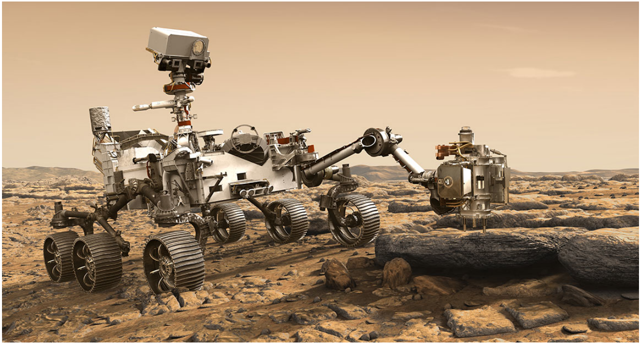
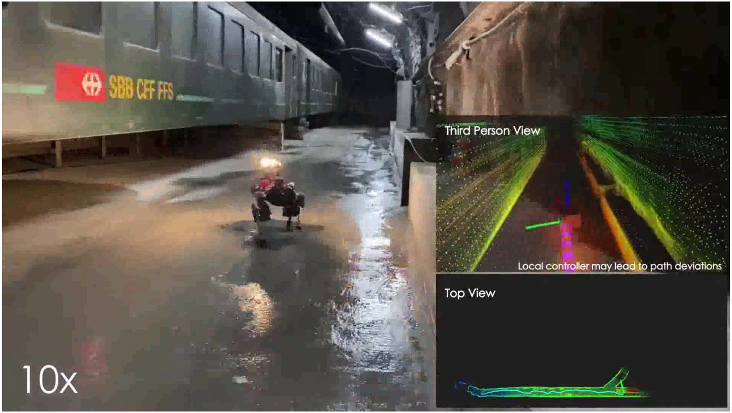
Robotic exploration

Vojtěch Vonásek

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague



Motivation scenarios



https://www.youtube.com/watch?v=P3uT4gHEFHw

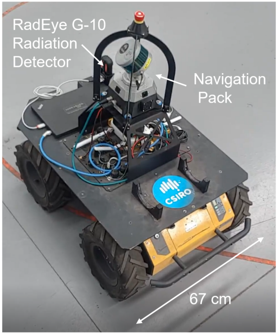
Finding precious metals, water sources, etc.

Motivation scenarios

Finding and rescuing people in debris

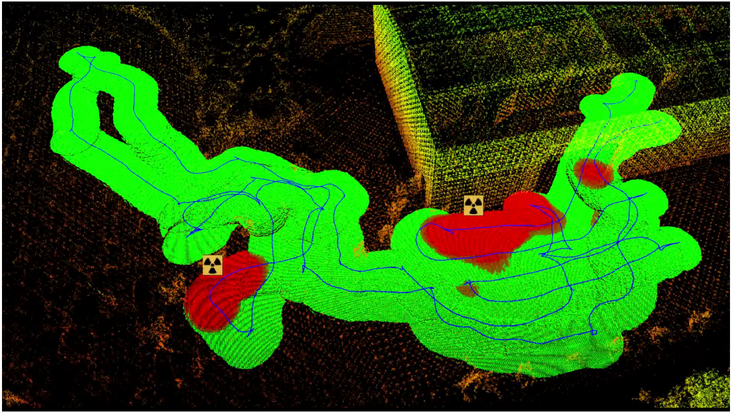
Source: robohub.org

Motivation scenarios



Mapping radioactive zones

☛Groves, K.; Hernandez, E.; West, A.; Wright, T.; Lennox, B. Robotic Exploration of an Unknown Nuclear Environment Using Radiation Informed Autonomous Navigation. Robotics 2021, 10, 78.



www.youtube.com/watch?v=Hj7xt7isOWc

Mapping radioactive zones

☛Groves, K.; Hernandez, E.; West, A.; Wright, T.; Lennox, B. Robotic Exploration of an Unknown Nuclear Environment Using Radiation Informed Autonomous Navigation. Robotics 2021, 10, 78.

