Speeded Up Elevation Map for Exploration of Large-Scale Subterranean Environments

Jan Bayer and Jan Faigl[0000−0002−6193−0792]

Faculty of Electrical Engineering, Czech Technical University in Prague,
Technicka 2, 166 27, Prague, Czech Republic
{bayerja1,faiglj}@fel.cvut.cz
https://comrob.fel.cvut.cz

Abstract. In this paper, we address a problem of the exploration of large-scale subterranean environments using autonomous ground mobile robots. In particular, we focus on an efficient data representation of the large-scale elevation map, where it is desirable to capture the shape of the terrain to avoid areas not traversable by a robot. Subterranean environments such as mine tunnel systems can be in units of kilometers large, but only a relatively small portion of the environment represents observable parts. Therefore, uniform grid-based elevation maps with resolution in units of centimeters are not memory efficient, and more suitable are hierarchical tree-based structures. However, hierarchical structures suffer from the increased computational requirements of accessing particular grid cells needed in determination of the navigational goals or evaluation of the terrain traversability in planning safe and cost-efficient paths. We propose a speed-up technique to combine the benefits of uniform grid-based and tree-based representations. The proposed elevation map representation keeps the memory footprint low using tree structure but enables fast access to the grid cells corresponding to the robot surroundings. The efficiency of the proposed data representation is demonstrated in an experimental deployment of the autonomous exploration of outdoor and subterranean environments.

1 Introduction

Mobile robots can be deployed in hard to access and dangerous areas to reduce possible risks for humans, specifically in search and rescue missions in underground tunnels or collapsed buildings [17]. A human operator can teleoperate robots, but a small communication range in underground tunnels limits operational radius. Therefore, autonomous exploration [27] is needed to search large underground environments such as in the DARPA Subterranean Challenge [5], where robots are requested to search for the artifacts such as survivors, extinguishers, drills machine, and cell phone, to name few.

Since an underground environment like mines and caves forms a net of corridors that can be very large (in kilometers), it is desirable to use a memory-efficient map representation to cover the whole environment with sufficient resolution for traversability assessment. On the other hand, it is also desirable to

keep the average access time to the map representation as low as possible because frequent access to the map is common for exploration algorithms. An example of exploration processes that access the environment map includes

- the insertion of the new measurements;
- traversability assessment;
- growing untraversable areas by the radius of the robot shape circumference;
- computation of the cost map;
- detection of possible goal locations;
- and path planning.

Therefore, efficient map representation can have a significant impact not only on the memory requirements but also on the computational requirements and the latency between the integration of new sensor measurements and a new decision.

Existing terrain models capable of covering large environments are based on a tree structure to save memory requirements such as OctoMap [12]. However, access time to the leaves of tree-like representations is slower than the access time to the uniform grid due to the necessity to search the tree for the leaves. For relatively large robots like Talon [19], or ClearPath Husky with sufficient payload capacity, it is possible to overcome the access time by increasing computational power and the required energy resources. However, for small robots like Micro Tactical Ground Robot (MTGR) [18] or walking hexapod robots [7] with limited payload, the computational resources are limited to small embedded computers. Therefore, computational efficiency might play a significant role in the operational time. Thus, we studied memory-efficient terrain model representation to capture large areas with sufficient precision for traversability assessment that also enables fast access to the terrain representation. The main contribution of the presented approach is considered in memory and computationally efficient data representation of the elevation map suitable for mobile robot exploration. The proposed solution is demonstrated on autonomous exploration with a small hexapod walking robot equipped with the relatively limited computational power of the onboard embedded computer. Besides, we report on experimental results from a practical deployment of the developed solution in the mine environment.

The rest of the paper is organized as follows. A brief overview of the related work is presented in Section 2. The developed map representation and the exploration framework are described in Section 3. The achieved results, including map captured during the exploration and evaluation of the computational requirements, are reported in Section 4. The concluding remarks and ideas for future work are in Section 5.

