Introduction to perception
Localization

Olivier Aycard

Professor

Grenoble INP - PHELMA

GIPSA Lab

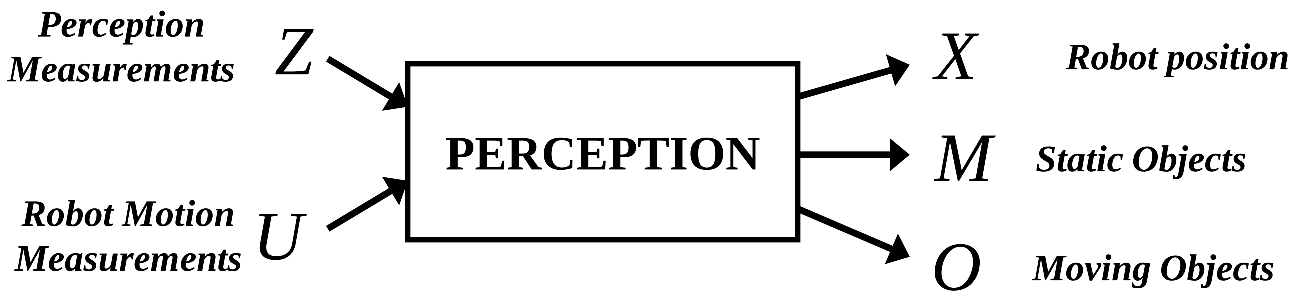
https://www.gipsa-lab.grenoble-inp.fr/user/olivier.aycard

olivier.aycard@grenoble-inp.fr

Outline

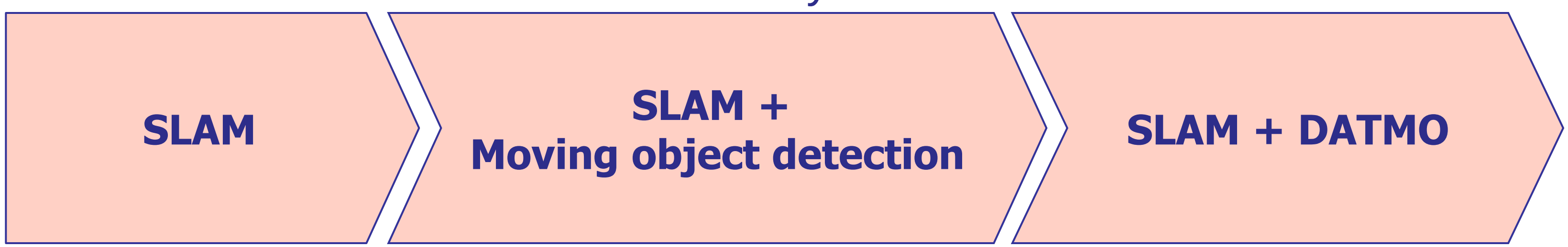
1. Introduction to localization
2. Requirements for localization
3. Localization algorithms

Problem statement



Static environments

Dynamic environments



$$P(X, M \mid Z, U)$$

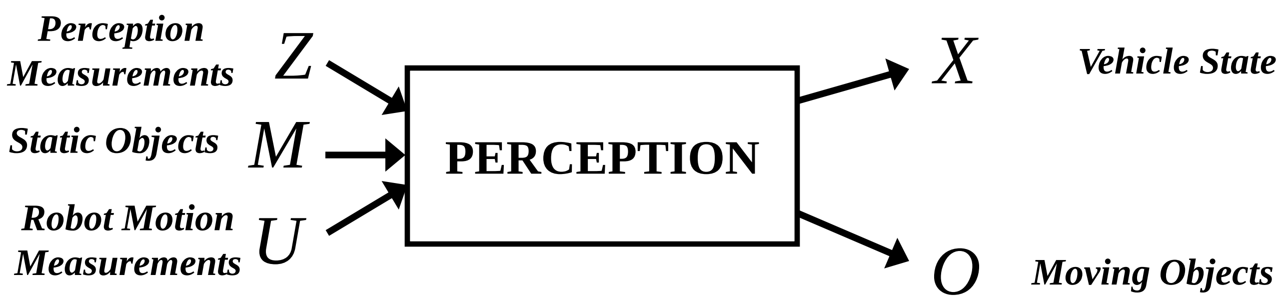
$$\begin{cases} Z = Z^{(s)} + Z^{(d)} \\ P(X, M \mid Z^{(s)}, U) \end{cases}$$

$$P(X, M, O \mid Z, U)$$

$$\begin{cases} P(X, M \mid Z^{(s)}, U) \\ P(O \mid Z^{(d)}) \end{cases}$$

Problem statement

If the map is known: a localization problem



Static environments | Dynamic environments

LOCALIZATION → **LOCALIZATION + Moving object detection** → **LOCALIZATION + DATMO**

$$P(X \mid Z, U, M)$$

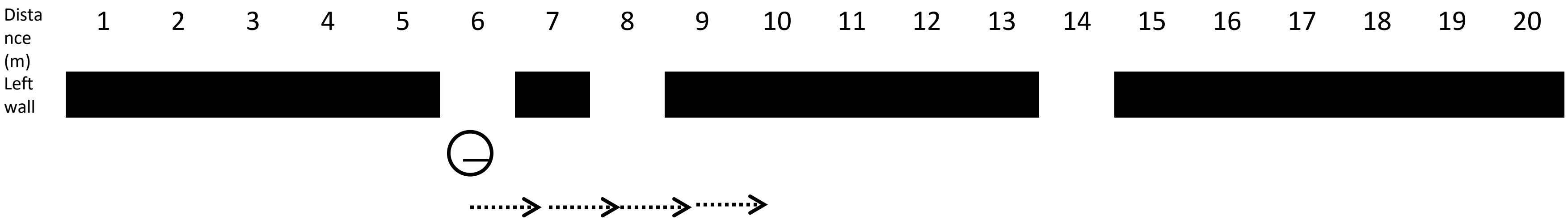
$$\begin{cases} Z = Z^{(s)} + Z^{(d)} \\ P(X \mid Z^{(s)}, U, M) \end{cases}$$

$$P(X, O \mid Z, U, M)$$

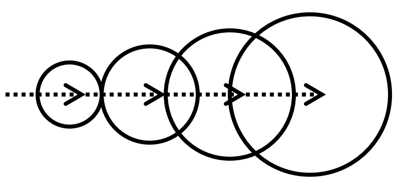
$$\begin{cases} P(X \mid Z^{(s)}, U, M) \\ P(O \mid Z^{(d)}) \end{cases}$$

Introduction to localization(1/3)

- While a mobile robot is moving in its environment, it needs to know its position in the environment.
- Lets take an example



- After 4 actions, robair should be at position 10
- …
- Odometry and motions have some uncertainties



The robot is lost !!!

Introduction to localization(2/3)

- Robair is equipped with a laser scanner to perceive its environment
- It should use the observation provided by the laser scanner to know its position

- After position 7, the laser scanner perceives a wall on the left side of robair: it is located at position 7 (if laser is perfect)
- Laser scanner has some uncertainties

Introduction to localization(3/3)

- The localization process is done in 2 steps:

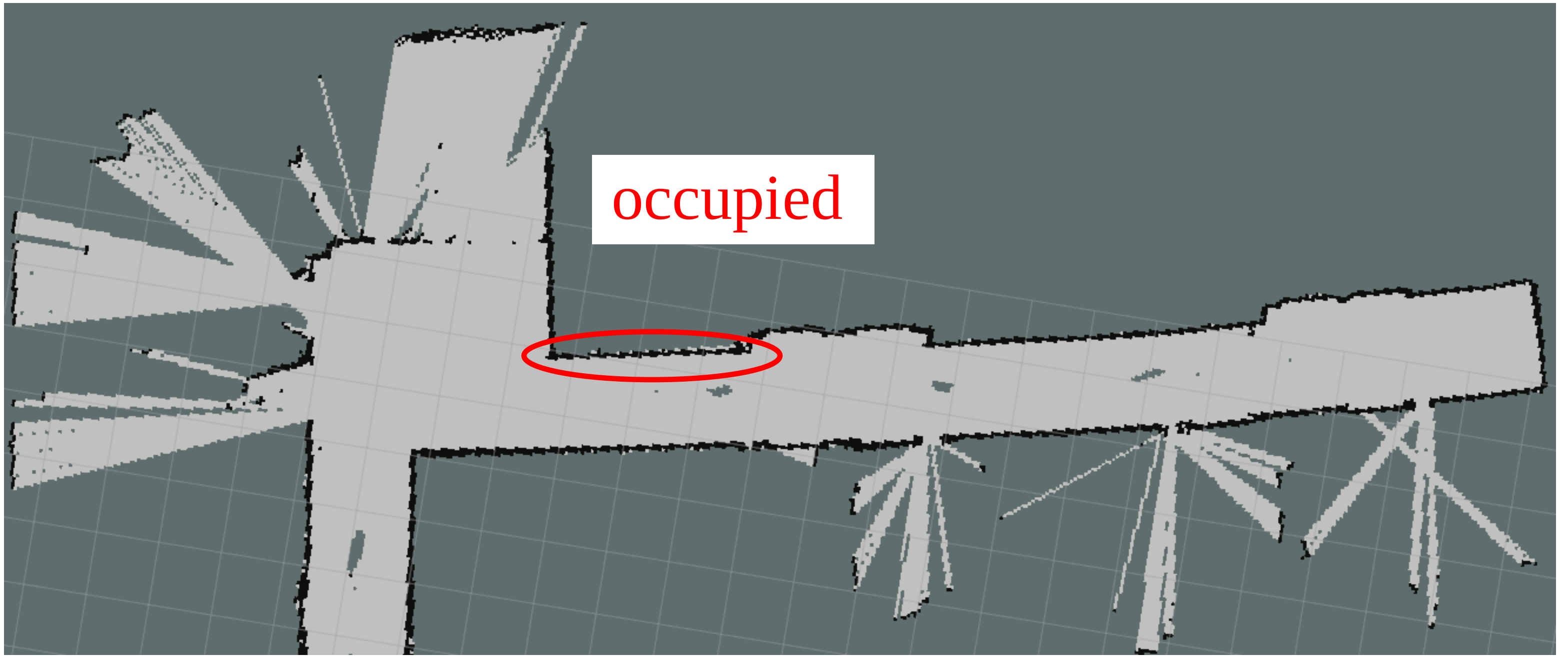
1. Robair moves: it estimates its motion (ie, odometry) to predict where it is:
   - Robair predicts where it could be after a motion: prediction phase
   - It should have a model of its motion and the associated uncertainty: motion/dynamic model
   - Prediction phase
2. Robair observes its environment: it improves its estimation of where it is, comparing/confronting its observation (ie, laser) with the prediction;
   - For each possible predicted position, Robair compares its observation with what it should see if it is located at this position: observation/sensor model
   - Estimation phase or confrontation between prediction and observation