Terminology

Exploration

- *the activity of searching and finding out about something* (Cambridge English dictionary)
- *. . . is a trip, but it's more than just a vacation — it's going somewhere to examine and discover new things* (vocabulary.com)

Robotic exploration

- *use a robot to maximize knowledge over a particular area*
    - Is there a precious metal? Where are the victims? Is there a radioactivity?
- Fundamental problem of robotics
- Single robot vs. multi-robot exploration
- Practical problem needed in many applications

How to approach exploration

Reactive

- measure, evaluate, act, measure, evaluate, . . .
- not optimal (e.g., time/energy consuming)
- can lead to cycles

Decision-based

- build a model of the environment
- make decision using the model
- more efficient, can be optimized
- extra effort to make the model of the environment

How to represent/model environment?

- Many approaches
- Model should always be selected according given application

Example of models of environments I

Low-level data

- Raw sensor data (e.g. LIDAR values, images)
- Neural network models (weights+topologies), learned policies (Reinforcement learning), rosbag
- File formats: (txt,bin,SQL,HDF,. . .)
- They are specific to given task/environment
- Hard to interpret by humans
- Poor generalization


```
-6.90069 -3.4217 5.68929 -0.337081 -0.0260944 0 -6.85441 -3.44054 5.68831 -0.436222 -0.0131647 0 2 2654
-6.85441 -3.44054 5.68831 -0.436222 -0.0131647 0 -6.81021 -3.46388 5.68797 -0.53538 -0.000235049 0 2 2654
-6.81021 -3.46388 5.68797 -0.53538 -0.000235049 0 -6.76854 -3.49147 5.68828 -0.634538 0.0126946 0 2 2654
-6.76854 -3.49147 5.68828 -0.634538 0.0126946 0 -6.72982 -3.52305 5.68924 -0.73368 0.0256243 0 2 2654
-6.72982 -3.52305 5.68924 -0.73368 0.0256243 0 -6.69381 -3.5577 5.69085 -0.77459 0.038554 0 2 2654
-6.69381 -3.5577 5.69085 -0.77459 0.038554 0 -6.65811 -3.59264 5.6931 -0.77459 0.0514837 0 2 2654
-6.65811 -3.59264 5.6931 -0.77459 0.0514837 0 -6.62244 -3.62755 5.69599 -0.77459 0.0644134 0 2 2654
-6.62244 -3.62755 5.69599 -0.77459 0.0644134 0 -6.58679 -3.66243 5.69953 -0.77459 0.077343 0 2 2654
-6.58679 -3.66243 5.69953 -0.77459 0.077343 0 -6.55118 -3.69728 5.70372 -0.77459 0.0902727 0 2 2654
-6.55118 -3.69728 5.70372 -0.77459 0.0902727 0 -6.51561 -3.73209 5.70855 -0.77459 0.103202 0 2 2654
-6.51561 -3.73209 5.70855 -0.77459 0.103202 0 -6.48009 -3.76685 5.71402 -0.77459 0.116132 0 2 2654
-6.48009 -3.76685 5.71402 -0.77459 0.116132 0 -6.44463 -3.80156 5.72013 -0.77459 0.129062 0 2 2654
-6.44463 -3.80156 5.72013 -0.77459 0.129062 0 -6.40922 -3.83621 5.72689 -0.77459 0.141991 0 2 2654
-6.40922 -3.83621 5.72689 -0.77459 0.141991 0 -6.37388 -3.87079 5.73429 -0.77459 0.154921 0 2 2654
-6.37388 -3.87079 5.73429 -0.77459 0.154921 0 -6.33861 -3.90531 5.74232 -0.77292 0.167851 0 2 2654
-6.33861 -3.90531 5.74232 -0.77292 0.167851 0 -6.30173 -3.93792 5.75099 -0.675262 0.18078 0 2 2654
-6.30173 -3.93792 5.75099 -0.675262 0.18078 0 -6.26195 -3.96671 5.7603 -0.577836 0.19371 0 2 2654
-6.26195 -3.96671 5.7603 -0.577836 0.19371 0 -6.21967 -3.99144 5.77024 -0.480656 0.20664 0 2 2654
-6.21967 -3.99144 5.77024 -0.480656 0.20664 0 -6.17532 -4.0119 5.78082 -0.383739 0.21957 0 2 2654
-6.17532 -4.0119 5.78082 -0.383739 0.21957 0 -6.12932 -4.02793 5.79202 -0.287101 0.232499 0 2 2654
-6.12932 -4.02793 5.79202 -0.287101 0.232499 0 -6.08214 -4.03943 5.80386 -0.190759 0.245429 0 2 2654
-6.08214 -4.03943 5.80386 -0.190759 0.245429 0 -6.03423 -4.04631 5.81632 -0.0947287 0.258359 0 2 2654
-6.03423 -4.04631 5.81632 -0.0947287 0.258359 0 -5.98604 -4.04857 5.82941 0.000974371 0.271288 0 2 2654
-5.98604 -4.04857 5.82941 0.000974371 0.271288 0 -5.93804 -4.04624 5.84311 0.0963339 0.284218 0 2 2654
```