2 Related Work

The robotic exploration is a problem to create a model of the environment by a mobile robot. The frontier-based exploration introduced in [26] is a well-known approach for spatial robot exploration widely adapted by existing exploration approaches [22,1]. In frontier-based exploration, robots are navigated towards

waypoints determined at the borders of known and unknown parts of the environment [11]. Besides, the navigation waypoints can also be determined using entropy [4], or the exploration can be driven by decreasing uncertainty of the terrain model being created, e.g., as reported in [21]. Nevertheless, the common property of the exploration approaches is frequent access to the data representation of the model being created [8]. Since, we are focused on the memory-efficient model representation, we consider relatively straightforward deployment of the frontier-based exploration [2] to demonstrate utilization of the model and impact of different memory representations.

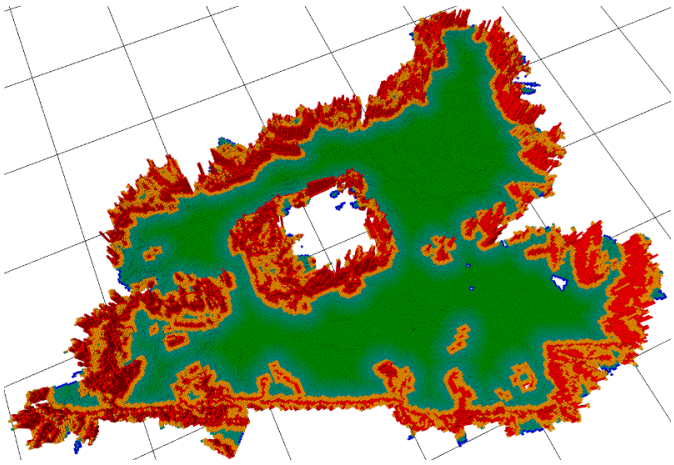


Fig. 1. Example of the elevation map with the resolution 7.5 cm. The grid at the backround has cell size 5 m.

In robotic exploration, the traditional terrain model representation is the occupancy grid [6] that covers the space by a uniform grid, where each cell is associated with the probability that the space represented by the cell is occupied. The occupancy grid is a suitable model for flat terrains with easily distinguishable obstacles that are represented as cells with a high value of the occupancy probability. However, a subterranean environment is often not flat, since the shape of the terrain is usually uneven. The extension of the occupancy grid to full 3D representation is presented in [24], where the authors represent the map by a uniform 3D grid. The disadvantage of [24] is the memory ineffectiveness, which is addressed by tree-like structured map representation known as the OctoMap [12]. Since the OctoMap is based on the octree, the complexity of the access to the tree leaves can be bounded by $O(\log(n))$, where n is the number of tree nodes. Thus the access to the octree is significantly slower than the access time to the terrain model represented by the uniform grid, which has the constant access time $O(1)$.

If we assume only a single layer of the terrain, the elevation map [20] can be employed to model the shape of the terrain by estimating the height of the ter-

rain at each cell of the grid. The advantage of the elevation map is its simplicity because cell heights can be stored in the 2D grid, which is more memory efficient than the 3D grid [24]. Due to the simpler structure [9], the access is also faster than for 3D maps stored in tree structures. Besides the uniform 2D grid, e.g., used in [20], the elevation map can also be stored in a tree structure (quadtree) similarly to the occupancy mapping technique described in [15]. Then, the elevation map becomes even more memory-efficient representation than 3D grids at the cost of the access time because of the tree-like structure.

The presented work is motivated by an outdoor exploration scenario with a small hexapod walking robot and a wheeled robot (see Figure 3) in the Tunnel Circuit event of the DARPA Subterranean Challenge [25]. The selected deployment scenarios with long tunnels allow us to assume only a single terrain layer. Hence, the terrain model is based on the memory-efficient elevation map stored as a quadtree. The fast access to the tree-like structure is supported by the proposed caching mechanism that significantly reduces the access time. The developed memory representation is demonstrated within a complete exploration framework, where particular processes access to the terrain model. Therefore, the reported experimental results represent realistic computational requirements of the practical deployments in exploration missions.