Outline

1. Introduction to localization
2. Requirements for localization
3. Localization algorithms

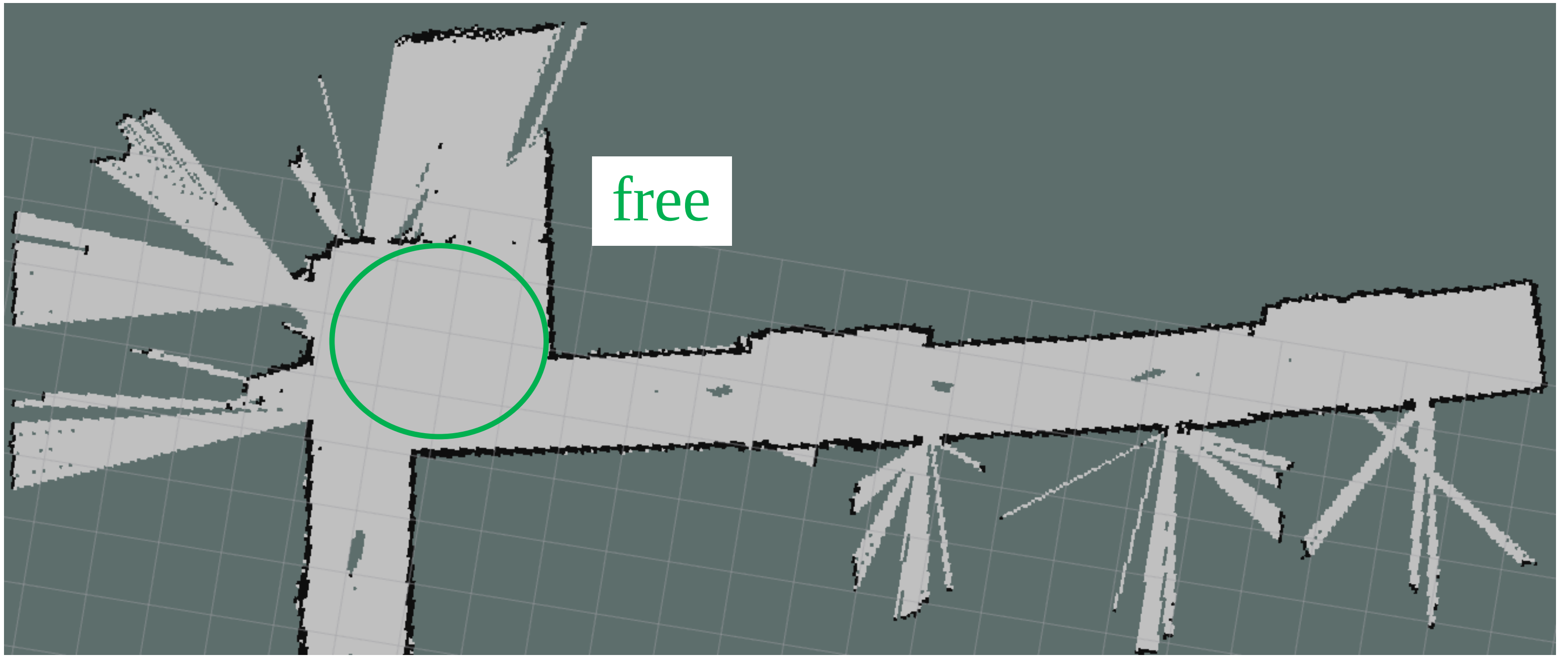
Requirements for localization: a map(1/4)

- There exist many representations for a map in robotics;
- Occupancy grid is the most used one;
  1. Discretization of the environment into cells;
  2. Each cell holds a probability value that the cell is occupied;
  3. Low level representation: close to laser data.



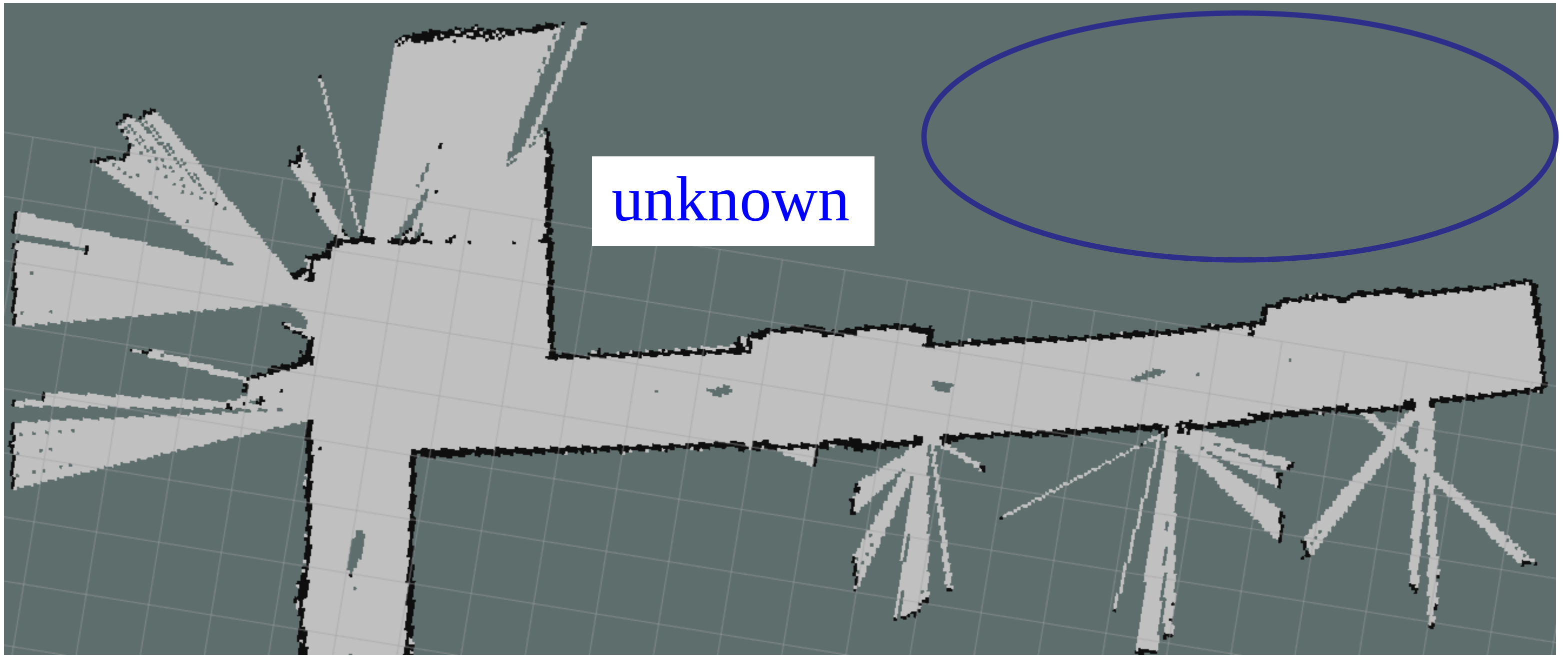
Requirements for localization: a map(2/4)

- There exist many representations for a map in robotics;
- Occupancy grid is the most used one;
  1. Discretization of the environment into cells;
  2. Each cell holds a probability value that the cell is occupied;
  3. Low level representation: close to laser data.

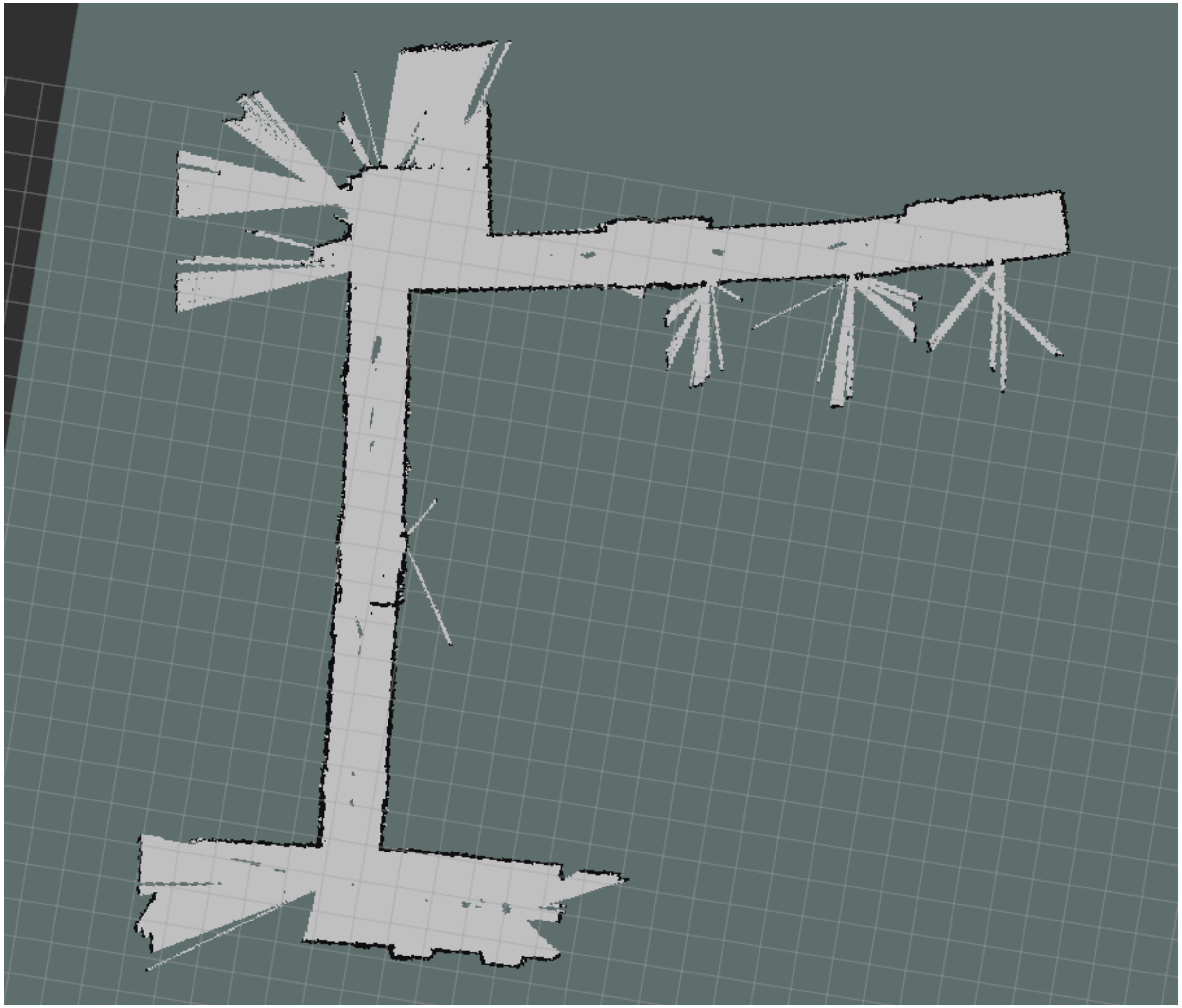


Requirements for localization: a map(3/4)

- There exist many representations for a map in robotics;
- Occupancy grid is the most used one;
    1. Discretization of the environment into cells;
    2. Each cell holds a probability value that the cell is occupied;
    3. Low level representation: close to laser data.

