This is an accepted version of the article: Kulich, M., Vonásek, V., and Přeučil, L. (2014). **Simulation-Based Goal-Selection for Autonomous Exploration**. *Modelling and Simulation for Autonomous Systems*, which has been published in a final form at https://doi.org/10.1007/978-3-319-13823-7_16.

Simulation-based goal-selection for autonomous exploration

Miroslav Kulich, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, kulich@labe.felk.cvut.cz, Czech Republic
Vojtěch Vonásek, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, vonasek@labe.felk.cvut.cz, Czech Republic
Libor Přeučil, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, preucil@labe.felk.cvut.cz, Czech Republic

Abstract

High-level planning can be defined as the process of selection of an appropriate solution from a set of possible candidates. This process typically evaluates each candidate according to some reward function consisting of (1) cost, i.e., effort needed to accomplish the candidate and (2) the utility of accomplishing it and then selects the best one according to this evaluation. The key problem lies in the fact that the reward function can be rarely evaluated precisely. At the example of the problem of exploration of an unknown environment by a modular robot we show that precise simulation-based estimation of the cost function leads to better decisions of high-level planning and thus improves exploration process performance. State-of-the-art techniques compute the cost function in goal-selection as a length of the path from the current robot position to a goal-candidate. This is sufficient for robots with simple kinematics for which time to reach a candidate highly correlates with a path length. As this does not hold for complex (modular) robots, we introduce the approach that generates a feasible trajectory to each goal-candidate (taking into account kinematic constrains of the robot) and determines the cost function as time needed to perform this trajectory in a simulator. The experimental results with a robot consisting of eight modules operating in several environments show that the proposed simulation-based solution outperforms standard solutions.

Keywords: exploration, modular robot, simulations, goal-selection

1 Introduction

Modular robots consist of many small interconnected robotic modules. Depending on hardware architecture, the modules can be equipped with various actuators to achieve a motion, basic sensors, communication bus and a simple processing unit; we refer to [17] for a comprehensive survey of existing modular platforms. In comparison to conventional fixed-shapes robots, which are usually built for performing a specific job, the modular robots are more flexible as they can be reconfigured to various shapes according to a target application. It may bring additional abilities in applications like space exploration [26], search & rescue missions [25, 3] or object manipulation [12].

For example, a quadruped modular robot can be used for efficient motion over a complex terrain. When the robot needs to pass a narrow space, it can reconfigure to a snake-like robot. Beside the ability to form robots of various shapes, the advantage of modular robots stands in the possibility of changing failed modules by simple reconnection for the organism.

The aforementioned properties predetermine modular robots for search and rescue missions. A typical task in these missions is exploration, which is the process of autonomous navigation of a robot in an unknown environment in order to build a model (map) of the environment. A natural condition is to perform the exploration with a minimal usage of resources, e.g. trajectory length, time of exploration, or energy consumption.

The key component of the exploration algorithm is determination of a next goal in each exploration step, which can be made in two stages: candidates for the next goal are found first and the most appropriate one is selected then. Two main streams for goal candidates generation were developed and widely used in the robotic community. Yamauchi [23] introduced a frontier-based approach, where candidates are placed at a border between the space detected as free and an unexplored area (frontiers). Gonzalez-Banos and Latombe [7] generate candidates in the free space and within the visibility range of frontiers (free curves in their terminology).

Many techniques evaluating the goal candidates were studied in the last fifteen years. They are based only on estimation of the effort needed to reach the goal (e.g. distance cost) [23, 11] or they combine a distance cost with other criteria. In [7], a measure $A(q)$ of an unexplored region of the environment, which is potentially visible from the candidate q, is combined with the distance cost $L(q)$ to get the overall utility of q:

$$g(q) = A(q)e^{-\lambda L(q)},$$

where λ is a positive constant. A utility of the next action as the weighted sum of the distance cost and expected information gain computed as a change of entropy after performing the action is presented in [19]. Another strategy taking into account the distance cost and the information gathered (based on the relative entropy) is introduced in [1] together with solid mathematical foundations. Moreover, the localization utility can be integrated into the overall utility to prefer places traveling to them improves information about the robot pose [16]. Criteria forming the overall utility are not typically independent. General approach that reflects dependency among the criteria based on multi-criteria decision making is used in [2].