Low-level data

- Raw sensor data (e.g. LIDAR values, images)
- Neural network models (weights+topologies), learned policies (Reinforcement learning), rosbag
- File formats: (txt,bin,SQL,HDF,. . .)
- They are specific to given task/environment
- Hard to interpret by humans
- Poor generalization

Database Structure | Browse Data | Edit Pragmas | Execute SQL

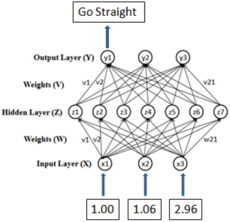
Table: runs

	id	erime	anner	iate	t_sc	solutio	iph_motic	graph_states	memory	on_clea	n_diff	ution_len	n_sec	on_smoot	solve
	Fi...	Filter	Filter	F...	F...	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Fil...	Filter	Filter
1	1	1	1	0	1	1	3155	3156	29.8047	0.0	0.0	24.0663	240	25934.3	
2	2	1	1	0	1	1	3224	3225	30.9531	0.0	0.0	26.4969	265	29009.0	
3	3	1	1	0	1	1	3380	3381	31.7266	0.0	0.0	24.9994	250	25997.5	
4	4	1	1	0	1	1	2912	2913	32.5	0.0	0.0	26.6361	267	29869.0	
5	5	1	1	0	1	1	2164	2165	33.0156	0.0	0.0	17.6915	178	16955.1	
6	6	1	1	0	1	1	3802	3803	34.0469	0.0	0.0	26.1314	262	29102.3	
7	7	1	1	0	1	1	2527	2528	35.0781	0.0	0.0	25.2687	253	29115.4	
8	8	1	1	0	1	1	4053	4054	35.5938	0.0	0.0	18.2766	183	18188.8	
9	9	1	1	0	1	1	2476	2477	36.8828	0.0	0.0	17.0454	171	17560.3	
10	10	1	1	0	1	1	3034	3035	36.8828	0.0	0.0	18.4882	185	19938.6	
11	11	1	2	0	1	1	7441	7442	38.6836	0.0	0.0	29.6971	14	13.4658	
12	12	1	2	0	1	1	11815	11816	41.0039	0.0	0.0	22.5842	11	15.7426	

Example of models of environments I

Low-level data

- Raw sensor data (e.g. LIDAR values, images)
- Neural network models (weights+topologies), learned policies (Reinforcement learning), rosbag
- File formats: (txt,bin,SQL,HDF,. . .)
- They are specific to given task/environment
- Hard to interpret by humans
- Poor generalization



T. Zhao and Y. Wang, "A neural-network based autonomous navigation system using mobile robots", International Conference on Control Automation Robotics & Vision, 2012

Example of models of environments II

Processed data

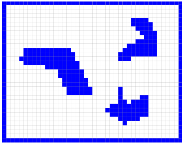
- Interpreted sensor data (e.g. obstacles, walls, ground, free-space, . . .)
- Good generalization
- Can be easily interpreted by humans