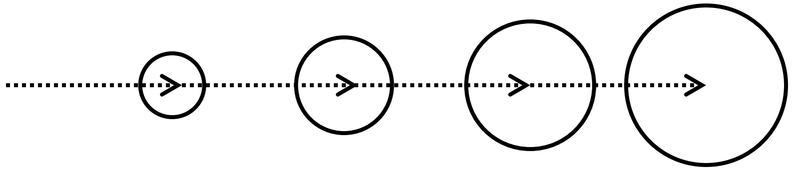


Requirements for localization: a map(4/4)



Requirements for localization: a dynamic/ motion model(1/2)

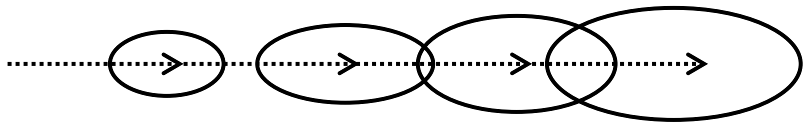
- This model gives an estimation of where robair is after doing an action



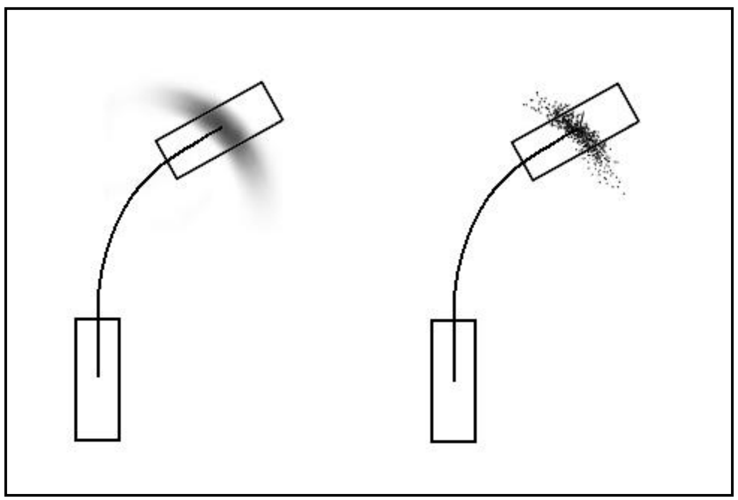
- After several translations (without localization), the uncertainties are accumulated
- Drift problem: robair is lost

Requirements for localization: a dynamic/motion model(2/2)

- A more realistic model for translations



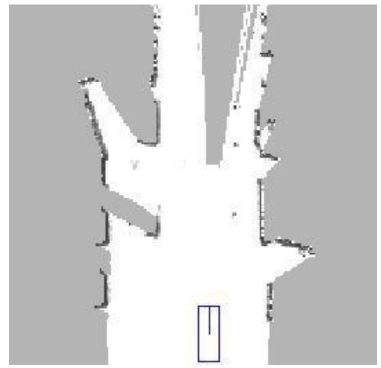
- Combination of translations and rotations

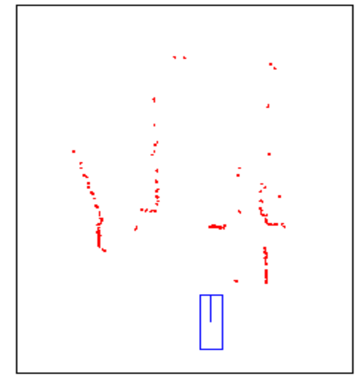


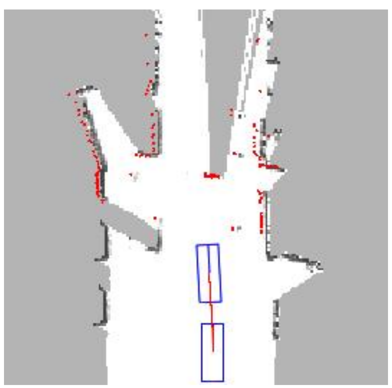
Probabilistic motion model and its sampling version

Requirements for localization: a sensor/observation model[Dung'07](1/5)

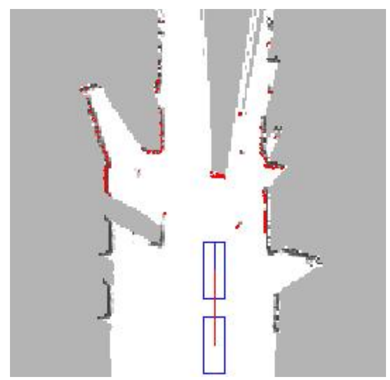
- For some predicted positions, we will compare what robair observes with what it should observe







score = 0.21



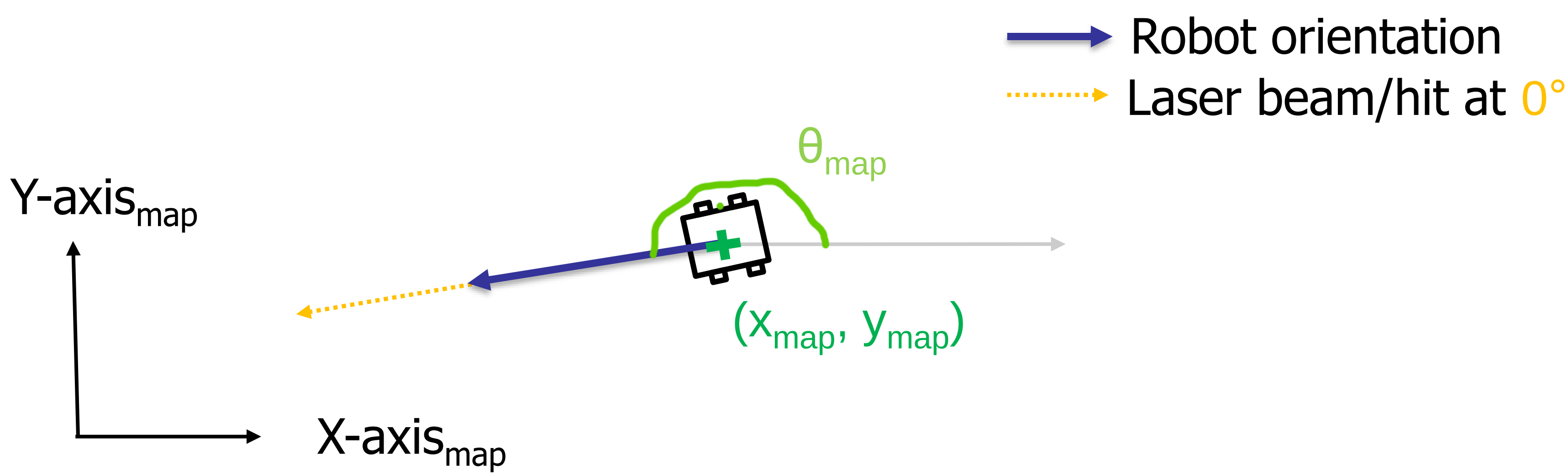
score = 0.92

score = 0.17

Requirements for localization: a sensor/observation model[Dung'07](2/5)

- Suppose that the robot is located at $(x_{map}, y_{map}, \theta_{map})$ in the frame of the map;
  1. For each hit of the laser (r_{hit}, θ_{hit}) in the polar space of the frame of the laser, we compute its position (x_{hit}, y_{hit}) in the cartesian space of the frame of the map;

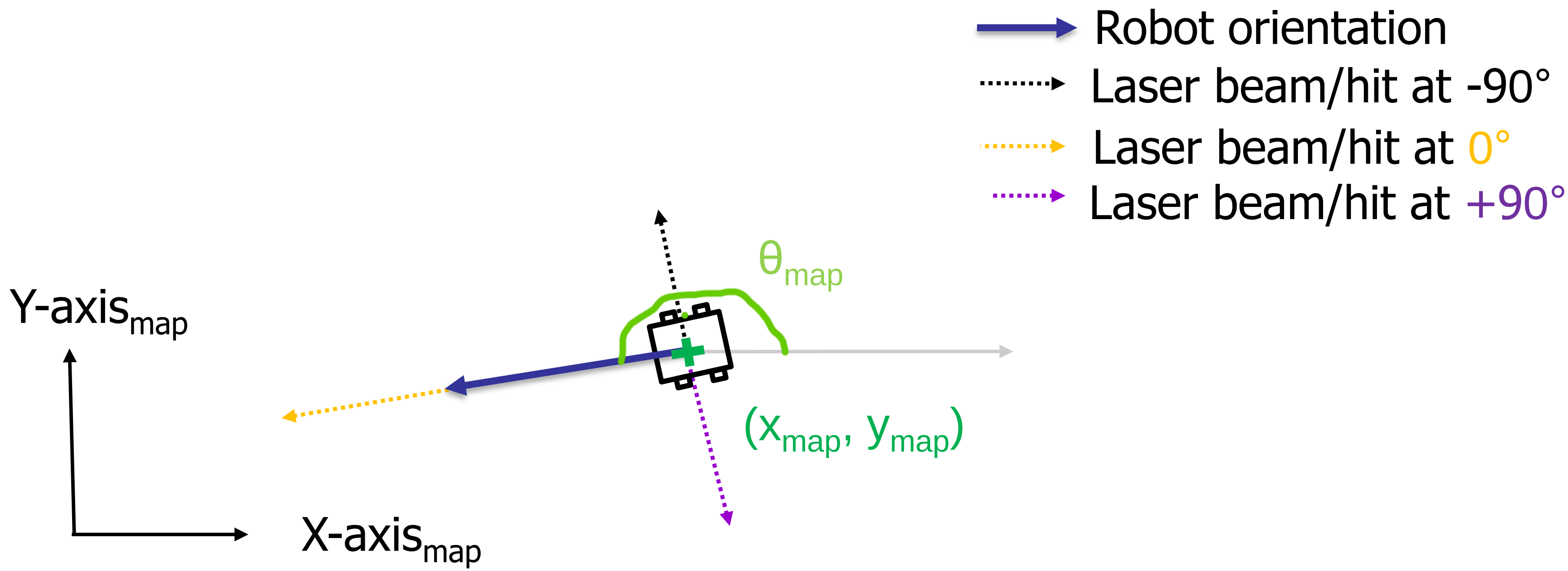
     For instance, for the hit with $(r_{hit}, \theta_{hit} = 0°)$:
     - $x_{hit} = x_{map} + r_{hit} * \cos(\theta_{map} + 0°)$
     - $y_{hit} = y_{map} + r_{hit} * \sin(\theta_{map} + 0°)$



