On Autonomous Mobile Robot Exploration Projects in Robotics Course

Jan Faigl[0000-0002-6193-0792], Miloš Prágr[0000-0002-8213-893X], and Jiří Kubík[0000-0002-2219-5764]

Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Czechia
{faiglj|pragrmi1|kubikji2}@fel.cvut.cz

Abstract. Autonomous mobile robot exploration can be considered a representative task where multiple problems need to be addressed and solutions integrated into a software framework that exhibits desired autonomous behavior of the robot. The problem includes online decision-making in selecting new navigational waypoints towards which the robot is autonomously navigated to explore not yet covered part of the environment. A mobile robot's navigation consists of localization, mapping, planning, and execution of the plan by following the path toward the waypoint. For these very reasons, we decided to include mobile robot exploration as one of the tasks in our Artificial Intelligence in Robotics course opened at the Faculty of Electrical Engineering, Czech Technical University in Prague. In this paper, we present our experience running the course, where the students start with relatively small isolated tasks that are then integrated into a full exploration framework. We share the students' feedback on our initial approach for the task that becomes a mandatory part of the course evaluation and grading.

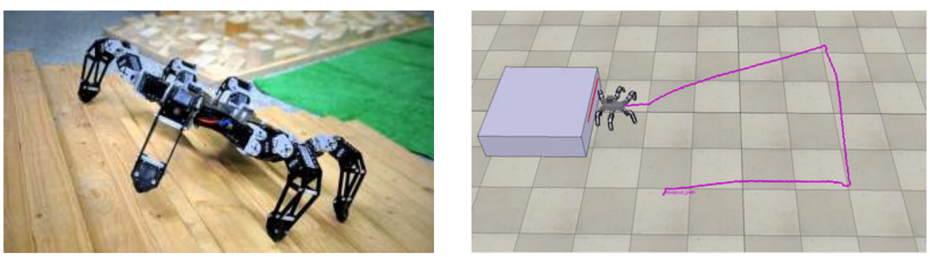
Keywords: Robot Course Design, Autonomous Robot Exploration.

1 Introduction

In 2016, we faced a challenge to update our Computer Science study program with a branch specialized in Artificial Intelligence (AI). Studying AI has a long tradition at the Czech Technical University in Prague (CTU) with an overlap to image processing, computer vision, and machine learning. We have decided to include robotics in the AI curricula to offer students an opportunity to experience hands-on robotics systems, where it is necessary to deal with sources of uncertainty in sensing and acting. We proposed a new course on AI in robotics [2] to provide an overview of robotic paradigms, path and motion planning methods, and environment modeling approaches. The first part of the course targets to combine particular tasks in the autonomous navigation problem, with autonomous exploration selected as one of the central problems of the course.

Mobile robot exploration is a problem to develop a system that operates one or multiple robots in a priory unknown environment intending to model some phenomenon, for example, building a geometrical environment map. From the AI perspective, exploration combines processing of the sensory inputs, environment modeling, reasoning about the future robot actions, planning, and navigation towards the determined waypoints. Besides, multi-robot exploration further extends the opportunity to study approaches for multi-robot coordination [9] and cooperation using centralized, distributed, or decentralized methods [29,4]. Exploration directly relates to search-and-rescue scenarios, where multi-criteria decision-making can be applied [3]. Besides, considering multi-legged robots [6] opens further challenges in motion planning and locomotion control.

Throughout the history of the course, we have initially started with focused lab exercises that have guided the students through individual building blocks of autonomous exploration. Later, we switched to a block of individual tasks that allowed the students to experience locomotion control, navigation, mapping, and decision-making. Students submit the individual tasks into our automated evaluation system called BRUTE (Bundle for Reservation, Upload, Test, and Evaluation), allowing them to work on the assignments outside the course's scheduled hours. The culmination of the student's effort is the semestral course project, where individual building blocks form an initial exploration framework that needs to be improved by various means. The improvements are mostly expected in decision-making using AI techniques that can be acquainted with the course lectures or selected research papers, allowing the students to understand and apply novel results from the literature. Teachers evaluate the projects, and students are requested to defend the project during the course exam, thus verifying the students' work on the project and their understanding of the topic.



(a) Real hexapod walking robot (b) Hexapod robot in the CoppeliaSim

Fig. 1: Hexapod walking robot utilized for the exploration task in the course.

