Introduction to Mobile Robotics

Lecture 8: Exploration

Zhi Yan

ENSTA, Institut Polytechnique de Paris

What is Exploration?

Definition:

Exploration is the task by which a robot autonomously moves through an **unknown** environment for the purpose of building a map.

It's the logical step *after* SLAM:

- **Passive SLAM**: The robot builds a map while being guided (teleoperation or a pre-defined plan).
  $\Longrightarrow$ It answers: "Where am I?"
- **Active exploration**: The robot *autonomously decides* where to go next to gain new information.
  $\Longrightarrow$ It answers: "Where should I go next?"

Objective:

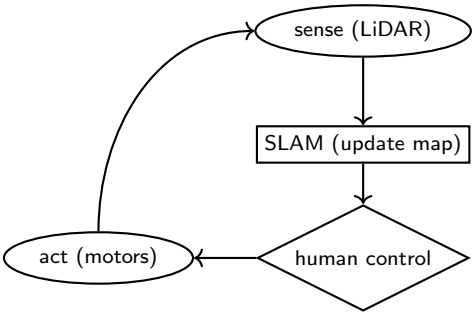
To gain the maximum amount of information about the environment, as efficiently as possible, to build a complete map.

ENSTA IP PARIS

The "Active SLAM" Problem

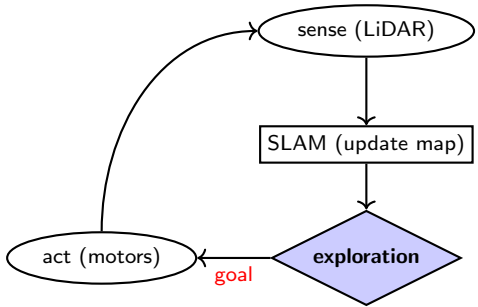
Exploration is not separate from SLAM, it is the "decision" layer on top of it.

Passive SLAM loop



The "goal" comes from an external operator.

Active exploration loop



The robot generates its own goals.

ENSTA IP PARIS

The Fundamental Problem: "Next-Best-View"

Given what I know (the current map), where should I move to see the most "new" information?

This involves a fundamental trade-off:

Greedy exploitation

- Go to the *closest* unknown area.
- Fast, safe information gain.
- **Risk**: Can be inefficient. May get "stuck" in complex local areas.

Pure exploration

- Go to a *distant* unknown area that might be very large.
- High potential reward, but high cost (e.g., travel time).
- **Risk**: The long journey may yield little information.

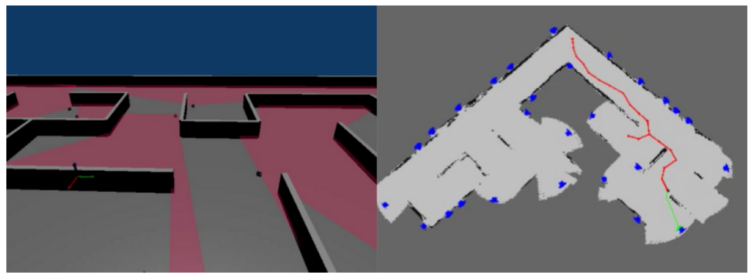
A good algorithm must balance the cost of travel with the potential information gain.

ENSTA IP PARIS

Map Representation for Exploration

To explore, the map needs three states. The Occupancy Grid (from Lecture 7) is perfect for this:

- **Occupied (black)**: $P(m_i) > 0.65$ (e.g., a wall)
- **Free (white)**: $P(m_i) < 0.25$ (e.g., open space)
- **Unknown (gray)**: $0.25 \leq P(m_i) \leq 0.65$ (e.g., unseen area)



The goal of exploration:

To convert all **gray** (unknown) cells into either **black** (occupied) or **white** (free).

ENSTA IP PARIS

Frontier-based Exploration (FBE)

The most common and foundational exploration algorithm.

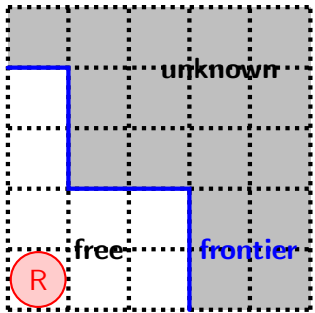
(Yamauchi, 1997)

ENSTA
IP PARIS

The Core Concept: Frontiers

Definition:

A **frontier** is the boundary (the edge) between *known free space* and *unknown space*.