Requirements for localization: a sensor/observation model[Dung'07](3/5)

- For instance, for the hit of the laser with (r_{hit}, θ_{hit}= -90°):
  - $x_{hit} = x_{map} + r_{hit} \cdot \cos(\theta_{map} + -90°)$
  - $y_{hit} = y_{map} + r_{hit} \cdot \sin(\theta_{map} + -90°)$

- For instance, for the hit of the laser with (r_{hit}, θ_{hit}= +90°):
  - $x_{hit} = x_{map} + r_{hit} \cdot \cos(\theta_{map} + +90°)$
  - $y_{hit} = y_{map} + r_{hit} \cdot \sin(\theta_{map} + +90°)$

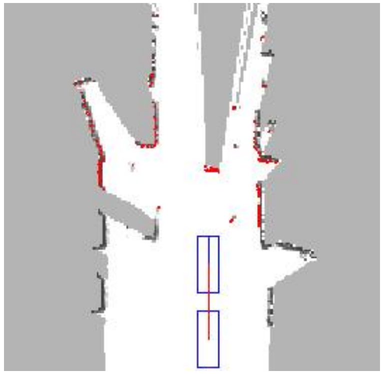


Olivier.Aycard@grenoble-inp.fr

Requirements for localization: a sensor/observation model[Dung'07](4/5)

- Suppose that the robot is located at $(x_{map}, y_{map}, \theta_{map})$ in the frame of the map;
  1. For each hit of the laser (r_{hit}, θ_{hit}) in the polar space of the frame of the laser, we compute its position (x_{hit}, x_{hit}) in the cartesian space of the frame of the map;
     - $x_{hit} = x_{map} + r_{hit} * \cos(\theta_{map} + \theta_{hit})$
     - $y_{hit} = y_{map} + r_{hit} * \sin(\theta_{map} + \theta_{hit})$
  2. If (x_{hit}, x_{hit}) matchs to an occupied cell in the map then increase score

- For each hit of a laser scan, I check if it matchs to the map or not



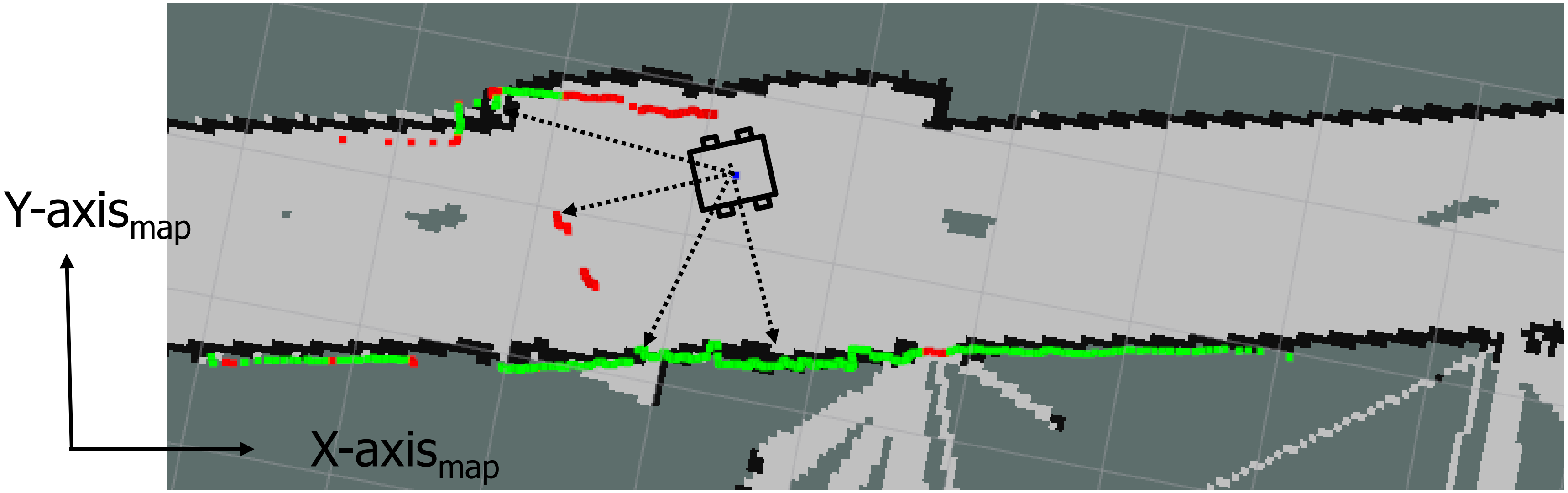
- This process is called scan matching
- It is based on a sensor model

Requirements for localization: a sensor/observation model[Dung'07](5/5)

- How to compute the scan matching score?

- For θ_{hit}= -45°, the cell is occupied, then score = 0 + 1 = 1
- For θ_{hit}= 0°, the cell is not occupied, then score = 1 + 0 = 1
- For θ_{hit}= +45°, the cell is occupied, then score = 1 + 1 = 2
- For θ_{hit}= +90°, the cell is occupied, then score = 2 + 1 = 3

- So, for these 4 hits, when the robot is at (x_{map}, y_{map}, θ_{map}), the score is 3



Outline

Olivier.Aycard@grenoble-inp.fr

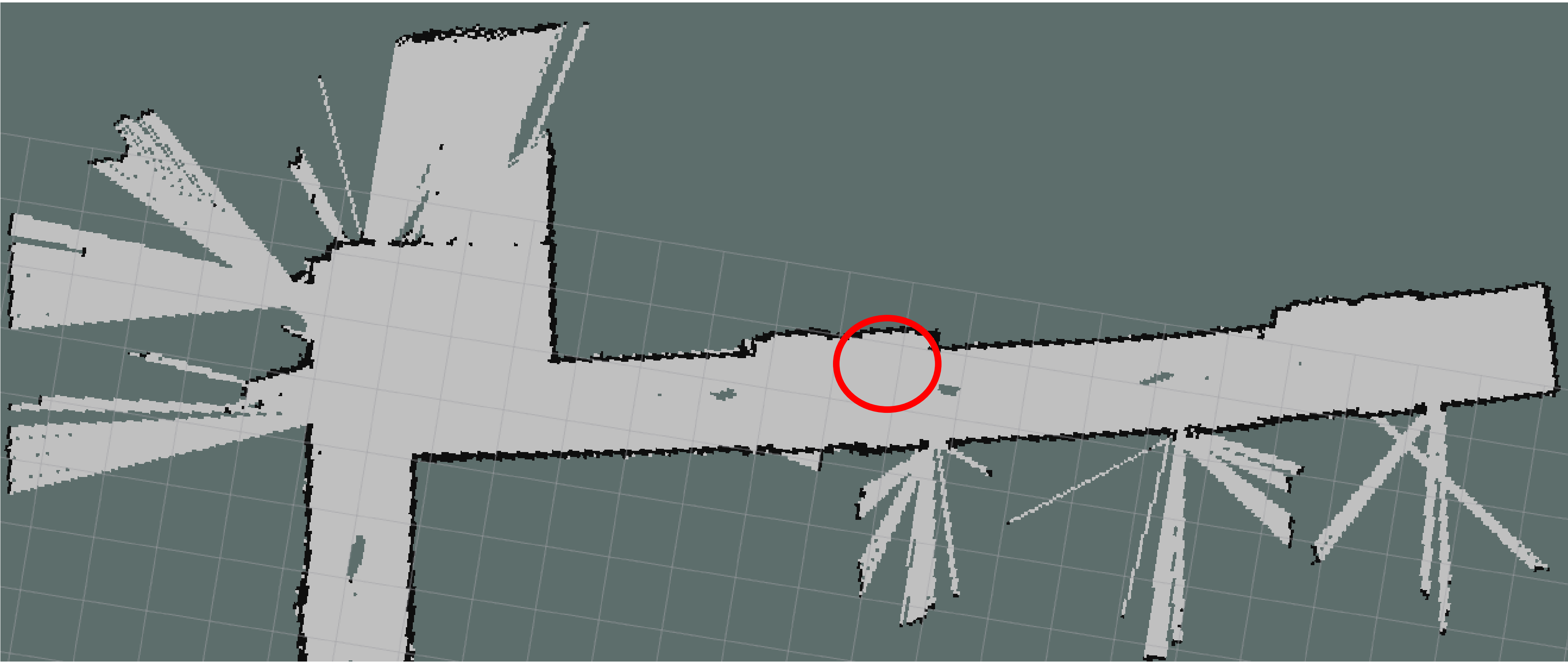
Localization algorithms: a simple one+example

- Each time, a given distance has been performed:
  1. For each possible predicted position with the motion model, compute its score with the sensor model;
  2. Keep the position with the highest score.

- Example
  - Robair is following a moving person in the LIG;
  - We have a rough idea of the initial position of Robair;
  - Every time Robair has moved by more than 1 meter, we perform localization;
  - See video on my web page.

Localization algorithms: a simple one+example (1/11)

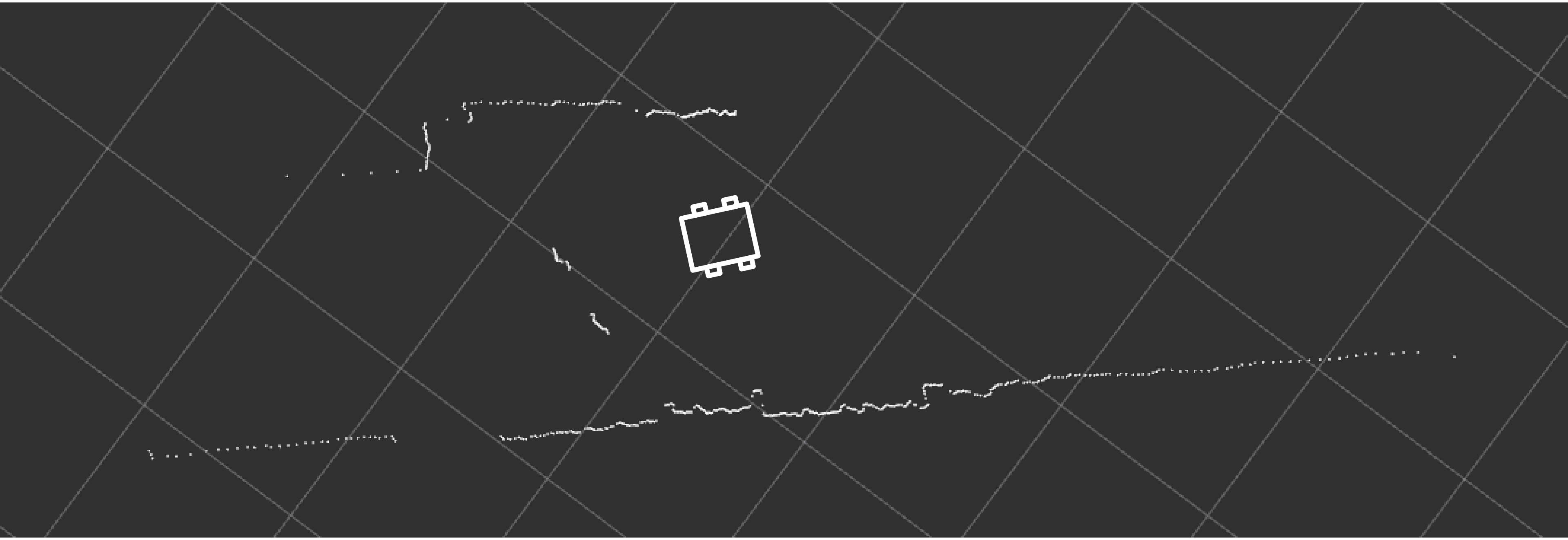
- At the beginning of the localization process, Robair is actually located somewhere here