An exploration framework for a modular robot is presented in this paper with a special attention to a distance cost definition. Contrary to the aforementioned approaches that evaluate the distance cost simply as the length of the path from the current robot position to the next goal position produced by a standard graph-based planners (Dijkstra, A*) or potential fields, we use the estimation of robot's real trajectory based on RRT (Rapidly-Exploring Random Tree) planner and a physical model of the robot. We show that realistic estimation of robot's effort needed to reach the goal can improve performance of the whole exploration process.

In order to focus on studying distance cost influence, several simplifications are made. First, a robot operates on a flat, smooth, and uniform terrain and it is equipped with a 2D range-finder (e.g. laser) that is always oriented parallel to the terrain. Moreover, the workspace is static and obstacles are detectable by the range-finder. Finally, we assume a fixed configuration of the modular robot, i.e. the robot can not change its structure (for example, it can not split into individual parts).

The rest of the paper is organized as follows. In Section 2 the exploration problem is defined, while Section 4 deals with RRT planning for a modular robot. A comparative study of distance is presented in Section 5 and concluding remarks in Section 6.

2 Problem definition

Exploration is the process in which a robot autonomously operates in an unknown environment with the aim to create a map of it. The map is built incrementally as actual sensor measurements are gathered and it serves as a model of the environment for further exploration steps.

The exploration algorithm consists of several steps that are repeated until some unexplored area remain. The process starts with reading actual sensor information. After some data processing, the existing map is updated with this information. New goal candidates are then determined and the next goal for the robot is assigned using a defined cost function. This assignment is called exploration strategy and can be formalized as follows.

Let the current n goal candidates be located at positions $\boldsymbol{G} = \{g_1, \ldots, g_n\}$. The problem is to determine a goal $g \in \boldsymbol{G}$ that will minimize the total required time (or the maximal traveled distance) needed to explore the whole environment.

Having assigned the goal, the shortest path from the robot's current position to the goal is found. Finally, the robot is navigated along the path. The whole exploration process is summarized in Algorithm 1.

Algorithm 1: The exploration algorithm

while *unexplored areas exist* **do**
    read current sensor information;
    update map with the obtained data;
    determine new goal candidates;
    determine the next goal;
    plan paths to determined goal;
    move the robot towards the goals;

3 Exploration framework

In this paper, we follow Yamauchi's frontier based approach [24], which is based on occupancy grids, i.e. the working space is divided into small cells, where each cell stores information about the corresponding piece of the environment in the form of a probabilistic estimate of its state. Moreover, it assumes that the next best view (goal) lies on the border between free and unexplored areas (this border is called *frontier*).

We present two strategies for goal-candidates determination. The first one (denoted "*all-frontiers*") is widely used; it takes all frontiers cells, while the second one ("*representatives*") is more sophisticated. It filters frontier cells in order to get a set of representatives approximating the frontier cells such that each frontier cell is detectable by the robot sensor from at least one representative. This is done by k-means clustering for each compact set each of frontier cells, where the representative of each cluster is the closest frontier cell to the cluster's mean. The number of clusters/representatives in a frontier is defined similarly to [4] as

$$n_f = 1 + \left\lfloor \frac{N_f}{1.8\rho} + 0.5 \right\rfloor ,$$

where N_f is the number of cells forming the frontier and ρ is the sensor range. This guaranties that all frontier cells will be explored (i.e. it will be detected whether the frontier contains the searched object or not) after visiting all representatives. Moreover, reduction of goal candidates dramatically decreases computational burden of more sophisticated and time-consuming goal-selection strategies and therefore allows their usage for non-trivial environments.

The key to effective exploration is selection of the most appropriate goal candidate as the next goal according to the defined criterion. The proposed method generates a trajectory from the current robot position to each goal-candidate (taking into account kinematics constrains of

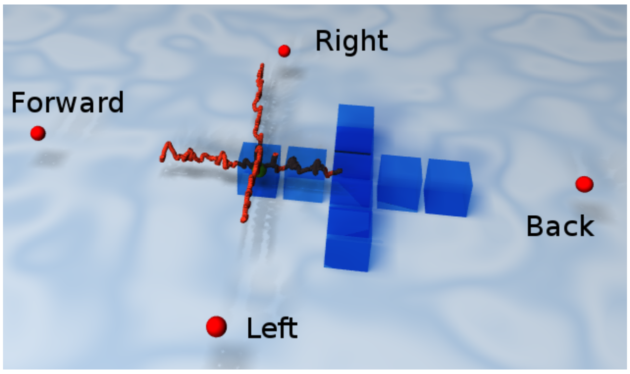


Figure 1: Four motion primitives for the Quadropod robot. The primitives are depicted as trajectory of the pivot module.

the robot) as described in the next section and selects the frontier to which the trajectory is shortest.