From the beginning, we planned to target the exploration task to our hexapod walking robot depicted in Fig. 1a because students can deploy their solutions on the real robot during the course labs. The robot is built from Robotis Dynamixel AX12A servomotors, using the adaptive locomotion controller [17] capable of negotiating the terrain and supporting the robot's endurance and robustness using the students' code. Besides, we have selected the CoppeliaSim (formerly V-REP) simulation environment [12] because it is a multiplatform, stand-alone simulation environment with multiple interfaces, including C/C++ and Python.

A model of the robot is available as shown in Fig. 1b, which supports direct deployment from the simulator to the real robot.

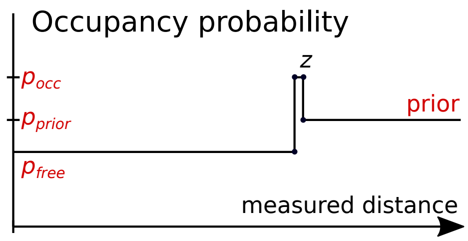
The course is primarily implemented in Python, allowing for rapid prototyping. In 2019, we opted for ROS (Robot Operating System) [25] to support the integration of the students' modules. However, in the regular university courses' evaluation, students complained about the initial challenges of setup the ROS environment, despite a dedicated lab for familiarizing with the necessary ROS infrastructure [26] suitable for exploration tasks. We acknowledged the complaints as ROS distracts attention towards the implementation aspects than the principal and algorithmic solutions. The course is in the last year of the AI master studies, where most students do not get in touch with ROS during their studies unless they take dedicated courses on robotics. We found that without ROS, the students progress faster, enjoy the course more, and dedicate more hours to advanced modules enhancing the basic exploration strategy.

The rest of the paper is organized as follows. A brief overview of the mobile robot exploration is summarized in the following section to familiarize the reader with the autonomous exploration framework's principles and basic building blocks. In Section 3, a description of the students' tasks and project assignments are presented. Results of the evaluation of the student's achievements are reported in Section 4. Finally, the paper is concluded in Section 5.

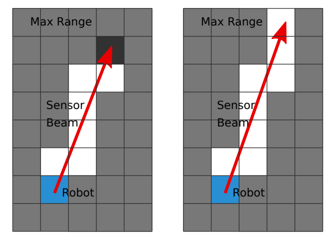
2 Mobile Robot Exploration

We consider exploration a representative task of autonomous behavior with online decision-making, while we restrict it to the problem of creating a map of a priory unknown environment for the course. It consists of fusing sensor measurements into the environment model and making decisions about the next robot's actions. For educational purposes, we limit the task to 2D mapping with an occupancy grid map [24]. Then, we can employ frontier-based exploration [28] to determine the navigational waypoints at the border of known free space and not yet explored part of the environment. The robot explores the environment while being navigated toward frontiers.

The occupancy grid map is a discretized spatial environment representation where each cell is associated with an occupancy probability value. Furthermore, we assume that a robot pose estimate is available. Hence, the sensor measurements are integrated using the Bayes filter with the sensor model. For simplicity, we utilize LiDAR-based sensors with relatively precise distance measurements to the obstacles; see Fig. 2a. The grid cells of the occupied grid map are updated using laser beam raycasting. The update of the individual grid cells is along a line determined by Bresenham's line algorithm [8] as depicted in Fig. 2b. Thus, the mapping is relatively straightforward and reflects dynamic obstacles as the map is continuously updated. Note that using a 2D grid map, various techniques of path planning can be employed, such as A*, Theta* [13], JPS [19], and D* Lite [21]. Hence, students can deploy various methods and see their impact on online decision-making.



(a) Laser model



(b) Ray-casting in a grid map

Fig. 2: Laser sensor model and update of the grid map. The probability values p_{occ} and p_{free} denote the probability of occupied and free, respectively.

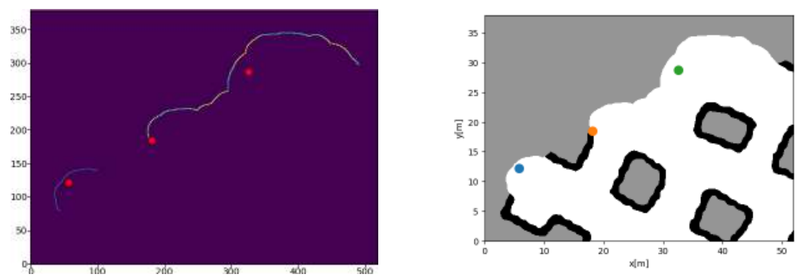


Fig. 3: Representative of frontiers determined as centroids of free-edge clusters.