Logic:

Move to the frontier in order to gain new information (from the unknown).

ENSTA IP PARIS

The FBE Algorithm

The algorithm is a continuous loop:

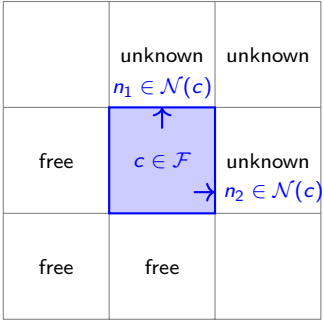
1. **Perceive (sense)**: Get the current map from the SLAM node.
2. **Detect frontiers**: Find all boundary cells between “free” and “unknown”.
3. **Cluster**: Group adjacent frontier cells into larger segments (clusters).
4. **Select goal**: Choose the “best” frontier cluster to move to.
5. **Navigate**: Use a path planner (e.g., Nav2) to go to the goal.
6. **Repeat**: Go back to Step 1 until there are no more frontiers to explore.

ENSTA IP PARIS

Step 2: Frontier Detection

Algorithm: Iterate through every cell in the map, and a cell c is a frontier cell if:

1. c is **known and free**, and
2. c has at least one neighbor cell n that is **unknown**.



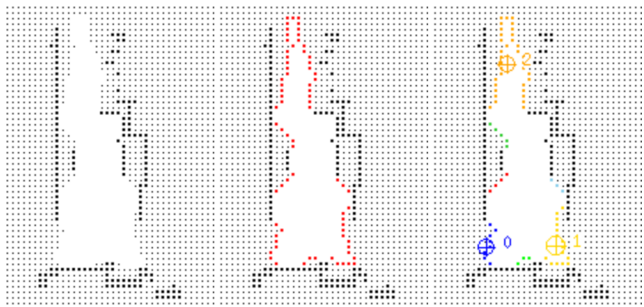
Formulation: Let $\mathcal{M}$ be the grid map and $\mathcal{N}(c)$ be the neighborhood of a cell c. The set of frontier cells $\mathcal{F}$ is defined as:

$$\mathcal{F} = \{c \in \mathcal{M} \mid S(c) = \text{free} \wedge \exists n \in \mathcal{N}(c) : S(n) = \text{unknown}\}$$

ENSTA IP PARIS

Step 3: Frontier Clustering

- Frontier detection may provide many cells.
- Which cell should the robot choose as the goal?
- We can use a simple algorithm like Breadth-First Search (BFS) to find connected frontier cells, the **cluster**.
- We then find the centroid (center point) of each cluster to use as a potential goal.



ENSTA
IP PARIS

Step 4: Goal Selection

How do we define "best"?

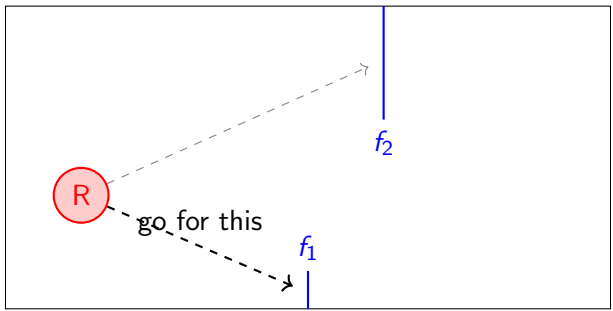
- **Utility Function:** A way to score each frontier cluster.