- I will check all the positions (x, y, θ_{hit}) in the red area

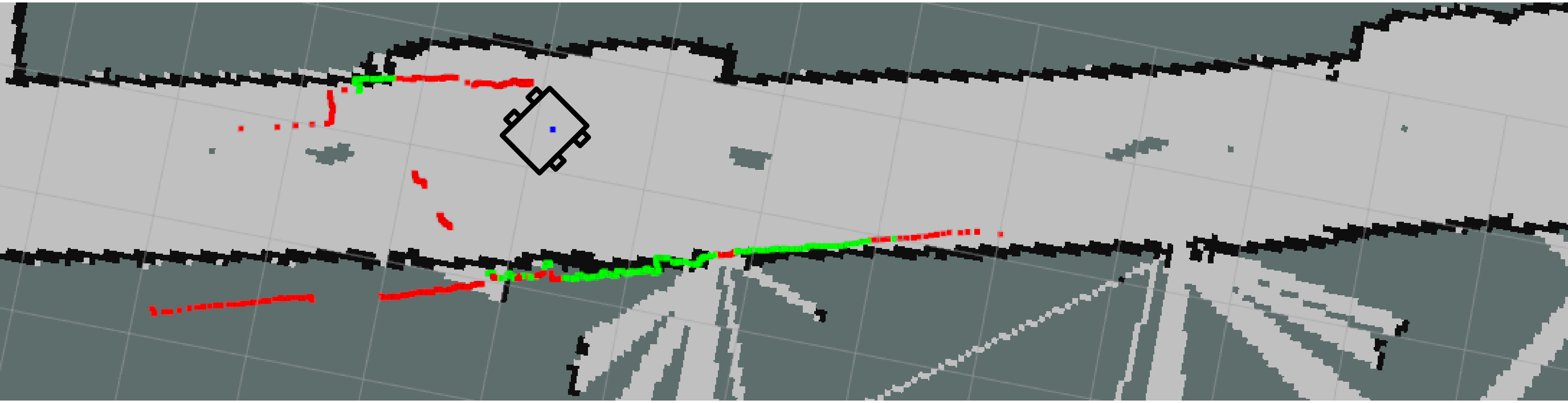
Localization algorithms: a simple one+example (2/11)

- First observation of Robair (laser scan)



Localization algorithms: a simple one+example (3/11)

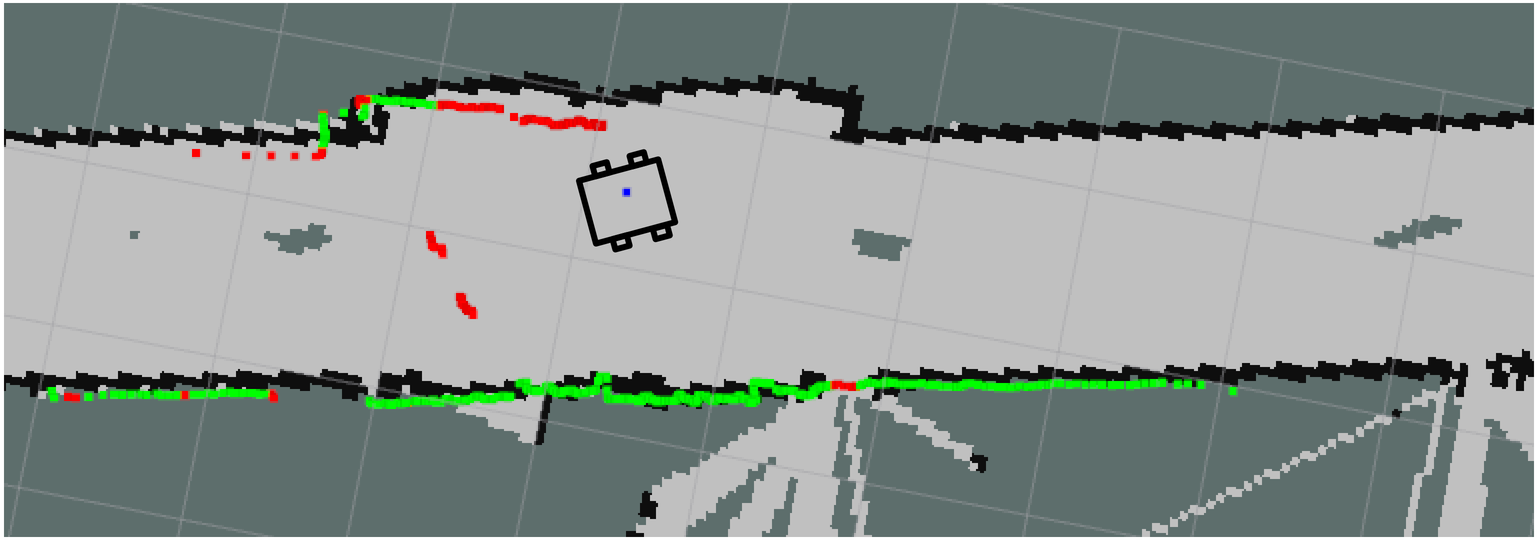
- First score with one position



- Green points are hits of the laser that match to the map
- Red points are hits of the laser that does not match to the map

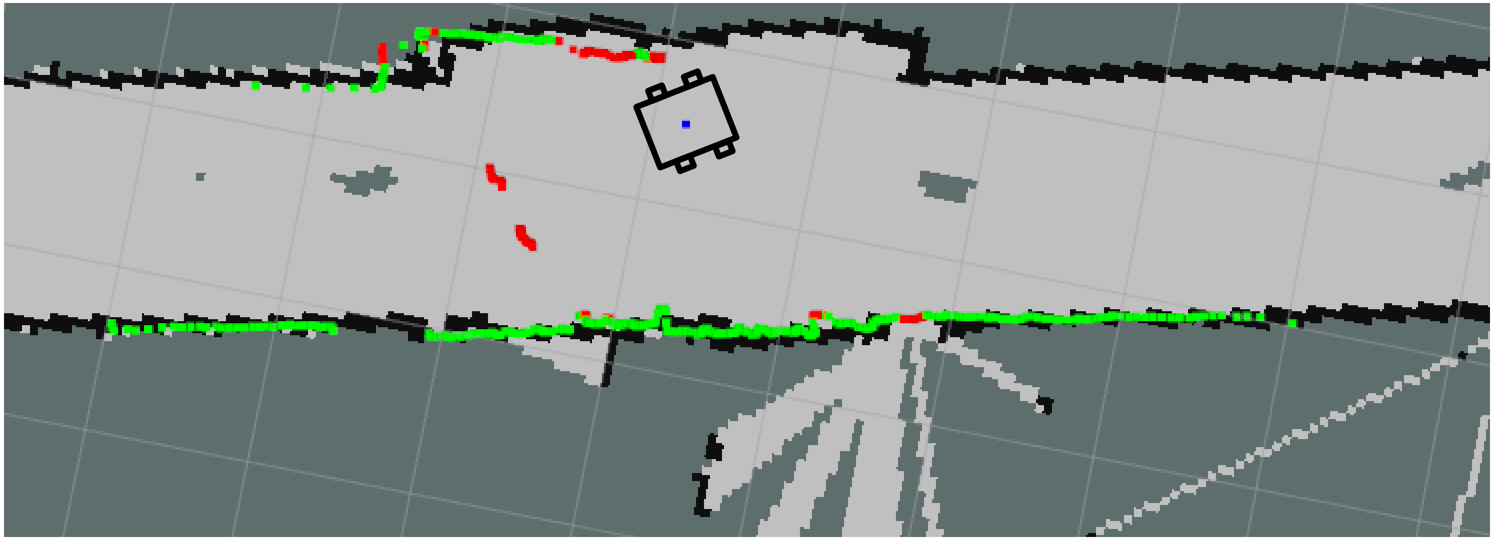
Localization algorithms: a simple one+example (4/11)

- A better score with an other position



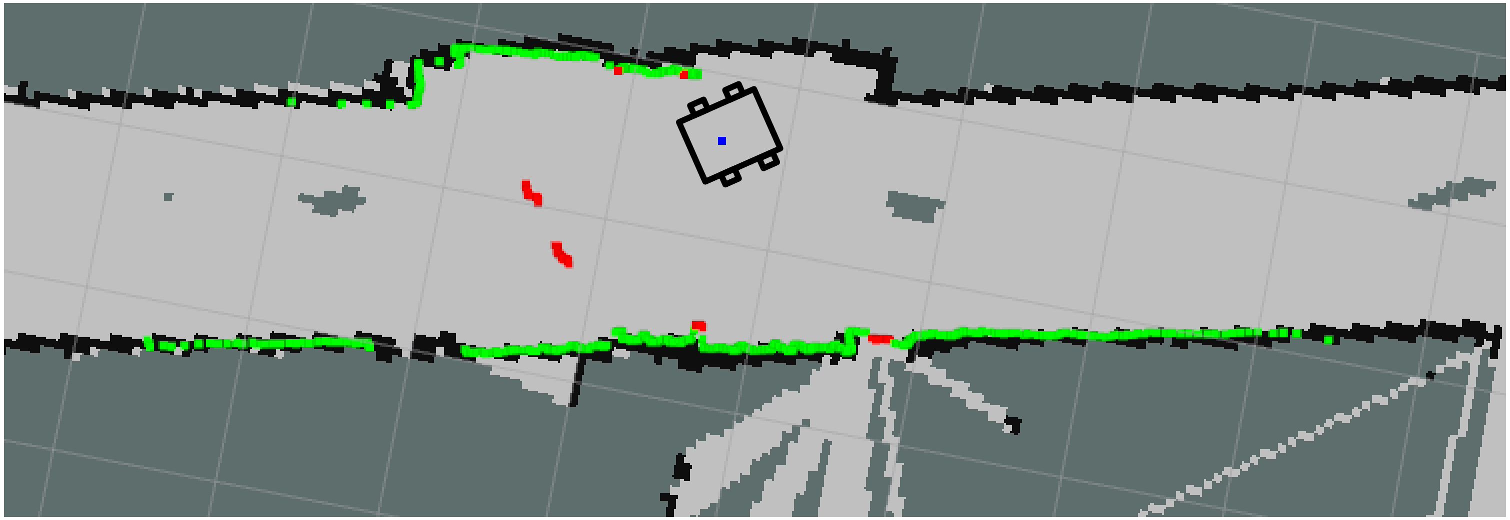
Localization algorithms: a simple one+example (5/11)