3 Autonomous Exploration Framework

The proposed terrain model representation is integrated into the autonomous exploration system that consists of five main modules: *sensors and localization*, *mapping*, *exploration*, *path following*, and robot *controller*. The modules and their connections into the system architecture are visualized in Figure 2. The exploration framework is developed in C++ using the ROS middleware [23]. The utilized *sensors* provide range measurements, and the *localization system*

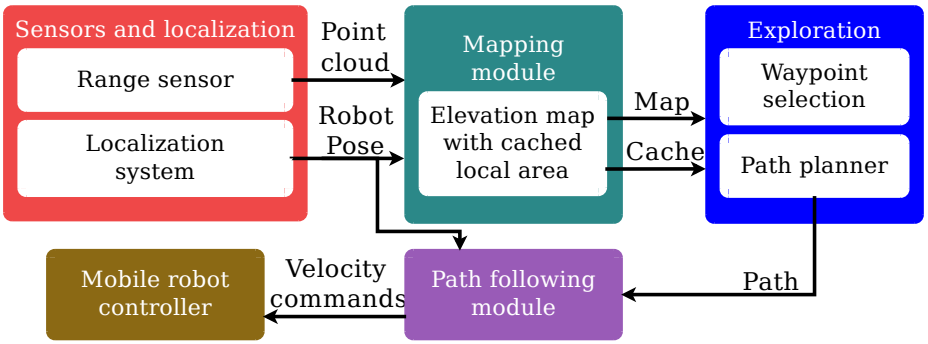


Fig. 2. Architecture of the developed autonomous exploration system.

provides a pose of the robot that are both used by the *mapping* module to build

the spatial model of the environment. The map is further used to determine possible goal locations to explore unknown parts of the environment. The next navigational goal is determined from the possible goal locations using the employed exploration strategy. The *path following* module ensures the robot follows the path planned to the selected navigational goal by steering the robot by the velocity. The individual modules are further described in the rest of this section to provide a complete overview of the utilized autonomous exploration system, and the proposed map representation is detailed in Section 3.1.

Sensors depend on the particular sensory system of the robot, but in this paper, we utilize two robotic platforms showed in Figure 3 for the experimental verification of the proposed exploration system. The first is the hexapod walking robot that uses the RGB-D camera Intel RealSense D435 [13] as the source of range measurements. The localization of the hexapod walking robot is provided by the small embedded localization module, the Intel RealSense T265 [14], that runs onboard visual SLAM from the fisheye stereo camera enhanced by sensory fusion with the inertial measurement unit. The wheeled robot can carry a more massive payload, and therefore, a laser range finder is used to obtain 3D scans of the robot environment. The localization of the wheeled robot is based on the Iterative Closest Point algorithm [3] combined with odometry.

(a) Hexapod walking robot (b) Wheeled robot

Fig. 3. Robotic platforms used for the experimental verification of the developed autonomous exploration system.

Mapping module builds the elevation map from the captured point clouds synchronized with the estimated robot pose. Although the representation of the proposed map is a tree-like structure, we describe the map as the uniformly sampled grid to improve the clarity and readability of the processes dealing with the map. Thus, each map sample (i, j) represents a height h and height variance σ_h^2 of the corresponding terrain place. From incoming point clouds, points that

do not belong to the top modeled surface of the terrain are filtered out. The filtered point cloud is then fused with the elevation map using one dimensional Kalman filter [9].

The measurements of the terrain height z_k and its variance $\sigma^2_{z,k}$ from the new k-th input point cloud are fused with the corresponding heights with the map according to the equations

$$h_k = \frac{\sigma^2_{z,k} h_{k-1} + \sigma^2_{h,k-1} z_k}{\sigma^2_{z,k} + \sigma^2_{h,k-1}} \tag{1}$$

$$\sigma^2_{h,k} = \frac{\sigma^2_{z,k} \sigma^2_{h,k-1}}{\sigma^2_{z,k} + \sigma^2_{h,k-1}}. \tag{2}$$

Parts of the map which were not affected by the new measurements are considered *unknown*.