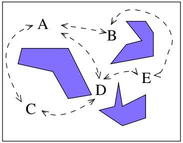

```
-0.436222 -0.0131647 0 -6.81021 -3.46388 5.68797 -0.53538 -0.000235049 0 2 2654
-0.53538 -0.000235049 0 -6.76854 -3.49147 5.68828 -0.634538 0.0126946 0 2 2654
-0.634538 0.0126946 0 -6.72982 -3.52305 5.68924 -0.73368 0.0256243 0 2 2654
-0.73368 0.0256243 0 -6.69381 -3.5577 5.69085 -0.77459 0.038554 0 2 2654
-0.77459 0.038554 0 -6.65811 -3.59264 5.6931 -0.77459 0.0514837 0 2 2654
-0.77459 0.0514837 0 -6.62244 -3.62755 5.69599 -0.77459 0.0644134 0 2 2654
-0.77459 0.0644134 0 -6.58679 -3.66243 5.69953 -0.77459 0.077343 0 2 2654
-0.77459 0.077343 0 -6.55118 -3.69728 5.70372 -0.77459 0.0902727 0 2 2654
-0.77459 0.0902727 0 -6.51561 -3.73209 5.70855 -0.77459 0.103202 0 2 2654
-0.77459 0.103202 0 -6.48009 -3.76685 5.71402 -0.77459 0.116132 0 2 2654
-0.77459 0.116132 0 -6.44463 -3.80156 5.72013 -0.77459 0.129062 0 2 2654
-0.77459 0.129062 0 -6.40922 -3.83621 5.72689 -0.77459 0.141991 0 2 2654
-0.77459 0.141991 0 -6.37388 -3.87079 5.73429 -0.77459 0.154921 0 2 2654
-0.77459 0.154921 0 -6.33861 -3.90531 5.74232 -0.77292 0.167851 0 2 2654
-0.77292 0.167851 0 -6.30173 -3.93792 5.75099 -0.675262 0.18078 0 2 2654
-0.675262 0.18078 0 -6.26195 -3.96671 5.7603 -0.577836 0.19371 0 2 2654
-0.577836 0.19371 0 -6.21967 -3.99144 5.77024 -0.480656 0.20664 0 2 2654
-0.480656 0.20664 0 -6.17532 -4.0119 5.78082 -0.383739 0.21957 0 2 2654
-0.383739 0.21957 0 -6.12932 -4.02793 5.79202 -0.287101 0.232499 0 2 2654
-0.287101 0.232499 0 -6.08214 -4.03943 5.80386 -0.190759 0.245429 0 2 2654
-0.190759 0.245429 0 -6.03423 -4.04631 5.81632 -0.0947287 0.258359 0 2 2654
-0.0947287 0.258359 0 -5.98604 -4.04857 5.82941 0.000974371 0.271288 0 2 2654
0.000974371 0.271288 0 -5.93804 -4.04624 5.84311 0.0963339 0.284218 0 2 2654
```

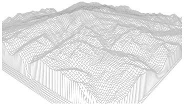

- These models are simply known as **maps**

Maps in robotics

- Map is the model of the world/environment
- Many types (2D/3D, grid, polygonal, . . .)
- Usually contain geometric features (e.g. walls, ground, obstacles)
- Necessary for decision making (e.g. planning, navigation, inspections, . . .)







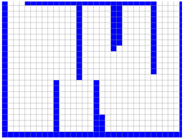
Properties

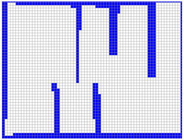
- Supported operations (e.g. merging maps, adding new information, deleting obstacles, . . .)
- Computational complexity of these procedures
- Memory requirements
- Precision
- Robustness (with respect to numerical errors)
- There is no ‘universal’ map
- One should always choose a map suitable for the given application

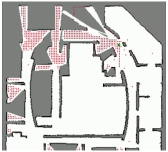
Grid maps

- 2D or 3D array (grid) of cells
- Binary maps: 0/1 (obstacle, free spaces)
- Probability: 0–1 (0=free space, 1=obstacle)
  - occupancy grid
  - often used for integration of sensor data, SLAM

✓ Metric information (distance/angle/area . . .)
✓ Easy implementation
✓ Efficient search for obstacle cells, nearest obstacle cell, . . .
✓ Straightforward update of cells & map merging
✓ Integration of data from different sensors
✗ High memory requirements
  - depends on environment size & map resolution
  - practical limit to 2D and 3D environments