- Still better…



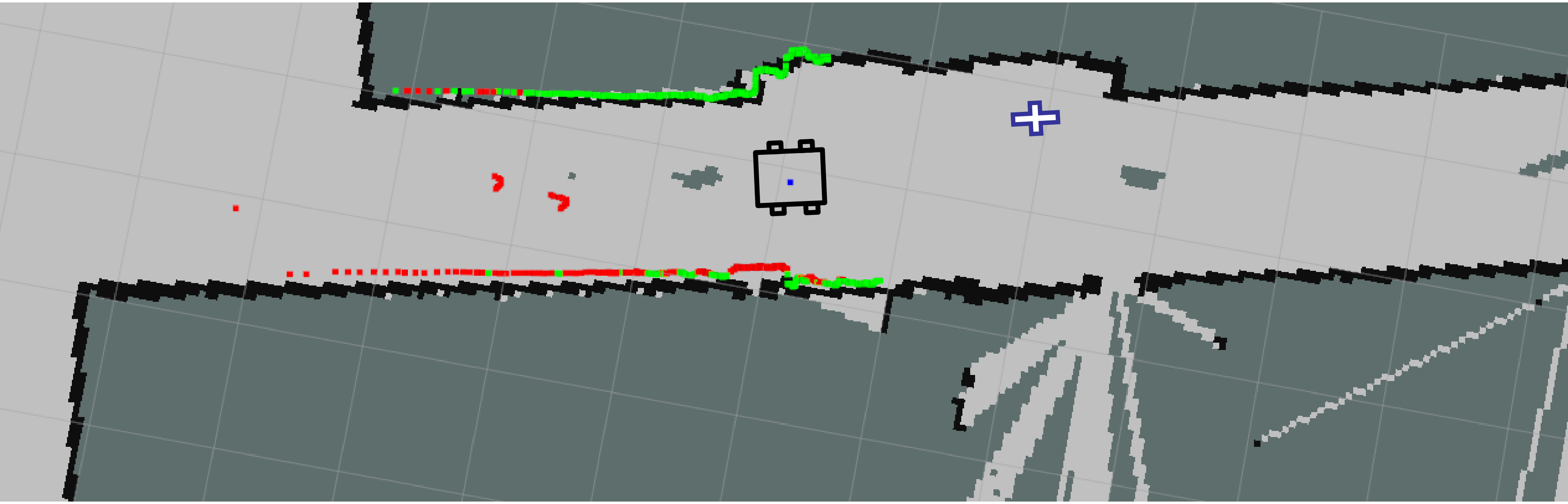
Localization algorithms: a simple one+example (6/11)

- The best score and the associated position



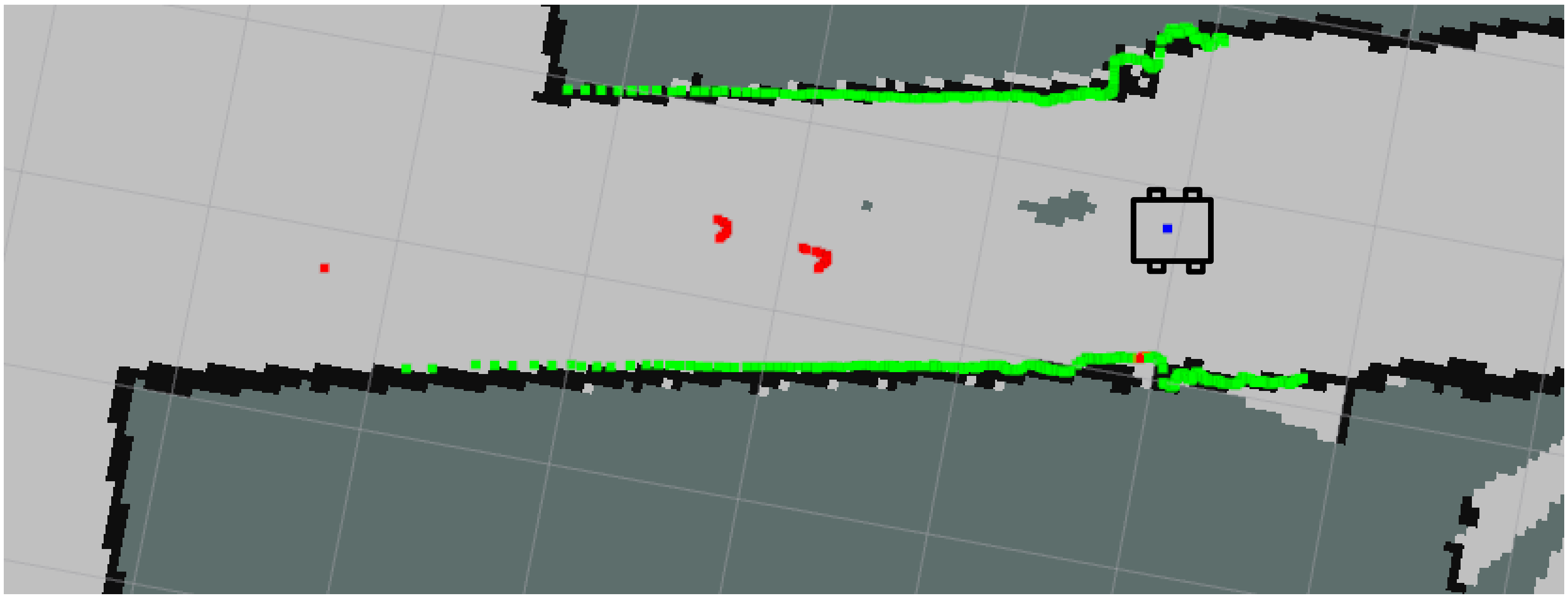
Localization algorithms: a simple one+example (7/11)

- Robair has moved of more than one meter
- We see the score of its predicted position (given the odometry)



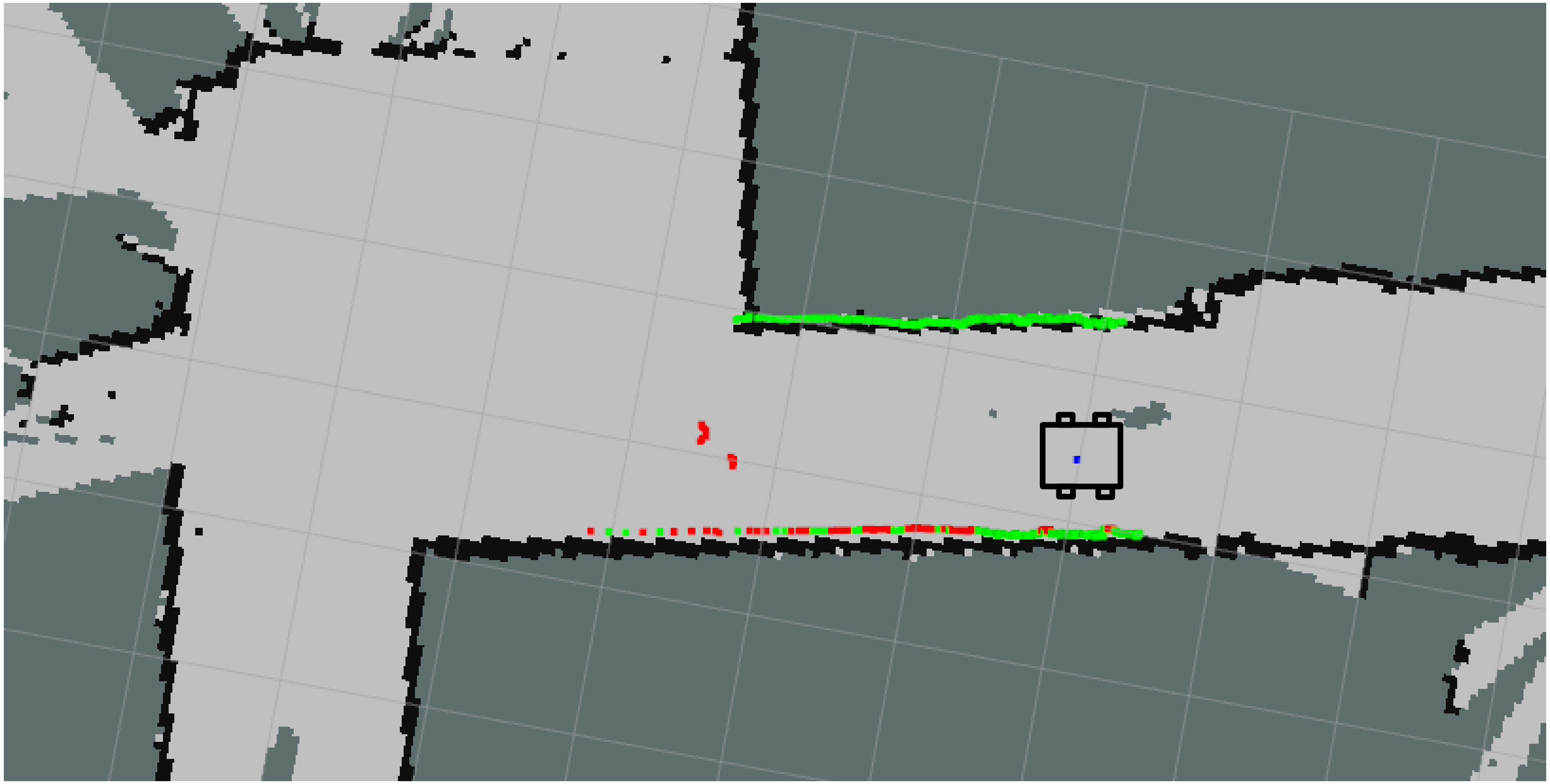
Localization algorithms: a simple one+example (8/11)