The proposed elevation map is further used for the traversability assessment based on the height difference of the neighboring cells as follows. If the local height difference $g_h(i,j)$ defined by (3) is higher than the threshold g_{max}, the cell is considered untraversable; otherwise, the cell is traversable. The value of g_{max} is estimated based on the kinematics of the particular robot. Since the planner plans the path for the center of the robot body, the physical dimensions of the robot are incorporated to the map by growing the untraversable cells by the radius of the robot shape circumference. Moreover, we generate the cost map to penalize robot states close to the untraversable cells by utilizing the distance transform [10]. The cost $d(i,j)$ of each cell is thus based on its distance to the closest untraversable cell, or it is labeled as unknown if measurement about the corresponding area is not available.

$$\begin{aligned} g_h(i,j) = \max(\{&|h(i,j) - h(i-1,j)|, \\ &|h(i,j) - h(i+1,j)|, \\ &|h(i,j) - h(i,j-1)|, \\ &|h(i,j) - h(i,j+1)|\}) \end{aligned} \tag{3}$$

Exploration module determines the next navigational goal based on the identified frontiers [26]. The frontier cells are determined in the elevation map as traversable cells that are incident with unknown cells. The total number of frontier cells depends on the resolution of the map. In general, for a high-resolution map, the number of frontier cells can be large, and it would be computationally very intensive to select possible goal locations from all the frontier cells. Thus, we followed the approach [16] and employed the clustering of the frontier cells. The nearby frontier cells are clustered into similarly sized sets, and a single representative of each set (cluster) is determined as the possible goal location.

The next navigational goal is selected as the goal location with the lowest cost of the path from the current robot pose. The path is determined by the A* algorithm with the heuristic function computed as the Euclidean distance to the goal location. The travel cost between two neighboring nodes n and n' is

computed as the distance on the eight-neighborhood (D_8) that is increased by the cost $d(n)$ to penalize robot presence near the obstacles. $d(n)$ is non-negative cost decreasing with D_8 distance from the closest untraversable cell [10]. The selection of the next navigational goal is focused on the nearby area of the robot, i.e., we consider a limited planning horizon. Therefore, a possible goal location is considered unreachable if a path between the robot pose and the goal location is not found in less than 20 000 expansions of the A* algorithm.

The robot is then navigated along the path to the selected goal location by the *path following* module. A new goal location is determined if the current waypoint is reached or after $T_{exp} = 8\,\mathrm{s}$. The exploration terminates if there is not a reachable goal location.

Path following module ensures that the robot follows the path planned by the *exploration* module. The path is represented as a sequence of waypoint locations that are progressively processed. The forward and angular velocities for the robot controller are generated based on the distance and angular displacement of the recent waypoint of the path. The next waypoint from the sequence is processed if the robot gets less than 1 cm far from the recent waypoint.

3.1 Map representation

Since we suppose exploration of the environment with tunnels, we represent the map by a **quadtree** structure, see Figure 4, to reduce the memory requirements. The disadvantage of the tree-like representation is the access time to the tree leafs that can be bounded by $O(\log n)$, where n denotes the number of the map cells that depends on the size of the map, which is slower than the access time for the map represented by a uniform grid that can be bounded by $O(1)$. Complex access to the tree-like representation increase computational requirements, and for limited computational power, it slows down map processing. It is specifically distinctive when the elevation map is processed by multiple methods, which happens in the exploration mission during the integration of new measurements, obstacle growing, determination of frontier cells and goal locations, and also during path planning.

We propose to overcome the issue of the high complexity of repetitive access to the map by an additional data structure that works as a **cache**. We suppose that new measurements do not affect all the cells of the elevation map, but only cells that are close to the robot since the range of the sensors is limited. Hence, the proposed cache stores the references to all the quadtree leafs close to the robot. In Figure 4, it can be seen that the cached area overlaps the area affected by measurements, which ensures the consistency of the area changed by the measurements with the rest of the map. The quadtree map representation enables us to represent the unknown space efficiently, while still, the operations like convolution can run almost as quickly as on the map stored in a uniform grid. The cost of maintaining the cache is its computation, which is needed when the new point cloud updates the map. The real impact of the proposed