- Three common heuristics for the utility function:
  - Proximity (cf. *greedy exploitation*)
  - Size (information gain) (cf. *pure exploration*)
  - Proximity + Size (a *trade-off*)

ENSTA IP PARIS

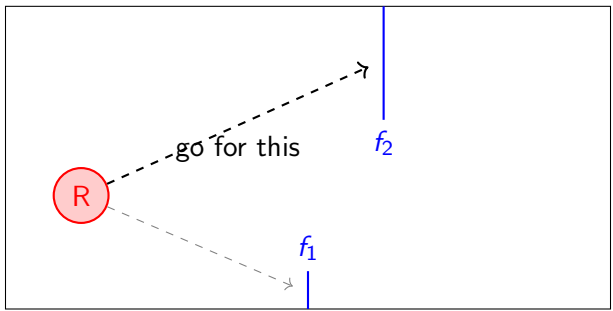
Heuristic 1: Proximity

- **Logic:** Choose the *closest* frontier.
- **Utility Function:** $Utility(f) = -Cost(robot, f)$
  - *Cost* can be the path length from the robot to the frontier's centroid, calculated by a planner like A*.
- **Pros:** Simple, fast, explores locally first.
- **Cons:** Can be inefficient (e.g., close frontier for little information)



Heuristic 2: Size

- **Logic:** Choose the *largest* frontier.
- **Utility Function:** $Utility(f) = Size(f)$
  - *Size* is the number of frontier cells in the cluster.
- **Pros:** Tries to find large open areas.
- **Cons:** Can be inefficient (e.g., long way for little information).



Heuristic 3: The Combined Approach

Logic: A weighted sum of cost and information gain.

$$Utility(f) = \alpha \cdot Size(f) - \beta \cdot Cost(robot, f)$$

where α and β are tuning weights to prioritize gain vs. cost.

$\Longrightarrow$ This balances the desire for new information with the cost of getting it.

ENSTA IP PARIS

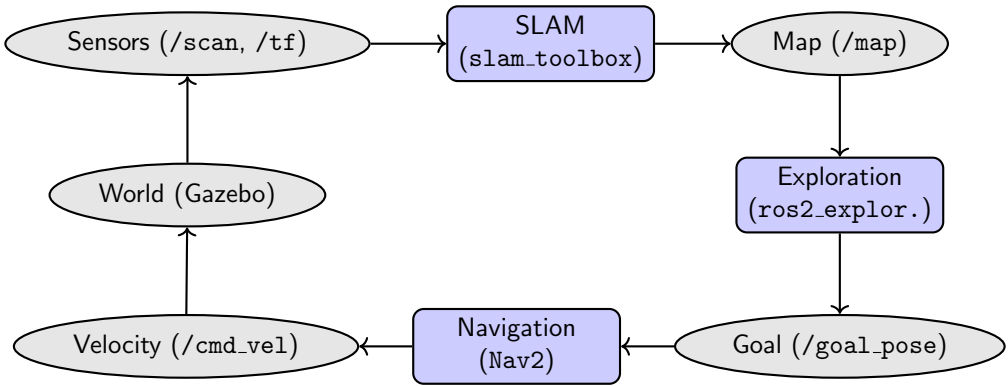
Step 5: Navigation

How to safely reach the goal?

- Details covered in Lectures 9 and 10.
- But here's a glimpse with **Nav2**:
  - Receives the goal.
  - Plans a path to f_{goal}.
  - Executes the path, avoiding obstacles.
  - Reports "Success" or "Failure".

ENSTA IP PARIS

The Full Autonomous Stack



Maintaining the Frontier List

Facts:

- When a robot moves, its sensors (especially LiDAR) “clear” a large arc of unknown space.
- Many frontiers can be cleared at once, not just the goal frontier.

What should be done:

- A good FBE implementation constantly re-evaluates the frontier list.
- If a frontier cluster is no longer adjacent to “unknown” space, it is removed from the list of goals.

ENSTA IP PARIS

Challenges in FBE

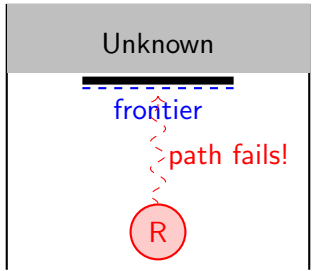
Why exploration can be hard in practice?

ENSTA
IP PARIS

Challenge: Inaccessible Frontiers

A frontier may be detected, but impossible to reach.

- **Example 1**: A frontier seen through a window $\Longrightarrow$ The map shows unknown space, but the planner can't find a path.
- **Example 2**: A frontier under a table $\Longrightarrow$ The frontier is detected, but the robot's C-space (inflation radius) is too large to fit.



Solution: The navigation stack will report "Failure". The exploration node must then blacklist this frontier and select the next-best-one.