Possible exploration strategies can range from improved frontier-based approaches [23] to information-based methods [1] using mutual information [10]. In multi-robot cases, the problem can be addressed as task allocation [18], or robots can make decisions independently [29]. Furthermore, decisions can be myopic, considering only the immediate reward when selecting the next waypoint, such as the closest waypoint. Alternatively, a suitable option for applying AI techniques is to perform non-myopic decisions considering a longer horizon. It can lead to a solution of the Traveling Salesman Problem (TSP) instance [30] and decisions made by the so-called TSP distance cost [22], which is also applicable in multi-robot exploration [16]. Since obtaining a solution for the TSP can be demanding, heuristics can be considered. Besides, considering individual frontier cells might be unnecessary, and therefore, representatives of the frontiers' free edges can be more suitable [14]. Therefore, determining frontiers representatives provides an opportunity to employ various AI methods such as unsupervised clustering as depicted in Fig. 3.

2.1 Building Blocks of Mobile Robot Exploration Framework

Although autonomous mobile robot exploration can be considered a complex problem, we build on our experience and effort on benchmarking exploration strategies [15] and summarize the problem into the following steps.

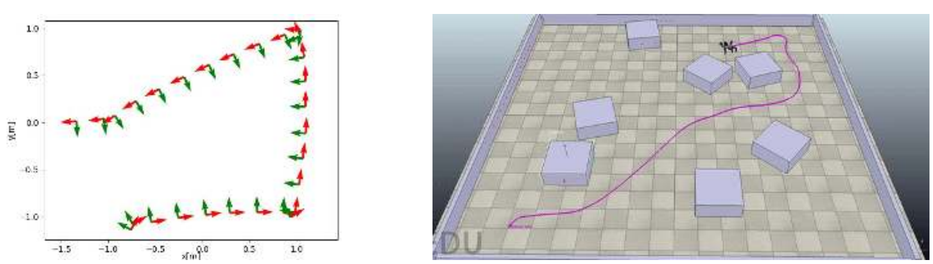
1. Initialization of the occupancy grid map and integrating of the first sensor measurements.
2. Creating a navigation grid map by thresholding the occupancy probability into free space, obstacles, and unknown areas, e.g., using probability threshold values 0.4 and 0.6.
3. Determining the next navigational waypoint, the exploration strategy.
4. Planning a path to the waypoint.
5. Navigating the robot along the planned path and integrating the new sensor measurements into the map.
6. Repeat the procedure from Step 2 when a replanning is triggered.

We can identify three main processes that can run in parallel and that combine building blocks of control, path planning, mapping, and decision-making. The first is mapping, which collects and integrates measurements into the occupancy grid map. It can run relatively fast depending on the robot's velocity. The second is a path following, responsible for determining control commands for the robot to navigate along the planned path. We can follow the hybrid robotics paradigm and employ reactive collision avoidance to ensure safe navigation along the planned path. The path following runs in a loop with a similar frequency as the mapping. Finally, the third process is computing the new plan, which consists of determining the next exploration goal and planning the respective path. The plan computations can run in a loop with a relatively small frequency or be triggered by reaching the previous exploration goal.

We define a sequence of small tasks based on the building blocks, each focusing on a particular subproblem combined in a simple frontier-based exploration framework. The tasks are summarized in the following section.

3 Exploration Task Assignments

Students are prepared to develop an autonomous mobile robot exploration framework through five consecutive tasks guiding them in implementing robot control, mapping, planning, and determining the exploration waypoints. The task assignments are briefly described in the following paragraphs.

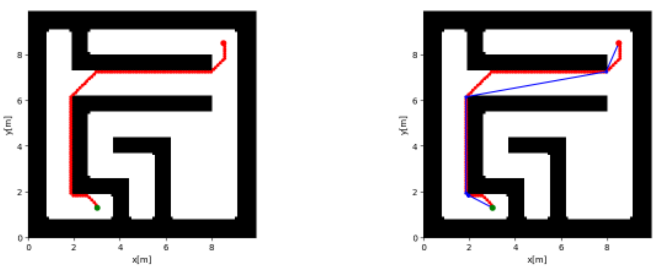


(a) T1a-ctrl (b) T1b-react

Fig. 4: Expected robot path in the robot control tasks T1a-ctrl and T1b-react.