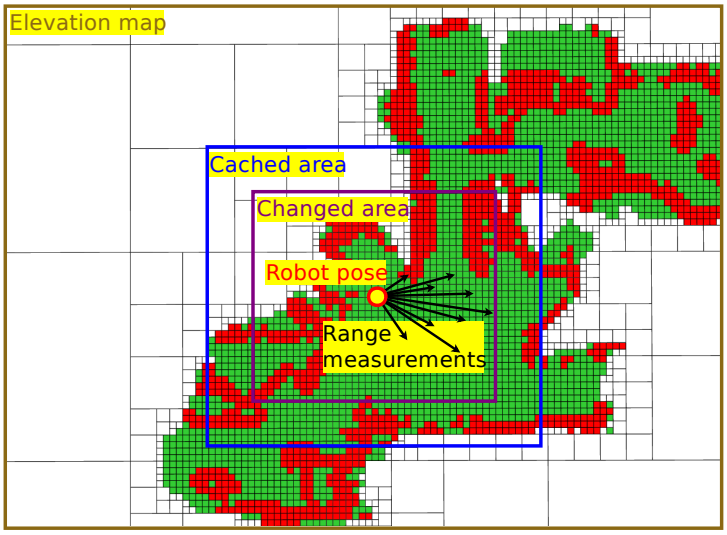


Fig. 4. Elevation map representation.

hybrid representation has been evaluated during real exploration missions, and the achieved results are reported in the following section.

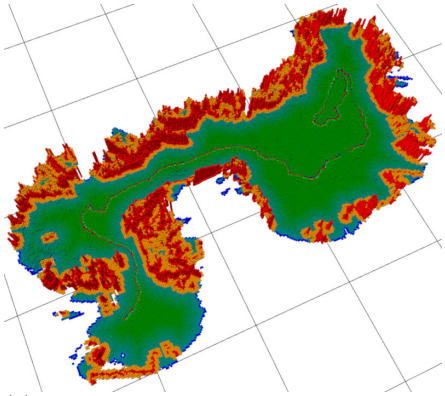
4 Results

We have designed two exploration experiments to show the speedup of the mapping due to the proposed cache, and demonstrate how the proposed memory-efficient solution represents a large-scale subterranean environment. The first experiment has been done using the small hexapod walking robot employed in the exploration of the outdoor rough grassy terrain. The second experiment reported is a deployment of the autonomous exploration in one trial during the Tunnel Circuit event of the DARPA Subterranean Challenge in August 2019, where the wheeled robot autonomously explored the entry part of the underground tunnel.

4.1 Exploration with Small Hexapod Walking Robot

During the experimental deployment of the exploration framework with the small hexapod walking robot in the scenario shown in Figure 5, we have captured data from the onboard sensors using the ROS rosbag tool. The captured sensor data can be played back at the same frequencies as they were generated by the sensors, which enables us to benchmark mapping performance with and without the proposed cache using different computational environments. In particular, the evaluation of the mapping processes has been performed with two computers. The first computer is equipped with the Intel i5 3320M processor clocked at 2.6 GHz and 8 GB RAM. The second computer is small embedded computer

(a) Outdoor experimental setup



(b) Resulting elevation map with resolution 7.5 cm with the visualization grid of the size 5 m at the background

Fig. 5. Exploration experiment with the hexapod walking robot used as a benchmark for mapping process using different memory representations of the map.

Odroid-XU4 with Octa-core CPU Samsung Exynos 5422 (four A15 cores running at 2.0 GHz and four A7 cores running at 1.4 GHz), and 2 GB memory. Both computers have sufficient computational power for autonomous exploration with the hexapod walking robot. The computational times for the particular mapping steps (including insertion of new sensor measurements) are summarized in Table 1 and Table 2. The input point clouds contain more than 60 000 measurements, and the depth camera produced the point clouds at the frequency 5 Hz. The robot traversed 46 m in 44 minutes during the experimental deployment.

Table 1. The mean computational time for mapping steps

Process name	Intel i5 CPU		Odroid-XU4	
	No cache	With cache	No cache	With cache
Cache update	0.0	5.2	0.0	9.9
Traversability assessment	17.5	1.8	30.5	3.2
Growing untraversable areas	19.1	1.8	32.4	4.7
Cost map building	17.4	5.8	34.6	22.3
Frontier detection and clustering	4.8	0.4	8.6	0.9
Total mapping time	76.1	32.0	146.4	76.7

All reported times are in milliseconds.