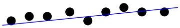
5VdzKHreB_s

Grid maps

Source: Robotic Dataset repository (Radish): fr097

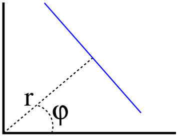
Line maps

- 2D worlds, suitable for structured indoor scenes
- Obstacles are represented by lines
- Compact way to store range-sensor measurements (x_i, y_i)



$$\tan 2\varphi = \frac{-2\sum_i (\bar{x} - x_i)(\bar{y} - y_i)}{\sum_i \left((\bar{y} - y_i)^2 - (\bar{x} - x_i)^2 \right)}$$

$$r = \bar{x} \cos\varphi + \bar{y} \sin\varphi$$



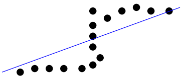
- Can be extended for 3D planes

✓ Memory efficient, easy to process, metric information

✓ Fast tests for collisions, point location

✗ What if data points are generated by multiple linear structures?

  - How many lines are needed?
  - How to assign points to individual lines?

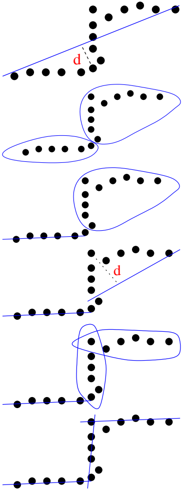


Line maps: fitting multiple lines

Split and merge recursive approach

1. Compute line for a given set of points $(x_1, y_1) \ldots (x_n, y_n)$
2. Find the most distant point i from the line, its distance is d
3. If d is smaller than a threshold, return line
4. Otherwise, split points to two groups $(x_1, y_1), \ldots (x_i, y_i)$ and $(x_{i+1}, y_{i+1}), \ldots (x_n, y_n)$ and proceed recursively on each group

- Easy to implement, fast
- The result is not optimal (does not minimize square distances of points from lines)



☛ D.H. Douglas, T.K. Peucker: Algorithms for the reduction of the number of points required to represent a line or its caricature, Cdn. Cartogr. 10(2), 1973

Polygonal maps

- 2D worlds
- Obstacle is represented by polygon

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

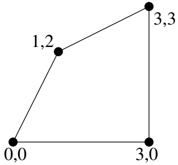
- (x_i, y_i) are vertices
- The map is the collection of obstacles
- Simple polygon: does not intersect itself, no holes
- Polygons with holes: contour + one or more holes

✓ Memory efficient, easy to process, metric information

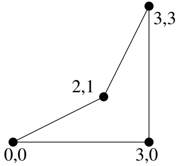
✓ Fast tests for collisions, point location

✗ Numerical stability of (some) algorithms

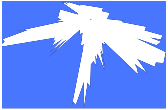
✗ Number of vertices can dramatically grow if map is built from (unfiltered) sensor data



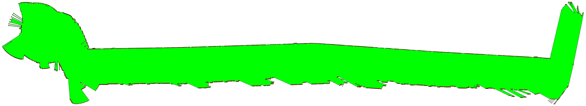
Convex



Non-convex



Polygon from Lidar



Map $\sim$ 100 $\times$ 5 m, $\sim$1k vertices

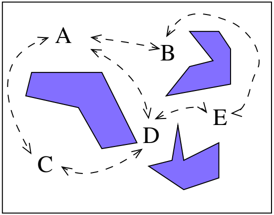
What kind of maps are using humans?

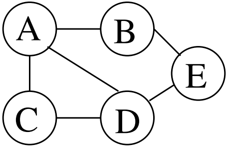
- grid, occupancy grid, polygonal, line-map . . . ?

What kind of maps are using humans?

- grid, occupancy grid, polygonal, line-map . . . ?