T1a-ctrl - Open-loop locomotion control – First, students get familiarized with the CoppeliaSim environment and control the robot in an open-loop fashion. The task is to implement a `goto` function that steers the robot towards the desired goal position using a velocity command consisting of the desired forward and angular speed of the robot that is passed to the provided controller. The robot is supposed to reach a sequence of positions as depicted in Fig. 4a.

T1b-react - Reactive obstacle avoidance – Next, students are requested to improve the navigation capabilities of the robot through sensory-motor feedback. Again, students implement the `goto` function that determines the robot steering command based on the sensory input. The accompanied labs guide students with Bug and Bug2 algorithms and AI models of Braitenberg vehicles [7]. The expected behavior of the robot is as depicted in Fig. 4b.



(a) Grown obstacles (b) Path simplification

Fig. 5: Grid-based path planning and path simplification. The red curve is the planned grid path. The simplified path with waypoints where the robot changes its orientation is in the blue.

T1c-plan - Path planning – The third task is grid-based path planning, where students are requested to deploy some graph (or grid) based search technique such as A*, which they already know from the AI courses. However, within the robotics context, the obstacles need to be grown to consider the embodiment of the robot; see Fig. 5a. Besides, the grid path represented as a sequence of neighboring grid cells can be too dense to be smoothly followed. Therefore, students are tasked to simplify the path to a sequence of waypoints; see Fig. 5b. Such a sequence of waypoints can be utilized in the reactive controller from task T1b. Students implement the functions `grow_obstacles`, `plan_path`, and `simplify_path` that are validated in the BRUTE for defined planning scenarios. The validation procedure is available with the reference solutions bundled in Python's `pickle` object serialization.

T1d-map - Mapping – In the mapping task, students are to fuse laser scanner measurements by implementing the `fuse_laser_scan` function. For the validation, students can utilize the reactive controller from T1b-react to capture scans along a defined path as depicted in Fig. 6. Besides, similar to the previous task,

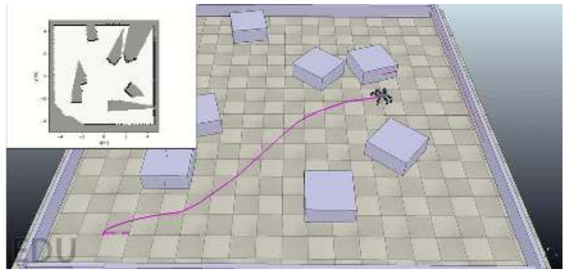


Fig. 6: Example of the robot path in the CoppeliaSim with visualization of the grid map created by thresholding the occupancy grid map. The gray cells denote unknown parts; the obstacles are black, and the free space is white.

testing scenarios are available. The automated task evaluation is based on computing an accumulated difference of the map fused by the student's implementation and the reference map. In real deployments, a small difference in a map built over multiple trials can be expected because of numerical issues and uncertainty in the pose estimation that is, however, provided by the CoppeliaSim. Differences between the reference and student maps are expected even in the simulation since students can experiment with different parametrizations of the Bayesian map update.

T1e-expl - Exploration – Finally, students are tasked to implement the function `find_free_edge_frontiers` that is supposed to provide a list of positions representing possible exploration waypoints. Here, students can exploit advanced functions of the SciPy [27] and use only a few lines of code to determine frontiers cells, cluster them, and find free edge representatives as depicted in Fig. 3. To that end, students need to reason about the problem, proper formulation, and design of the convolution mask used to identify the frontier cells.

As the result of the overviewed tasks, students have prepared the individual functions to be integrated into the exploration framework consisting of three threads working in parallel as described in Section 2.1. Besides, when implementing the tasks, students are provided with a support code in Python that handles communication with the simulator, low-level robot motion control given the velocity command, and data types for sensory inputs.

4 Evaluation of the Students' Achievements

Students submit tasks' implementations and project into BRUTE, where they can report time spent on the particular task. The time reporting is voluntary, but students fill reasonable values in most cases; therefore, we only filter out outlier values longer than 100 hours, which is unrealistic, specifically as they have one to two weeks per each small task. The filtered reported time spent is depicted as the five-number summary in Fig. 7.