Table 2. Utilization of the computational resources

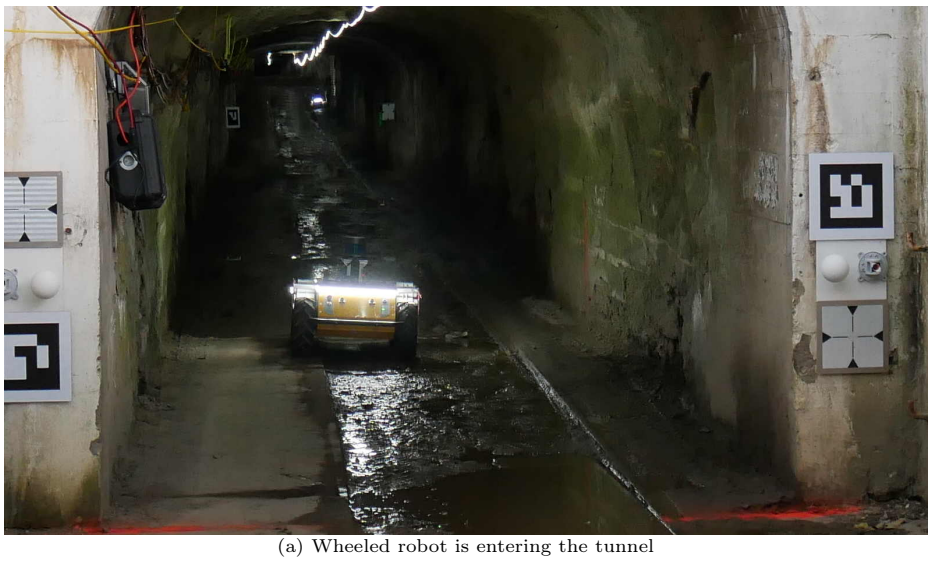
Mapping	Intel i5 CPU [%]	Odroid-XU4 [%]
Mapping without the cache	38	92
Mapping with the cache	18	52

Although the results indicate that both computational platforms are sufficient for the exploration even without the proposed cache, the cache reduces the

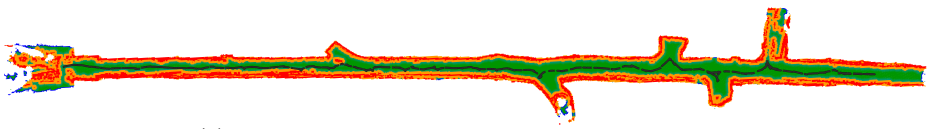
computational requirements about two times, which might further support the deployment of more sophisticated exploration strategies. Note that the power consumption of the computational environment with the Odroid-XU4 is about two times less than for the setup with the Intel i5.

4.2 Exploration of Subterranean Environment

The second experimental deployment is based on the exploration of the 130 m long mine entrance during the Tunnel Circuit event of the DARPA Subterranean Challenge, see Figure 6. We have measured the size of the created elevation map, and compare it to the size of the uniform grid map needed to cover the same area. The resulting memory requirements summarized in Table 3 support that the developed memory representation of the terrain map is memory efficient. Even though the memory requirements per each cell are larger for the proposed representation (because of the quadtree structure) than for the uniform grid, the results show that the memory footprint of the proposed map representation with the uniform grid-like cache is still lower than for the uniform grid only.



(a) Wheeled robot is entering the tunnel



(b) Resulting elevation map with market robot trajectory

Fig. 6. Wheeled robot in the tunnel of the Safety research course during the Tunnel Circuit event of the DARPA Subterranean Challenge.

Note that the results in Table 3 are calculated for the 130 m long entrance to the mine, the total length of the tunnels in the mine is several kilometers.

Table 3. Memory requirements of the map representation

Map representation	Map cell size [B]	Total map size [MB]
Grid based representation	9	58.0
Proposed quadtree representation	64	4.8

5 Conclusion

In this paper, we report on the experimental results of the developed autonomous exploration system with the proposed memory representation of the elevation map. Although the proposed hybrid memory representation based on the quadtree tree with a uniform grid cache is straightforward and relatively easy to implement, it provides noticeable benefits in the reduced memory and computational resources. The presented results support the memory and computational efficiency of the developed solution and enable autonomous exploration using a relatively small robotic platform with limited computational resources. The exploration system has also been successfully deployed in the Tunnel Circuit of the DARPA Subterranean Challenge. In the future, we aim to generalize the idea of the memory representation for the full 3D map of the explored environment.

Acknowledgement

The presented work has been supported under the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics". The support under grant No. SGS19/176/OHK3/3T/13 to Jan Bayer is also gratefully acknowledged.