- The best score and the associated predicted position



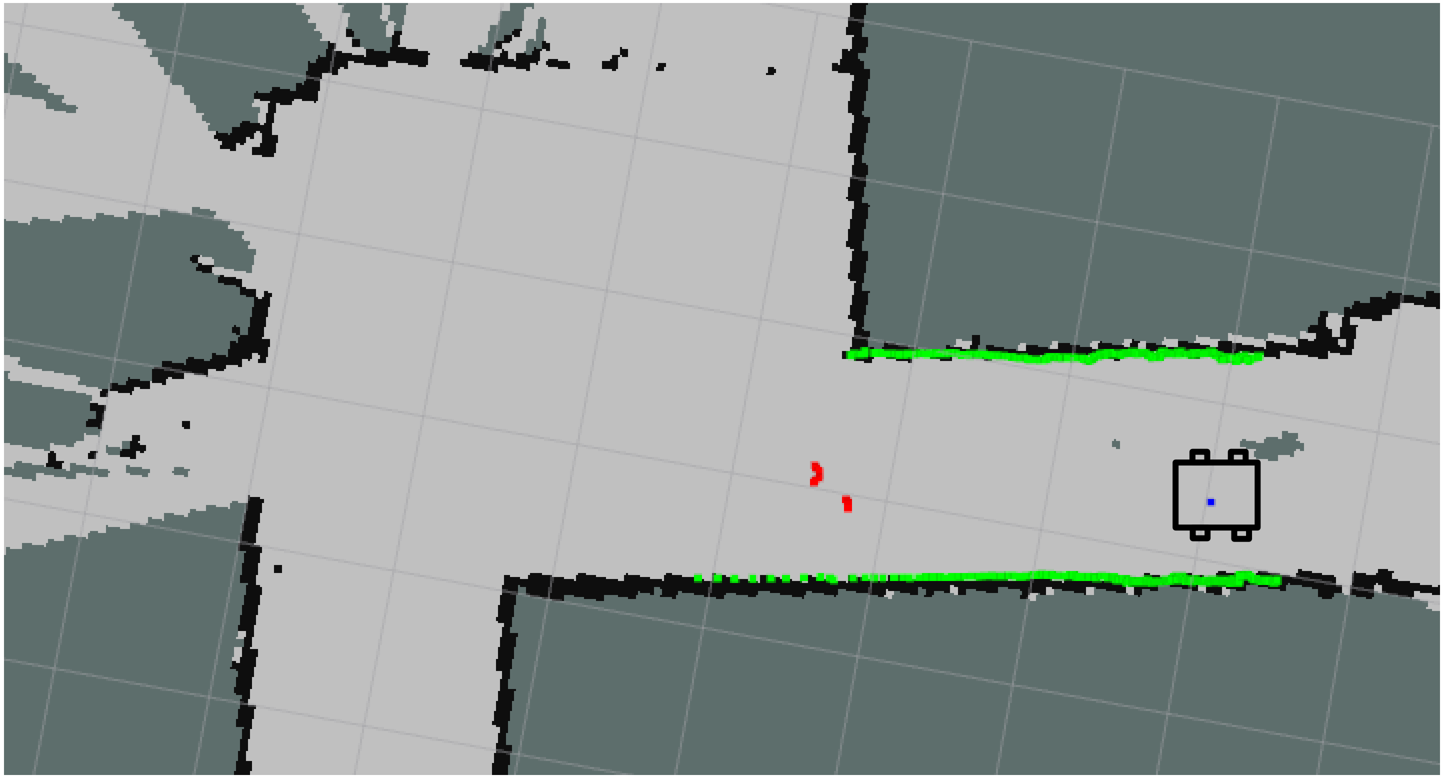
Localization algorithms: a simple one+example (9/11)

- Robair has moved again of more than one meter
- Score of its predicted position (given the odometry)



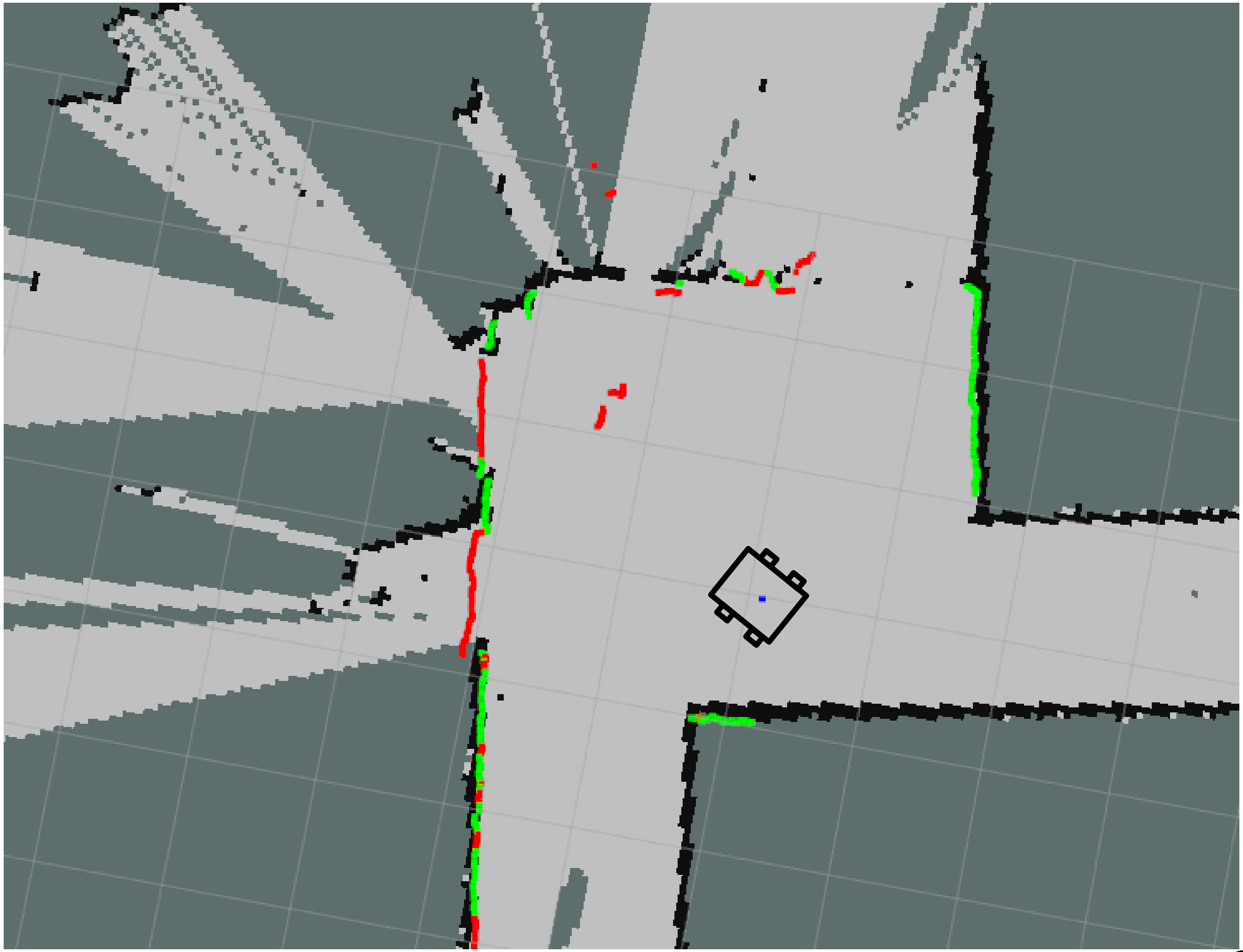
Localization algorithms: a simple one+example (10/11)

- The best score and the associated predicted position



Localization algorithms: a simple one+example (11/11)

- And so on…



Outline

1. Introduction to localization
2. Requirements for localization
3. Localization algorithms
   1. Example of execution
   2. Formalization
   3. Advanced localization

Localization algorithms: a simple one

- How to determine initial position?
  - Initial position given (by another module or manually)
  - Initial position not given: need to initialize localization:

One approach: Discretize the possible positions p (x_{map}, y_{map}, θ_{map})
Given the first laser scan, compute scores of all positions exhaustively

```
best_score = 0
For (all possible positions p) {
        score = sensor_model(p, scan)
        if (score >= best_score) {
                best_pos = p
                best_score = score
        }
}
```

Localization algorithms: a simple one

- One step of relocalization:

  The initialisation gives us the estimate of our previous position p_{prev}

  Predict new possible p 's with motion model: First, get ideal position:

  $p_{odom} = p_{prev}$ + motion_done(odometry)

  Then, account for uncertainty and odometry drift by considering all positions p close to the ideal one:

  $(p_{odom} - constant) < p < (p_{odom} + constant)$

  Then, we compute scan matching scores like in initialization;

  Best possible position becomes our p_{prev} for the next relocalization;

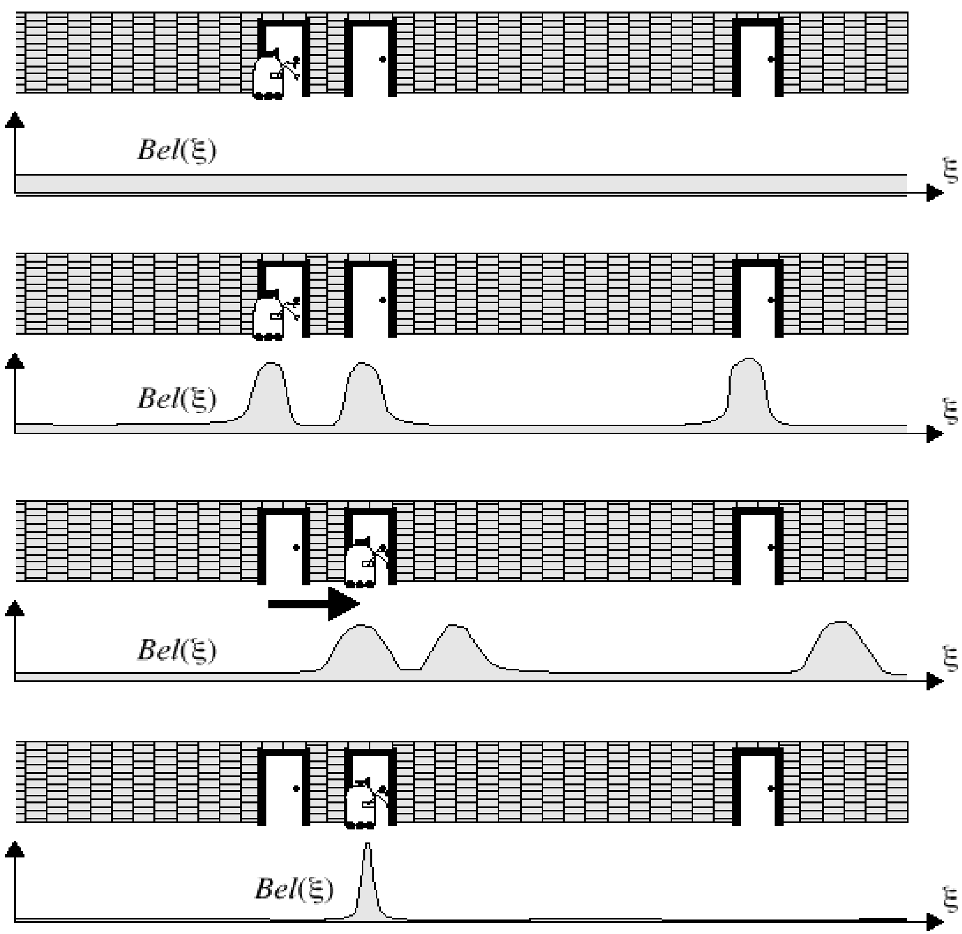
  We relocalize when odometry tells us we have moved enough;

```
motion_done = new_odom_data - prev_odom_data
if (motion_done > 1meter) { relocalize( );}
```