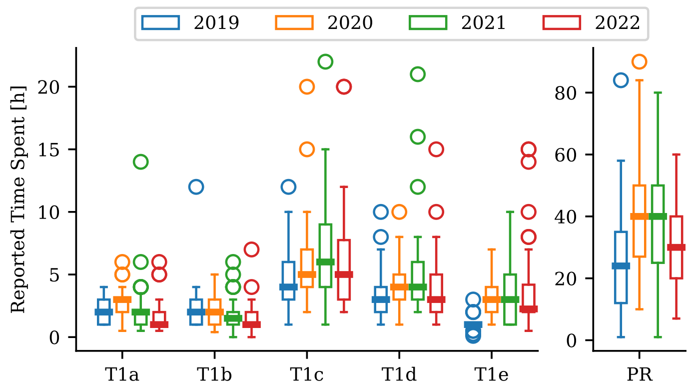


Fig. 7: Reported students times spent on the tasks T1a–e and project PR.

The maximal reported values support the suitability of outliers being thresholding at 100 h. Besides, in the authors' opinion, it might be the case that students often round the reported hours to tens for the project. The maximum hours spent on the project correspond to more than ten working days, which might be realistic if a student struggles with programming. However, most of the values are around half of that, which is our expected value. The success of our teaching mission can be further evaluated by an in-depth analysis of how students spent the time budget and what features they implemented within the autonomous mobile robot exploration project, detailed in the rest of this section.

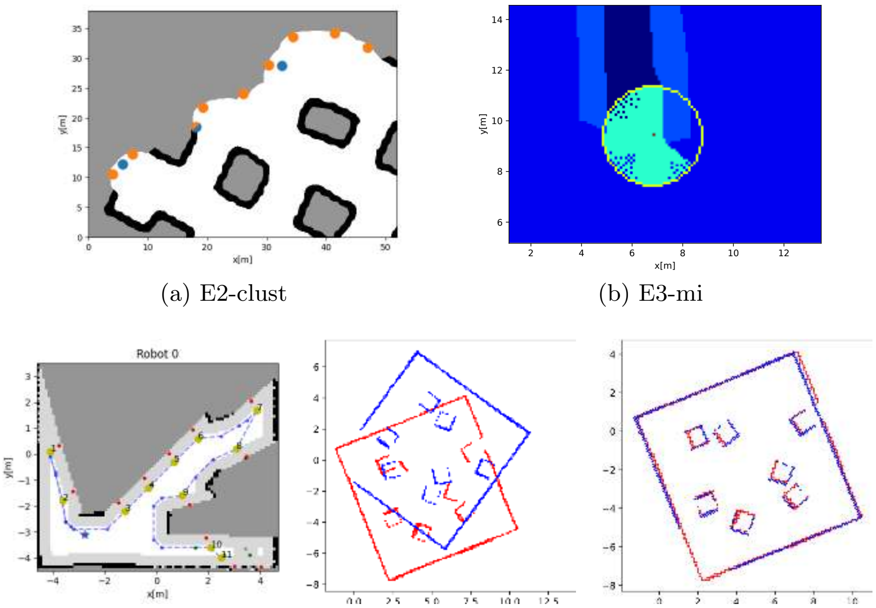
4.1 Selected Students' Implementations

We can imagine various extensions of the exploration framework to improve mission performance. For students' convenience, we list possible extensions with expected scoring as the project can be up to 30 % of the course grading. However, students are not limited to the list and are encouraged to discuss the viability of other extensions with lab instructors. The mandatory implementation of the project represents about 10 % corresponding to 10 points. Seven selected extensions are further discussed as follows.

E1-dmap - Dynamic map size (+2 points) – Since a fixed size, a sufficiently large map is assumed in the T1d-map task, the extension is to implement a dynamically resizable map representation.

E2-clust - Multiple-representative free-edge clusters (+2 points) – A single representative is determined even for a long free-edge in the T1e-expl task; hence, students can implement splitting long free-edges using [16] as depicted in Fig. 8a.

E3-mi - Mutual information (+7 points) – Multiple representatives can be ranked by considering the information gained by observing their surroundings [10], where each cell yields information based on the entropy of its obstacle-probability and cells are considered independent. Since assessing the expected information gain can be demanding, students can implement a simplified solution that approximates the information gain as the sum over all cells within



(a) E2-clust (b) E3-mi

(c) E4-tsp (d) Individual frames (e) E7-icp

Fig. 8: Illustrations of the suggested project extensions. (**a**) single representative of the free edge is in blue, while multiple representatives are in orange; (**b**) Raycasting for determining visible cells in E3-mi; (**c**) TSP-based selection of the exploration waypoint in E4-tsp; (**d**) and (**e**) maps without and with the ICP-based transformation of the coordinate frames, respectively, in E7-icp.

the sensor range, e.g., using raycasting to determine the visible cells as depicted in Fig. 8b.