4 Planning for a modular robot

To move a modular robot towards a desired frontier, suitable locomotion of the robot needs to be generated. The problem of locomotion generation is well studied in the modular robotics. Usually, the Central Pattern Generators (CPG) [9] providing rhythmic control signals are utilized. By choosing suitable parameters of the CPGs, various locomotions like walking, crawling of even swimming can be achieved.

In the exploration task, the robot typically operates in a environment with obstacles, where more types of locomotion need to be utilized and a switching mechanism is required. In this paper, we utilize a motion planner equipped with the CPGs to find a feasible plan to reach the goal position.

While the motion planning can be solved using grid-based A* planner, Visibility graphs or Voronoi diagrams in the case of mobile robots [14], these methods cannot be used in the case of modular robots due to many DOFs. To solve the motion planning for a complex many-DOF system, sampling-based methods like Probabilistic Roadmaps [10] or Rapidly Exploring Random Trees [13] and their variants can be used. As these methods work in the configuration space, whose dimension equal to the number of DOF of the system, the sampling-based methods are suitable especially for systems with many DOFs. The methods has been utilized in many applications [8, 15, 5] including modular robotics [27, 22, 21].

To reduce the complexity of the planning problem for modular robots, one can employ a predefined set of motion primitives. Motion primitive provides a short-term control strategy for the robot, like 'go-forward' or 'turn-left'. In the case of modular robots, these motion primitives can be realized using the CPGs. In this paper, the motion planning problem for the modular robots is solved using the RRT-MP (RRT with Motion Primitives) [22] method.

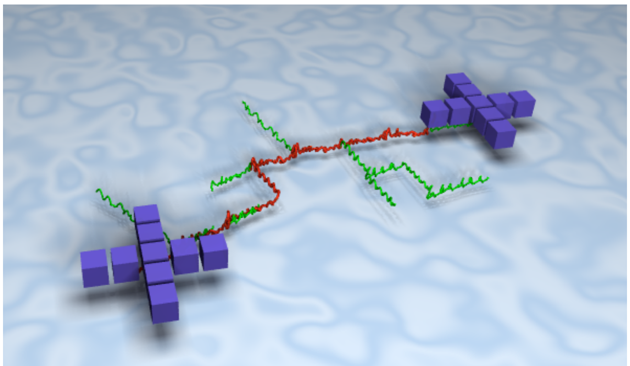


Figure 2: A tree generated for the Quadropod robot with the four primitives used.

Configuration of a modular robot with n modules is described as $q = (x, y, z, \alpha, \beta, \gamma, a_1, \ldots, a_{n-1})$, where (x, y, z) and (α, β, γ) denotes position and orientation of the pivot modules resp., and a_i are the angles between the connected modules. The set of all possible configuration is the configuration space $\mathcal{C}$ and the set of feasible configurations is the $\mathcal{C}_{free} \subseteq \mathcal{C}$. The robots is controlled by a control signal $u(t) = (a_1(t), \ldots, a_{n-1}(t))$, where $a_i(t)$ are desired angles of joints connecting two neighbor modules. In the RRT-MP approach, the control signals $u_i(t), i = 1, \ldots, m$ are modeled using m CPGs. The details about finding a suitable CPGs can be found in [22].

The RRT-MP planner iteratively builds a tree of feasible configurations rooted at the initial configuration q_{init}. In each iteration, a random configuration $q_{rand} \in \mathcal{C}$ is generated and a nearest node in the tree q_{near} is found. Then, the tree is expanded from the q_{near} using the motion primitives to obtain a set of new reachable configurations R. From this set, the nearest configuration towards q_{rand} is selected and added to the tree. The algorithm terminates, if the tree approaches the goal configuration q_{goal} to a predefined distance ϱ_g. The details of the RRT-MP as well as the preparation of the motion primitives can be found in [22]. The example of the four motion primitives for the Quadropod robot is depicted on Fig 1 and a tree generated with these motion primitives is on Fig. 2.

As the response of a modular robot to a control signal depends both on its kinematics as well as on the shape of underneath terrain, it is not easy to derive a closed-form motion model. Instead, the motion of the robot is simulated in a physical simulator. Therefore, to obtain a response of the robot to a control signal $u(t)$ defined by a motion primitive being examined, the simulated robot is placed to position q_{near} and controlled by the control signals $u(t), t = (0, T_{mp})$, where T_{mp} denote the duration of each motion primitive. The resulting configuration of the robot is then added to the set R if it is feasible and the robot does not hit any obstacle or moved over an unknown area during execution of the primitive.