Outline

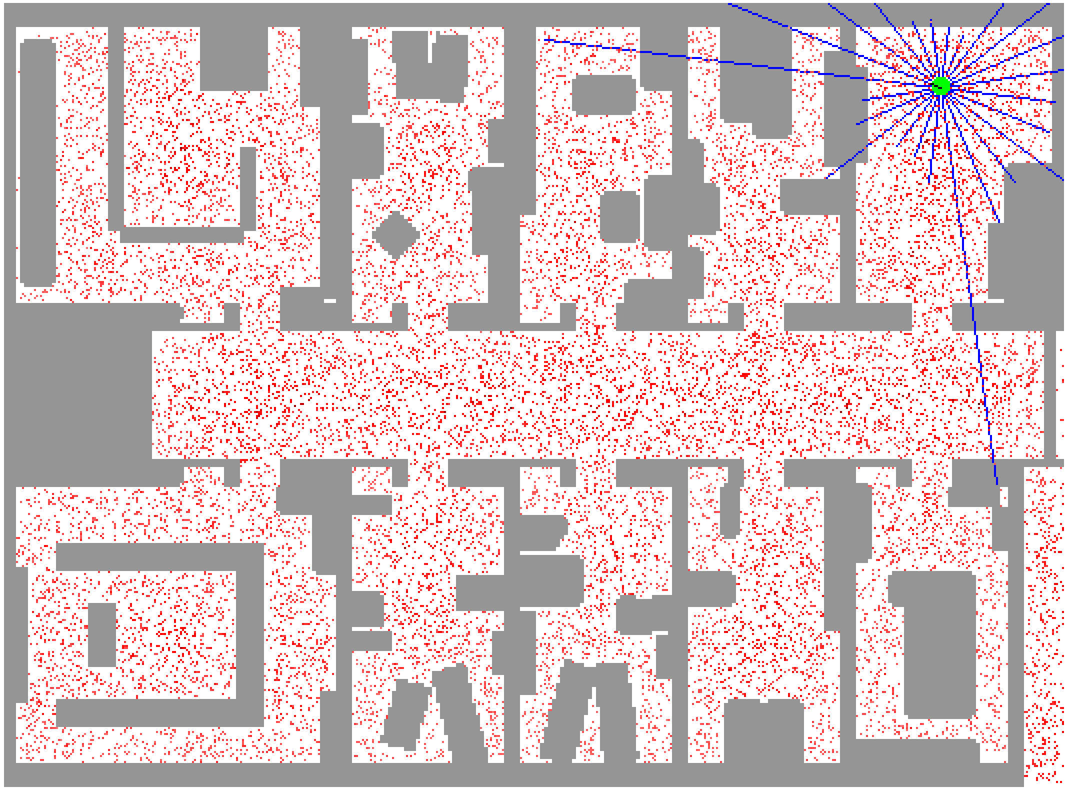
1. Introduction to localization
2. Requirements for localization
3. Localization algorithms
   1. Example of execution
   2. Formalization
   3. Advanced localization

The localization problem[Fox'98]

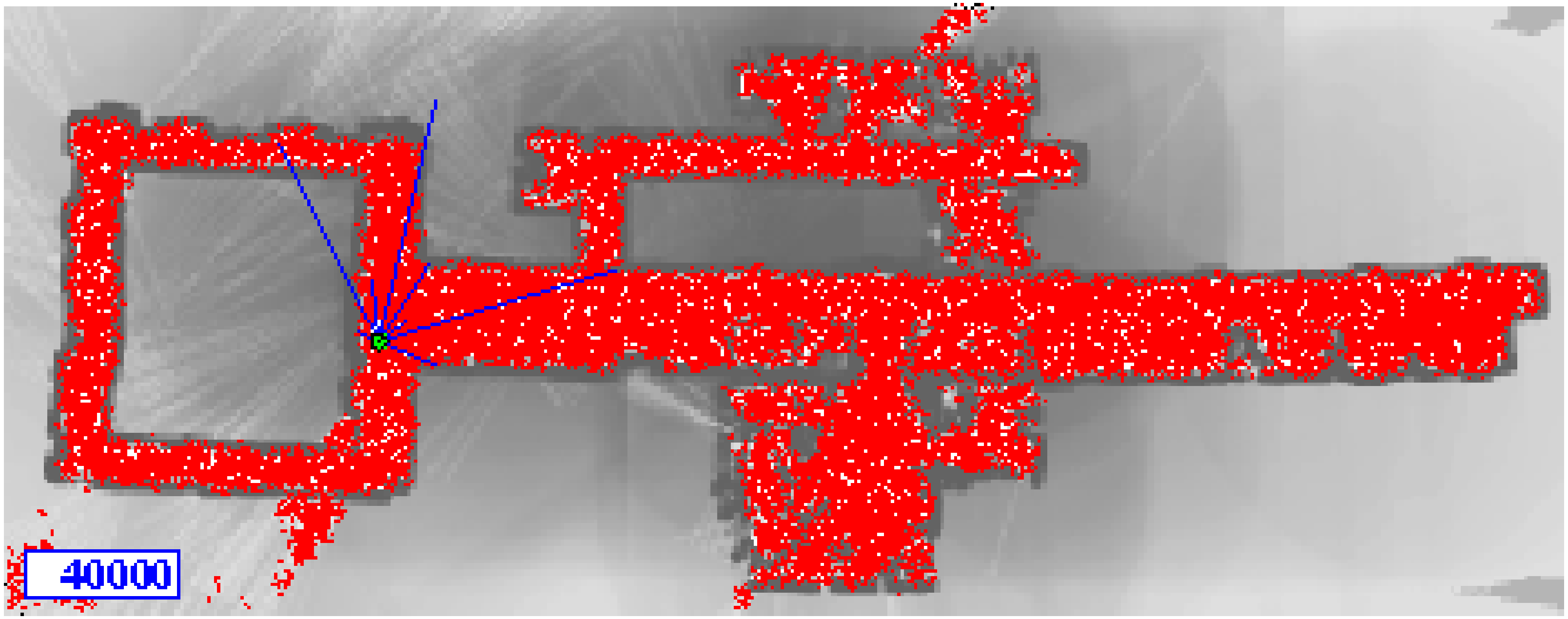


- If the initial position is unknown:
  - Robair observes its environment to have an idea of where it is
  - It will manage several positions /hypotheses
- Managing several hypotheses/positions improves the robustness of localization

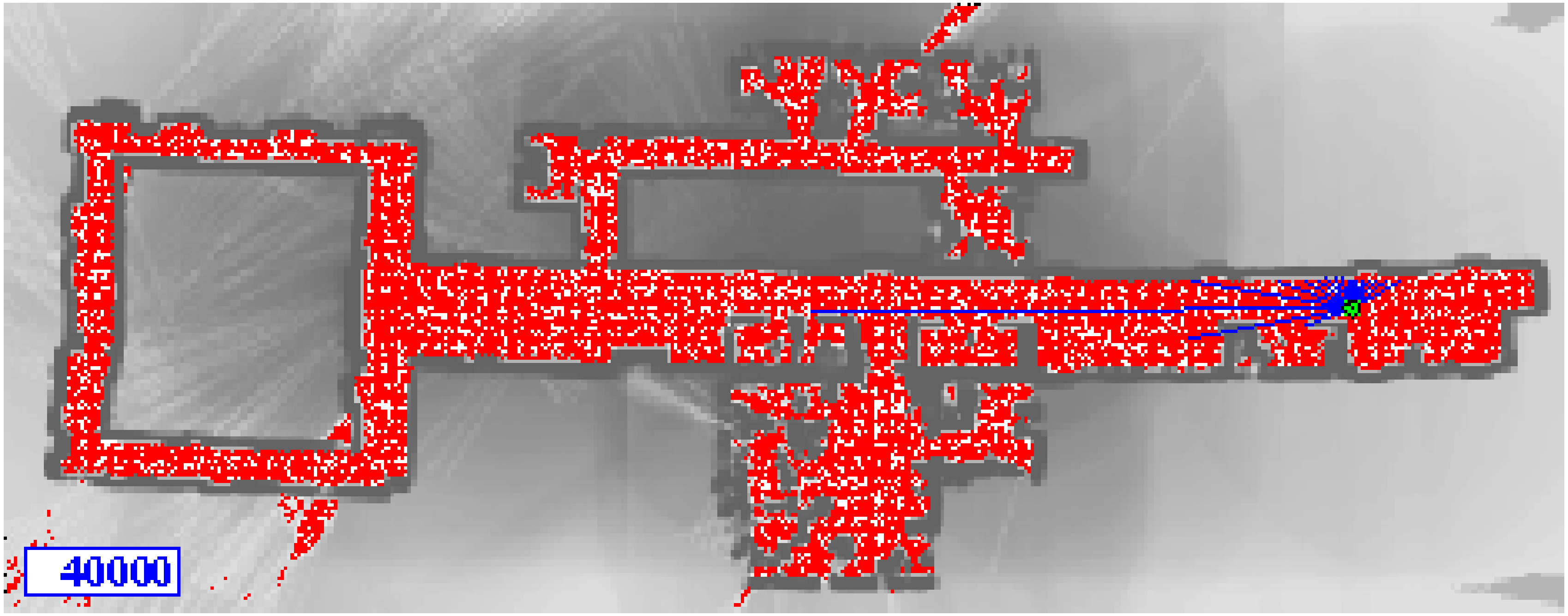
Initial position unknown: ultrasonic sensor[Fox'98]



Initial position unknown: ultrasonic sensor[Fox'98]



Initial position unknown: laser sensor[Fox'98]



Summary(1/2)

- The localization process is done in 2 steps:
  1. Prediction of possible positions after a given motion;
  2. Find the predicted position that corresponds the best to the observation
- 2 possible initial positions: known or unknown;
- The localization process is a passive process;
- The localization process is used in all navigation systems;

Summary(2/2)

- Public nodes for mapping and localization are available in ROS: gmap and amcl;
- A complete formalization of the localization process can be found on my web page;
- A video illustrating the localization process can be found on my web page.