E4-tsp - Non-myopic planning (+4 points) – Non-myopic decision-making can be implemented as a solution of an open-ended TSP instance, as illustrated in Fig. 8c. Students are expected to utilize the LKH solver [20], already used during the courses. However, they can also use alternative solvers.

E5-mre - Multi-robot exploration (+5 points) – Using multiple exploring robots can improve exploration performance, and students can use various technical solutions, such as inter-process communication.

E6-dec - Multi-robot exploration with decentralized task allocation (+8 points) – Further extension of multi-robot exploration towards decentralized decision-making is suggested to be based on the MinPos [4], where each robot ranks possible waypoints based on its and other robots' distance to the waypoint being ranked.

E7-icp - Multi-robot exploration with individual coordinate frames (+10 points) – In the CoppeliaSim, the robots' positions are reported in a common coordinate frame. Students can assign each robot its coordinate frame based on its respective initial position; see Fig. 8d. Then, the transformation between the coordinate frames can be found by the Iterative Closest Point (ICP) algorithm [11] to get a joint coordinate frame as depicted in Fig. 8e.

Table 1: No. of students implementing the individual project extensions

Year	PR	E1-dmap	E2-clust	E3-mi	E4-tsp	E5-mre	E6-dec	E7-icp
2021	42	31	39	30	19	21	12	1
2022	33	21	30	26	13	18	14	0

We collected students' implementations from 2021 and 2022, where 42 and 33 students implemented at least one extension of the exploration project among 55 and 66 enrolled students in the course, respectively. The distribution of the implemented extensions is reported in Table 1.

The distribution suggests that students prefer to implement single-robot extensions over multi-robot options. Overall, the dynamic map size (E1-dmap), multiple-representative free-edge clusters (E2-clust), and mutual information (E3-mi) extensions are the most popular. The popularity of E3-mi is somewhat surprising to the course instructors since mutual information computation is relatively complex compared to the other popular options; however, closer inspection reveals that more than one-third of the students opted for the simplified version each year omits raycasting, making the computation significantly easier. Among the single robot extensions, the TSP-based planning (E4-tsp) is the least popular, likely due to their choice to work with an external library, for which students report issues when using MacOS and Windows.

About half of the students opted to use multiple robots (E5-mre). The data suggest that after implementing the multi-robot exploration, the students are motivated to add the advanced task allocation (E6-dec), with about two-thirds doing so. The exploration without a common coordinate frame (E7-icp) is selected rarely, likely because it requires an extensive modification of the provided supporting code. The only student who implemented the extension in 2021 noted his interest in mobile robot localization.

One more student has implemented ICP-based matching of the robot scan to a priori prepared maps of the environment; however, although similar, the extension was part of a single robot exploration and evaluated as a custom extension. Other custom extensions include D* Lite path planning [21], an obstacle distance field [5], or a ROS2-based implementation.

Overall, the project is popular with students, showing a considerable participation rate through non-trivial and custom extensions.

5 Conclusion

In this paper, we share our experience on autonomous mobile robot exploration in a robotic course; within it, we aim to prepare the students for the final project - an exploration framework - through a sequence of small tasks that are then integrated into the framework. Based on the students' progress and reported feedback, the students enjoy the way of small incremental steps to build a solution for a relatively complex behavior of autonomous exploration, which most students would not imagine at the beginning of the course. Although the small tasks have been fixed for several years, and our submission system automates evaluating the students' solutions, we do not detect significant plagiarism. The students acknowledge the tasks as steps to understand the topics and the incremental way of building a complex solution to avoid frustration from being overwhelmed. Based on our course experience, we prepared a dedicated, short, a few days long course[1] with four introductory lectures and a series of tasks leading to a robotic exploration framework. The supporting files can be used for further similar courses.

Acknowledgements

The work on mobile robot exploration has been supported by the OP VVV-funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics." The effort of Petr Čížek and Jan Bayer on tasks implementation and Daniel Večerka on BRUTE design are also gratefully acknowledged.