- Abstract map, lack of geometric and metric features
- Represented by a graph
  - Vertices are (distinguishable) places
  - Edges connecting places between the robot can navigate
- Scalable, used for high-level planning





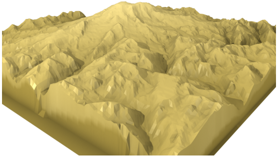
Topological maps

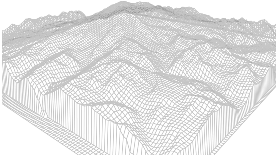
- Abstract map, lack of geometric and metric features
- Represented by a graph
  - Vertices are (distinguishable) places
  - Edges connecting places between the robot can navigate
- Scalable, used for high-level planning

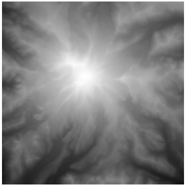
Topological maps

- Abstract map, lack of geometric and metric features
- Represented by a graph
  - Vertices are (distinguishable) places
  - Edges connecting places between the robot can navigate
- Scalable, used for high-level planning

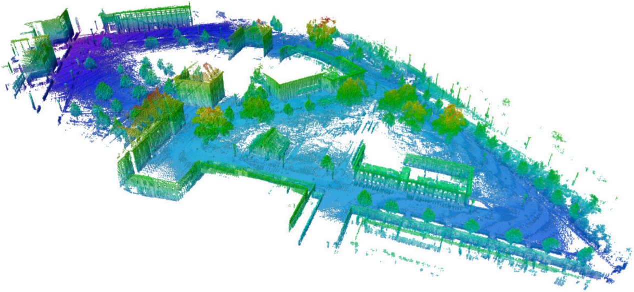
- Elevation (2.5D grid map): each cell describes altitude





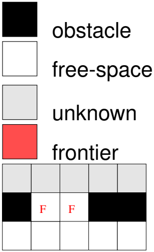


- Pointclouds, octomap



Occupancy grid

- A variant of grid map
  - Known cell: value of $c_i \geq 0$ (contains prob. of being occupied)
  - Unknown cell: value of $c_i = -1$
- Interpretation of known cells:
  - Free-space (no obstacle): $p(occupied) < T$
  - Obstacle: $p(occupied) > T$
  - where T is a threshold, e.g. 0.8
- Frontier: the border between known and unknown regions
- Frontier cell
  - is a free-space cell that is incident with an unknown cell
  - it may not be reachable



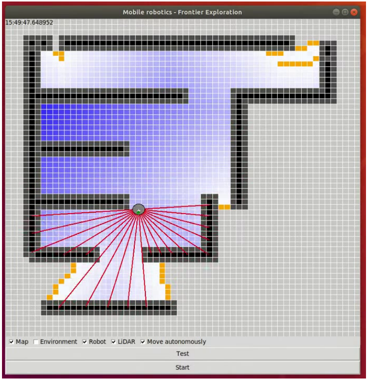
☛ YAMAUCHI, B., et al. Frontier-based exploration using multiple robots. Agents. 1998; 47-53.

Robotic exploration

- Robot is gathering (desired) information in an environment
- Search & rescue, searching for Barbie, precious metals, etc.
- We used model of the environment (map) to do it efficiently
- Often used solution: SLAM (Simultaneous localization and mapping)

Challenges

- How to represent the map
- How to update it
- How to localize
- How to determine where to go
- How to get there



https://www.youtube.com/watch?v=B-dSyKx4Fsc

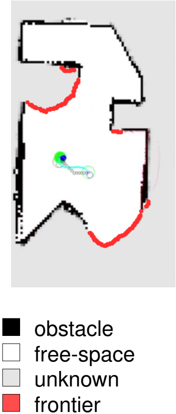
Frontier-based exploration

Principle: use a frontier as a temporary goal

1. Identify frontiers in the map
2. Filter out unreachable frontiers (if any)
3. Select a frontier and go there
4. Goto 1 until no frontier exists

Notes

- Unreachable frontiers detected using path planning
- Consider navigating to the closest frontier
- Consider detecting frontiers during movement of the robot
- Detection of frontiers should be fast



☛ YAMAUCHI, B., et al. Frontier-based exploration using multiple robots. Agents. 1998; 47-53.

☛ KEIDAR, Matan; KAMINKA, Gal A. Ecient frontier detection for robot exploration. The International Journal of Robotics Research, 2014, 33.2: 215-236.