ENSTA IP PARIS

Challenge: Greedy Failures

1. Robot is in the hallway. Sees frontiers for Office 1 and Office 2.
2. Chooses Office 1 (it's closer).
3. Enters and explores all of Office 1.
4. Now, the closest frontier is... back in the hallway, at the entrance to Office 2.
5. Robot drives all the way out of Office 1 and down the hall to Office 2.

Inefficient! A better algorithm would have explored the hallway *first*, then each office.

ENSTA IP PARIS

Challenge: Oscillations

What happens if the robot is in a wide hallway, and sees two equally "good" frontiers on the left and right?

1. Selects frontier f_{left}.
2. Moves 1 meter towards it.
3. Re-evaluates. Now f_{right} is slightly closer!
4. Selects frontier f_{right}.
5. Moves 1 meter towards it.
6. Re-evaluates. Now f_{left} is slightly closer!

$\Longrightarrow$ The robot gets stuck oscillating in the middle of the hallway.

Solution: Add *hysteresis* (e.g., a "commitment" timer) : Once a goal is chosen, the robot must pursue it for at least N seconds before re-evaluating.

ENSTA IP PARIS

Post-lecture Reading: Beyond FBE

Information Theory & Multi-Robot Systems

ENSTA
IP PARIS

Information-Theoretic Exploration

Idea: Don't just go to the "edge". Go to the spot that will **maximize the reduction in map uncertainty (entropy)**.

- We treat the map not as three states, but as a full probability distribution.
- Entropy is a measure of uncertainty: The gray area has high entropy, while the black and white areas have zero entropy.
- For several candidate "next poses" (views):
  1. Simulate the scan the robot would get from that pose.
  2. Calculate how much that scan would reduce the total map entropy.
  3. Go to the pose with the highest "Mutual Information" (entropy reduction).

Pros: More efficient, globally aware.
Cons: High computational cost.

ENSTA IP PARIS

Frontier-based vs. Information Gain

Frontier-based (greedy algorithm)

- Simple, fast, robust.
- Asks: "Where is the edge of the unknown?"
- Can be inefficient.
- Very widely used.

Information-Theoretic (global optimization)

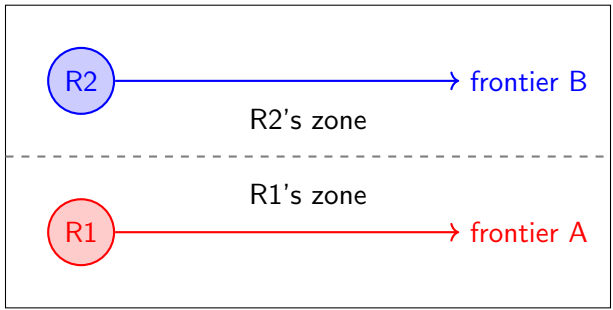
- Complex, slow (computational).
- Asks: "What action gives me the most information?"
- More efficient in travel time.
- Used in research.

ENSTA
IP PARIS

Multi-robot Exploration

Problem: How to coordinate multiple robots to explore an area faster?

- **Challenge 1: Map Merging** $\Longrightarrow$ How do robots combine their individual maps?
- **Challenge 2: Task Allocation** $\Longrightarrow$ How to divide the work?



ENSTA IP PARIS

The Task Allocation Problem

- Frontiers are "tasks" that need to be done.
- Robots "bid" on tasks.
- A bid is based on the robot's cost to complete the task (e.g., distance, energy consumption, etc.).
- The robot with the lowest bid "wins" the task (the frontier) and navigates to it.
- Other robots ignore that frontier and bid on the remaining ones.

$\Longrightarrow$ A core topic in multi-robot systems (Lecture 11).

ENSTA IP PARIS

Summary

- Exploration is the autonomous **decision-making** layer on top of SLAM.
- **FBE** is the foundational concept.
- Goal selection is a **utility function** balancing the "cost" of travel against the potential "information gain".
- This is a key component of a **fully autonomous** system, acting as a "client" that sends goals to the "server" (the Navigation stack).

ENSTA
IP PARIS

Questions?

Next: Practical Work 8 - Frontier-based Exploration

ENSTA
IP PARIS