References

1. Amigoni, F., Caglioti, V.: An information-based exploration strategy for environment mapping with mobile robots. Robotics and Autonomous Systems **58**(5), 684–699 (2010)
2. Course Web Page - B4M36UIR and BE4M36UIR - Artificial Intelligence in Robotics. `https://cw.fel.cvut.cz/wiki/courses/uir/start`, accessed: 2023-01-26
3. Basilico, N., Amigoni, F.: Exploration strategies based on multi-criteria decision making for searching environments in rescue operations. Autonomous Robots **31**(4), 401–417 (2011)
4. Bautin, A., Simonin, O., Charpillet, F.: Minpos : A novel frontier allocation algorithm for multi-robot exploration. In: International Conference on Intelligent Robotics and Applications (ICIRA). pp. 496–508 (2012)
5. Bayer, J., Faigl, J.: On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In: European Conference on Mobile Robots (ECMR). pp. 1–6 (2019)
6. Bouman, A., Ginting, M.F., Alatur, N., Palieri, M., Fan, D.D., Touma, T., Pailevanian, T., Kim, S.K., Otsu, K., Burdick, J., a. Agha-Mohammadi, A.: Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2518–2525 (2020)
7. Braitenberg, V.: Vehicles, experiments in synthetic psychology. MIT Press (1984)
8. Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Systems Journal **4**(1), 25–30 (1965)
9. Burgard, W., Moors, M., Stachniss, C., Schneider, F.E.: Coordinated multi-robot exploration. IEEE Transactions on Robotics **21**(3), 376–386 (2005)
10. Charrow, B., Liu, S., Kumar, V., Michael, N.: Information-theoretic mapping using Cauchy-Schwarz Quadratic Mutual Information. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 4791–4798 (2015)

[1] `https://cw.fel.cvut.cz/wiki/courses/crl-courses/redcp/start`

11. Chen, Y., Medioni, G.: Object modeling by registration of multiple range images. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 2724–2729 (1991)
12. CoppeliaSim. `https://www.coppeliarobotics.com/`, accessed: 2023-01-26
13. Daniel, K., Nash, A., Koenig, S., Felner, A.: Theta*: Any-angle path planning on grids. Journal of Artificial Intelligence Research **39**(1), 533–579 (2010)
14. Faigl, J., Kulich, M.: On determination of goal candidates in frontier-based multi-robot exploration. In: European Conference on Mobile Robots (ECMR). pp. 210–215 (2013)
15. Faigl, J., Kulich, M.: On benchmarking of frontier-based multi-robot exploration strategies. In: European Conference on Mobile Robots (ECMR). pp. 1–8 (2015)
16. Faigl, J., Kulich, M., Přeučil, L.: Goal assignment using distance cost in multi-robot exploration. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3741–3746 (2012)
17. Faigl, J., Čížek, P.: Adaptive locomotion control of hexapod walking robot for traversing rough terrains with position feedback only. Robotics and Autonomous Systems **116**, 136–147 (2019)
18. Gerkey, B.P., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research **23**(9), 939–954 (2004)
19. Harabor, D., Grastien, A.: Improving jump point search. Proceedings of the International Conference on Automated Planning and Scheduling **24**(1), 128–135 (2014)
20. Helsgaun, K.: An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. European Journal of Operational Research **126**(1), 106–130 (2000)
21. Koenig, S., Likhachev, M.: Fast replanning for navigation in unknown terrain. IEEE Transactions on Robotics **21**(3), 354–363 (2005)
22. Kulich, M., Faigl, J., Přeučil, L.: On distance utility in the exploration task. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 4455–4460 (2011)
23. M. Wurm, K., Stachniss, C., Burgard, W.: Coordinated multi-robot exploration using a segmentation of the environment. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1160–1165 (2008)
24. Moravec, H.: Sensor fusion in certainty grids for mobile robots. AI Magazine **9**(2), 61–74 (1988)
25. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. vol. 3, p. 5 (2009)
26. ROS - Robot Operating System. `https://www.ros.org/`, accessed: 2023-01-26
27. SciPy - Fundamental algorithms for scientific computing in Python. `https://scipy.org/`, accessed: 2023-01-26
28. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA). pp. 146–151 (1997)
29. Yamauchi, B.: Decentralized coordination for multirobot exploration. Robotics and Autonomous Systems **29**, 111–118 (1999)
30. Zlot, R., Stentz, A., Dias, M., Thayer, S.: Multi-robot exploration controlled by a market economy. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 3016–3023 (2002)