For the purpose of exploration task, the motion planner needs to be extended. First, more desired goal position may be presented to the planner. Therefore, the RRT-MP

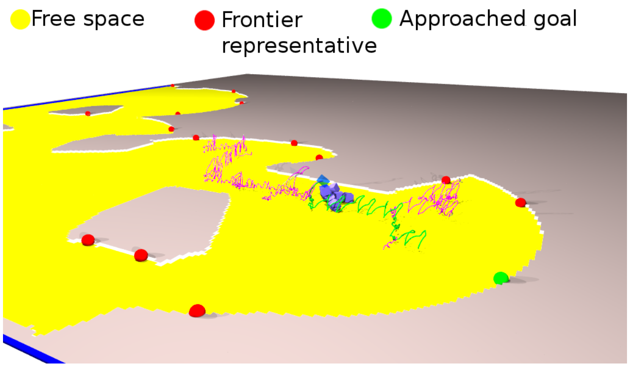


Figure 3: An example of a plan generated during the exploration. The plan is depicted in magenta and the final path towards a goal is in green.

planner is terminated when all goals are reached by the tree to a distance less than radius ϱ_g. To speed up the motion planner, the principle of the goal-bias [14] can be used. The random sample q_{rand} is set to q_{goal} with probability p_g, otherwise it is generated from the $\mathcal{C}$ with probability $1-p_g$. When more than one goal is presented to the planner, the probability of selecting q_{goal_i} instead of the q_{rand} is determined by its distance d_i to the tree:

$$p_{g_i} = \begin{cases} \frac{d_i}{\sum_{j=0}^{G} d_j} & d_i > \varrho_g \\ 0 & otherwise \end{cases} \qquad (1)$$

This ensures, that the closest goals are preferred until they are reached by the tree. This is important when the number of allowed iterations of the RRT-MP is limited, because this increases the probability, that a trajectory will be found to at least one goal.

5 Experiments

The performance of the above presented exploration framework has been experimentally verified in the Player/Stage framework [6]. All experiments were performed on the Intel P4@2.0 GHz with 4 GB of RAM under FreeBSD 8.2. To obtain the motion model of the Quadropod modular robot (depicted on Fig 1), ODE physical engine has been utilized [18] with 10 ms physical time step. The size of each module is 0.01 m and its weight is 1 kg. The robot was equipped with four motion primitives: 'go-x', where $x \in \{left, right, forward, back\}$). The duration of each primitive is $T_{mp} = 5$ s. The primitives are modeled using simple sinus-based CPG, where $a_i(t) = A_i \sin(\omega_i t + \varphi_i) + B_i$. We refer to [22] for details about searching of suitable parameters of these CPGs. The laser sensor is placed in the middle of the robot. To allow scanning of the environment parrallely to the ground, a 0.5 s decline phase is added to the end of each motion primitive. Therefore, the inputs $a_i(t) = 0$ for $t = (T_{mp} - 0.5, T_{mp})$.

The algorithms have been implemented in C++ as client programs for the Player/Stage in version 3.0.2 and compiled by the GCC 4.3.5 with -O2 optimization flag.

Map	Frontiers	Nearest	Trajectory	p-value	
Empty	represent.	212.9 (11.57)	189.6 (13.23)	8.10^{-6}	+
	all-frontiers	208.6 (13.34)	199.6 (15.63)	0.261	+
Small	represent.	46.39 (3.83)	43.45 (5.06)	0.0265	+
	all-frontiers	46.18 (4.79)	45.0 (4.61)	0.08	=

Table 1: Comparison of lengths of a path traversed during the exploration in form: mean-value (deviation). The column *p-value* is the p-value of the Wilcoxon test between mean values of the *trajectory* and *nearest* strategies. The last column highlights in the mean lengths of *trajectory* or *nearest* strategies is same (=) or if the *trajectory* approach provides significantly shorter paths (+) at 0.05 significance level.

