complies with IP66 (not certified)

Run time: Estimated 4 hrs of nominal driving

Connection range: Up to 100m (with live video stream)

Camera

Camera resolution: 5 MPx
Lens: Fisheye with 170 deg field of view (IR non-filtered; night-vision allowed)

Network

WiFi 2.4 GHz access point with external antenna
WiFi 2.4 GHZ + 5 GHz on internal RPi antennas for connectivity

Wheels

Motors: 4 x in-hub DC motor with 73.2:1 planetary gearbox and 12 CPR encoder
Wheel diameter: 130 mm
Tire material: rubber with foam insert (non-pneumatic)

Battery

Voltage: 11.1 V DC
Capacity: 5000 mAh
Type: Li-Ion with internal PCM
Short-circuit, overcurrent and overdrain safety features
Max. current: 8A (total for the whole Rover)

Speed

Max. linear speed: ca. 0.4 m/s
Max. angular speed: ca. 60 deg/s

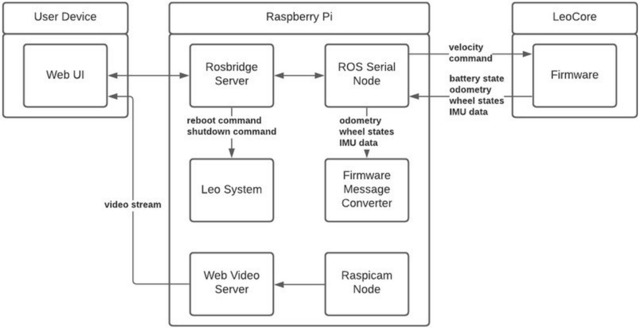
Rover@UTBM project

Software

Operating system: Ubuntu 20.04 + Robot Operating System

Ready-to-go UI located under '10.0.0.1' when using standard Leo Software Image.

Open source firmware



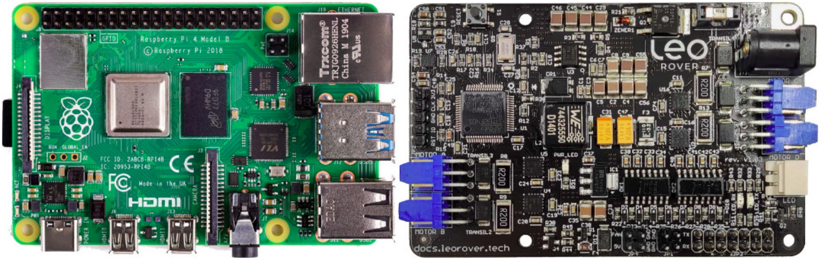
Electronics

RaspberryPi 4B 2GB (or higher) as the main computer

LeoCore as real-time microcontroller: STM32F4 (@84MHz, 64KB RAM, 256KB Flash)

Rover@UTBM project

- Higher level control (like cerebrum): Raspberry Pi, running Linux and ROS.
- Lower level control (like cerebellum): LeoCore board, runing firmware (C/C++),providing:
  - differential drive controller (*cmd_vel* interface)
  - wheel states monitoring (*joint_states* interface)
  - battery voltage monitoring
  - wheel odometry calculation
  - IMU support



Rover@UTBM project

- The outer diameter of the tires affects the rover's speed and force (slope/climbing ability).
  - higher outer diameter = higher speed & lower force
  - lower outer diameter = lower speed & higher force
- The stiffness of the tires affects the rover's turning resistance and ability to climb obstacles.

$$\text{Rover Speed } [m/s] = \frac{\text{wheel diameter } [m] \times \pi \times \text{motor speed } [rpm]}{60}$$

$$\text{Rover Force } [N] = \frac{4 \times \text{motor torque } [Nm]}{\text{wheel diameter } [m]}$$

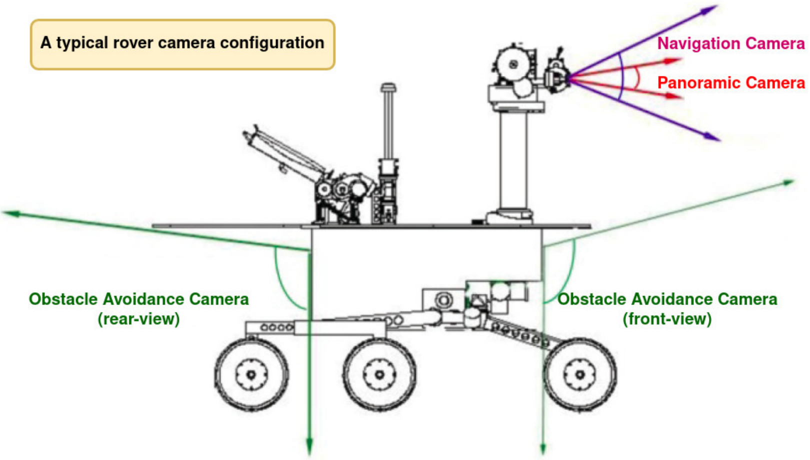
Mars rover

Landed rover

Landing date	Rover name	Active duration	Main sensor
1997-7-4	Sojourner (USA)	85 days	2x gray & 1x color cameras 5x laser stripe projectors
2004-1-4	Spirit / MER-A (USA)	2249 days	4x different cameras 1x thermal spectrometer
2004-1-25	Oppy / MER-B (USA)	5250 days	4x different cameras 1x thermal spectrometer
2012-8-6	Curiosity (USA)	still operational	17x different cameras
2021-2-18	Perseverance (USA)	still operational	19x different cameras 2x microphones
2021-5-15	Zhurong (China)	still operational	4x different cameras 1x ground-penetrating radar

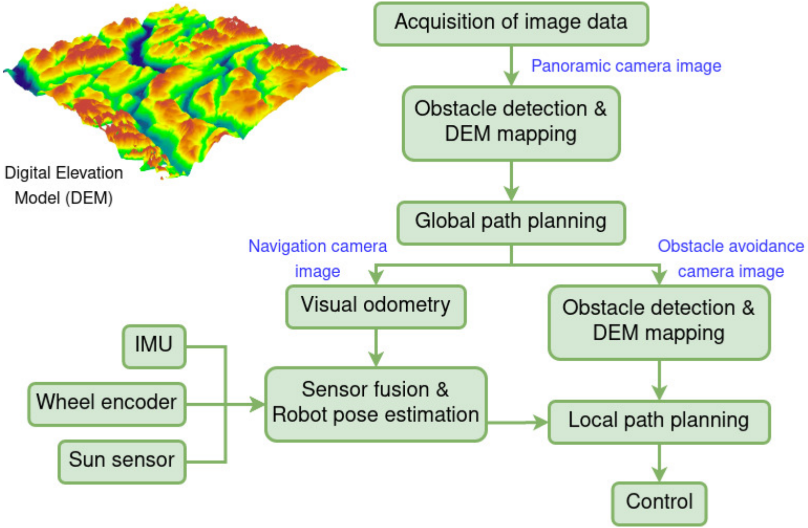
Mars rover

Cameras rule, lidar step aside



Mars rover

Basic flowchart of the exploration system



Summary

- Exploration: what, why and how.
- Occupancy grid map, frontier-based exploration, multi-robot exploration.
- Rover@UTBM project.
- The practice of exploration will be covered in TD and TP.
- This lecture pays more attention to cultivating students' interest in mobile robotics, especially its application in space exploration.
- Let's wrap up today's lecture with a video.

The end

Thank you for your attention!

Any questions?