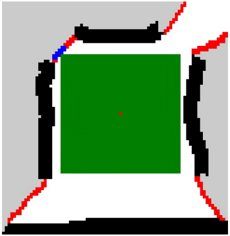
- Image-based
    - Convert occupancy grid to binary image, run edge detection
- Wavefront Frontier Detector (WFD) (☛ Keidar)
    - Graph-search method to detect frontiers
    - Run BFS from actual position of the robot
    - This BFS explores only free cells (i.e., also frontier cells)
    - Run another BFS if frontier cell is visited
    - The second BFS explores only frontier cells
    - The goal of second BFS is to extract all cells belonging to the actually detected frontier
- Both BFS share open/close list

- obstacle
- free-space
- unknown
- frontier

☛ YAMAUCHI, B., et al. Frontier-based exploration using multiple robots. Agents. 1998; 47-53.
☛ KEIDAR, Matan; KAMINKA, Gal A. Ecient frontier detection for robot exploration. The International Journal of Robotics Research, 2014, 33.2: 215-236.

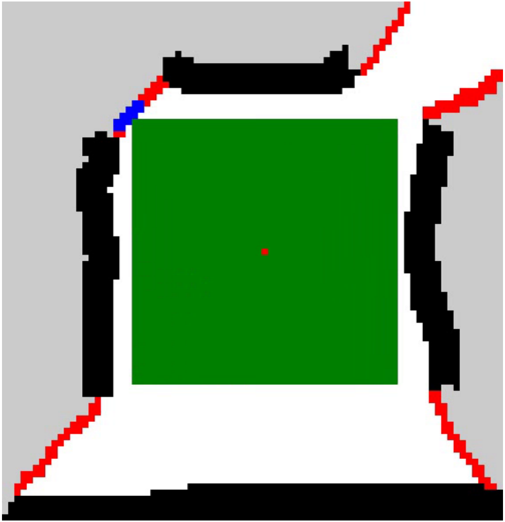
Frontier-based exploration


```
O = { robot position }
while |O| ≥ 0 do
    pos = O.pop()            // open list for main BFS
    mark pos as known
    for c in neighbors(pos) do
        if c is explored or c is obstacle then
            continue
        if c is frontier cell then
            O_f = {c}   // open list for second BFS
            while |O_f| ≥ 0 do
                pos_f = O_f.pop()
                mark pos_f as known
                for c_f in neighbors(pos_f) do
                    if c_f is not frontier cell or c_f is known
                      then
                        continue
                    O_f.push(c_f)
        else
            O.push(c)
```



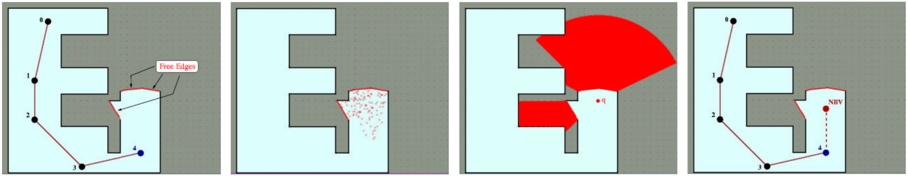
Explanation of variables

- pos, pos_f = coordinates of a cell in grid, e.g. (x, y)
- robot position = actual position of the robot in the grid
- O: open list for main BFS search — free cells
- O_f: open list for exploration of individual frontiers — frontier cells



Several ideas to get better (faster) exploration

- Consider cost of path to the frontier for frontier selection
- Consider how much are is ‘behind’ the frontier (aka ’view’), visit the most promising frontiers first → next best view approach
- Combination of above



☛ Gonzalez-Banos, H. H.,Latombe, J. C. (2002). Navigation strategies for exploring indoor environments. The International Journal of Robotics Research, 21(10-11), 829-848.

Frontier-based exploration: resources

- YAMAUCHI, Brian, et al. Frontier-based exploration using multiple robots. In: Agents. 1998. p. 47-53.
- TOPIWALA, Anirudh; INANI, Pranav; KATHPAL, Abhishek. Frontier Based Exploration for Autonomous Robot. arXiv preprint arXiv:1806.03581, 2018
- USLU, Erkan, et al. Implementation of frontier-based exploration algorithm for an autonomous robot. In: 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA). IEEE, 2015. p. 1-7.
- KEIDAR, Matan; KAMINKA, Gal A. Ecient frontier detection for robot exploration. The International Journal of Robotics Research, 2014, 33.2: 215-236.