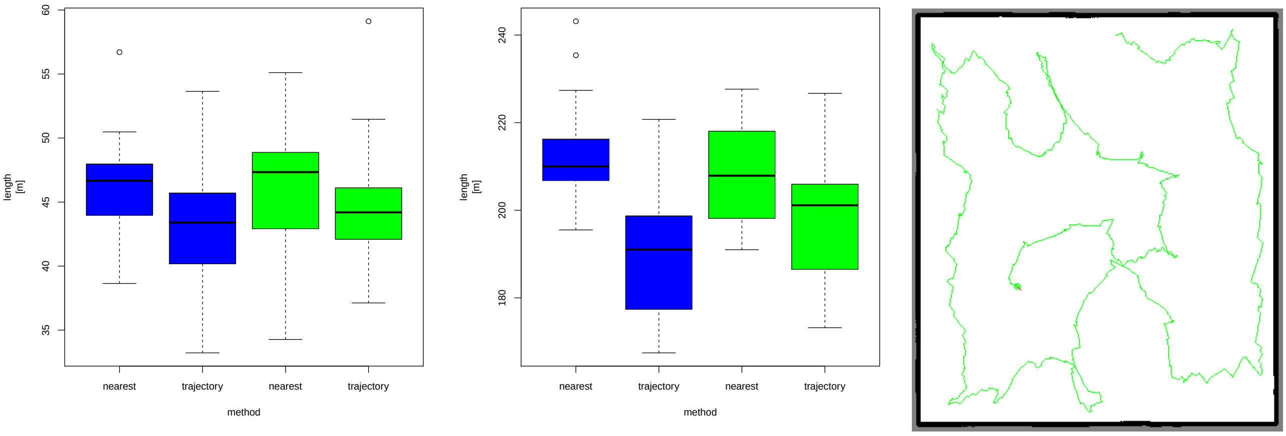
The experiments were conducted in two environments both representing an empty space without obstacles. The first environment (*small*) is scaled so it represents an area of 8×12 m, while the second one (*empty*) is scaled to 24×21 m. To gather information about the environment, a sensor with 360° field of view with 3 m range and 0.5° resolution was used, while the occupancy grid with cell size 0.05×0.05 m was chosen to represent the working environment.

As the current implementation of the motion planning algorithm is slow, the exploration iteration (i.e. the body of the loop in Algorithm 1) was run whenever the robot reached the previous goal, and the robot was stopped during computation of the new goal.

The experiments were run for both goal-candidates determination strategies presented in Section 3: *all-frontiers* and *representatives* and two goal selection approaches: the one presented in Section 3 *trajectory* and a standard strategy based on path planning on a graph and selecting the nearest goal-candidate to the current robot position (*nearest*). For each setup consisting of a pair ⟨goal-candidates determination, goal-selection⟩ twenty runs were performed. The results are depicted in Fig. 4 and in Table 1. They show that decreasing the number of goal candidates does not worsen exploration, on contrary it can give better results. Moreover, *trajectory* strategy outperforms standard *nearest* approach. For *empty* map and *representatives* it is more than 10%.

6 Conclusion

In this paper an autonomous exploration of an unknown environment by a modular robot is presented. The results show that more precise estimation of the distance cost based on trajectory length determined by motion planning taking kinematics constrains into account leads to better performance of exploration, i.e. paths needed to explore



(a) small (b) empty (c) best solution

Figure 4: Statistics for (a) *small* and (b) *empty* environment. *representatives* goal-candidates determination strategy is in blue, *all-frontiers* in green. (c) The best solution found for the *empty* environment.

		Nearest	
		Representation	All-frontiers
Trajectory	Representation	+/+	+/=
	All-frontiers	+/=	=/=

Table 2: Comparison of all methods on empty/small maps. Here, '+' denotes, that the method in column provides longer path than the method in the row at 0.05 significance level; '=' indicates, that mean values of the path lengths are same.

the whole environment are significantly shorter than for standard approaches.

Probabilistic nature of the standard RRT makes it inappropriate for fast replanning as trajectories generated in two consecutive exploration iterations are typically different, which leads to undesirable oscillations of robot movement. Therefore, to use the presented approach for frequency-based decisions [20] (i.e. exploration iterations are run at some fixed frequency, without need to reach the current goal), RRT have to be derandomized. This can be for example done by using a pruned tree grown in the previous iteration for the current iteration.

The proposed motion planning algorithm is slow so it is not feasible to run the exploration real-time. In the future work, the planning methods will be improved to provide real-time planning. The number of goal-candidates will be decreased by preselecting most promising ones, which should also speed-up the planning process. Of course, experiments in more complex environments and with real robots are necessary to confirm the presented results.