References

1. Amigoni, F., Banfi, J., Basilico, N.: Multirobot exploration of communication-restricted environments: A survey. IEEE Intelligent Systems **32**(6), 48–57 (2017). DOI: 10.1109/MIS.2017.4531226
2. 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) (2019)
3. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(2), 239–256 (1992). DOI: 10.1109/34.121791
4. Carrillo, H., Dames, P., Kumar, V., Castellanos, J.A.: Autonomous robotic exploration using occupancy grid maps and graph slam based on shannon and rényi entropy. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 487–494 (2015). DOI: 10.1109/ICRA.2015.7139224
5. Chung, T.: DARPA Subterranean (SubT) Challenge. https://www.darpa.mil/program/darpa-subterranean-challenge (accessed July 12, 2019)
6. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer **22**(6), 46–57 (1989)

7. 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). DOI: 10.1016/j.robot.2019.03.008
8. Faigl, J., Kulich, M.: On benchmarking of frontier-based multi-robot exploration strategies. In: European Conference on Mobile Robots (ECMR). pp. 1–8 (2015). DOI: 10.1109/ECMR.2015.7324183
9. Fankhauser, P., Bloesch, M., Hutter, M.: Probabilistic terrain mapping for mobile robots with uncertain localization. IEEE Robotics and Automation Letters **3**(4), 3019–3026 (2018). DOI: 10.1109/LRA.2018.2849506
10. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. Theory of Computing **8**, 415–428 (2012)
11. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robotics: Science and Systems (RSS) **61**(12), 1258–1276 (2013). DOI: 10.1016/j.robot.2013.09.004
12. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots **34**(3), 189–206 (2013). DOI: 10.1007/s10514-012-9321-0
13. Intel RealSense Depth Camera D435. https://click.intel.com/intelr-realsensetm-depth-camera-d435.html, accessed August 4, 2018
14. Intel RealSense Tracking Camera T265. https://click.intel.com/order-intel-realsense-tracking-camera-t265.html, accessed May 23, 2019
15. Kraetzschmar, G.K., Gassull, G.P., Uhl, K.: Probabilistic quadtrees for variable-resolution mapping of large environments. IFAC Proceedings Volumes **37**(8), 675–680 (2004). DOI: 10.1016/S1474-6670(17)32056-6
16. 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). DOI: 10.1109/ICRA.2011.5980221
17. Liu, Y., Nejat, G.: Robotic Urban Search and Rescue: A Survey from the Control Perspective. Journal of Intelligent & Robotic Systems **72**(2), 147–165 (2013). DOI: 10.1007/s10846-013-9822-x
18. Micro tactical ground robot. http://www.robo-team.com/products/mtgr/ (accessed July 11, 2019)
19. Peter Wells, D.D.: Talon: a universal unmanned ground vehicle platform, enabling the mission to be the focus. In: SPIE. vol. 5804 (2005). DOI: 10.1117/12.602887
20. Pfaff, P., Triebel, R., Burgard, W.: An efficient extension to elevation maps for outdoor terrain mapping and loop closing. International Journal of Robotics Research **26**, 217–230 (2007). DOI: 10.1177/0278364906075165
21. Prágr, M., Čížek, P., Bayer, J., Faigl, J.: Online incremental learning of the terrain traversal cost in autonomous exploration. In: Robotics: Science and Systems (RSS) (2019). DOI: 10.15607/RSS.2019.XV.040
22. Quattrini Li, A., Cipolleschi, R., Giusto, M., Amigoni, F.: A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings. Autonomous Robots **40**(4), 581–597 (2015). DOI: 10.1007/s10514-015-9480-x
23. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
24. Roth-Tabak, Y., Jain, R.: Building an environment model using depth information. Computer **22**(6), 85–90 (1989). DOI: 10.1109/2.30724
25. Subterranean challenge. https://www.subtchallenge.com/ (accessed July 12, 2019)

26. 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). DOI: 10.1109/CIRA.1997.613851
27. Yanguas-Rojas, D., Mojica-Nava, E.: Exploration with Heterogeneous Robots Networks for Search and Rescue. IFAC-PapersOnLine **50**(1), 7935–7940 (2017). DOI: 10.1016/j.ifacol.2017.08.768