Acknowledgments

This work has been supported by the Technology Agency of the Czech Republic under the project no. TE01020197 "Centre for Applied Cybernetics".

References

[1] Francesco Amigoni and Vincenzo Caglioti. An information-based exploration strategy for environment mapping with mobile robots. *Robotics and Autonomous Systems*, 58(5):684 – 699, 2010.

[2] N. Basilico and F. Amigoni. Exploration strategies based on multi-criteria decision making for an autonomous mobile robot. In *Proc. of the Fourth European Conference on Mobile Robots*, pages 259–264. KoREMA, 2009.

[3] I. Erkmen, A.M. Erkmen, F. Matsuno, R. Chatterjee, and T. Kamegawa. Snake robots to the rescue! *Robotics Automation Magazine, IEEE*, 9(3):17 – 25, sep 2002.

[4] Jan Faigl, Miroslav Kulich, and Libor Preucil. Goal assignment using distance cost in multi-robot exploration. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3741 –3746, oct. 2012.

[5] R. Gayle, S. Redon, A. Sud, M.C. Lin, and D. Manocha. Efficient motion planning of highly articulated chains using physics-based sampling. In *ICRA 2007*, 2007.

[6] Brian P. Gerkey, Richard T. Vaughan, and Andrew Howard. The Player/Stage project: Tools for multi-robot and distributed sensor systems. In *Proc. of the 11th Int. Conf. on Advanced Robotics*, pages 317–323, 2003.

[7] Hector H. Gonzalez-Banos and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, Oct. 2002.

[8] L.J. Guibas, C. Holleman, and L.E. Kavraki. A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach. In *IROS 1999*, 1999.

[9] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: A review.

Neural Networks, 21(4):642 – 653, 2008.

[10] Lydia E. Kavraki, Petr Svestka, Jean claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12:566–580, 1996.

[11] Miroslav Kulich, Jan Faigl, and Libor Preucil. On distance utility in the exploration task. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4455–4460, 2011.

[12] H.Y.K. Lau, A. Ko, and T.L. Lau. A decentralized control framework for modular robots. In *IROS 2004*, 2004.

[13] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning, 1998. TR 98-11.

[14] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[15] S.M. LaValle, J.H. Yakey, and L.E. Kavraki. A probabilistic roadmap approach for systems with closed kinematic chains. In *ICRA 1999*, 1999.

[16] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte. An experiment in integrated exploration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System*, pages 534–539. IEEE, 2002.

[17] Paul Moubarak and Pinhas Ben-Tzvi. Modular and reconfigurable mobile robotics. *Robotics and Autonomous Systems*, 60(12):1648 – 1663, 2012.

[18] Open source library. ODE — Open Dynamics Engine, available at. http://www.ode.org/.

[19] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Proc. of Robotics: Science and Systems*, Cambridge, MA, USA, 2005.

[20] Aris Valtazanos and Subramanian Ramamoorthy. Evaluating the effects of limited perception on interactive decisions in mixed robotic domains. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, HRI '13, pages 9–16, Piscataway, NJ, USA, 2013. IEEE Press.

[21] V. Vonasek, K. Kosnar, and L. Preucil. Motion planning of self-reconfigurable modular robots using rapidly exploring random trees. In *Lecture Notes in Computer Science series, TAROS 2012*, volume 7429, 2012.

[22] V. Vonasek, M. Saska, K. Kosnar, and L. Preucil. Global motion planning for modular robots with local motion primitives. In *International conference of robotics and automation (ICRA)*, Karlsruhe, Germany, 2013.

[23] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151. IEEE Comput. Soc. Press, 1997.

[24] Brian Yamauchi. Frontier-based exploration using multiple robots. In *Proc. of the Second International Conference on Autonomous Agents*, pages 47–53, 1998.

[25] M. Yim, D. G. Duff, and K. Roufas. Modular reconfigurable robots, an approach to urban search and rescue. In *1st International Workshop on Human-friendly Welfare Robotics Systems*, January 2000.

[26] Mark Yim, Kimon Roufas, David Duff, Ying Zhang, Craig Eldershaw, and Sam Homans. Modular reconfigurable robots in space applications. *Auton. Robots*, 14(2-3):225–237, March 2003.

[27] E. Yoshida, H. Kurokawa, A. Kamimura, K. Tomita, S. Kokaji, and S. Murata. Planning behaviors of a modular robot: an approach applying a randomized planner to coherent structure. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 2056 – 2061 vol.2, sept.-2 oct. 2004.