DCEGen: Dense Clutter Environment Generation Tool for Autonomous 3D Exploration and Coverage Algorithms Testing

Evgeni Denisov[1], Artur Sagitov[1], Roman Lavrenov[1](✉), Kuo-Lan Su[2], Mikhail Svinin[3], and Evgeni Magid[1]

[1] Higher Institute for Information Technology and Intelligent Systems (ITIS), Kazan Federal University, 35 Kremlyovskaya street, Kazan, Russian Federation
evdenisov@stud.kpfu.ru, {sagitov,lavrenov,magid}@it.kfu.com

[2] Department of Electrical Engineering, National Yunlin University of Science and Technology, Tainan, Taiwan
sukl@yuntech.edu.tw

[3] Robot Dynamics and Control Laboratory, College of Information Science and Engineering, Ritsumeikan University, Noji Higashi 1-1-1, Kusatsu 525-8577, Japan
svinin@fc.ritsumei.ac.jp

Abstract. Autonomous exploration and coverage in 3D environments recently has became a rapidly developing research field. Emerging 3D reconstruction methods, designed specifically for exploration and coverage, allows capturing an environment in a greater details. However, not much work addresses certain difficulties inherent to dense clutter environments. We observed those difficulties and made an attempt that seeks to expand the applicability of such methods to more demanding scenarios. Automating the process of testing and evaluation by designing a dense clutter environment generation algorithm (DCEGen) allows us to measure comparative performance of available algorithms. We focus on path-planning algorithms used in an unmanned ground vehicles. The algorithm was implemented and verified using Gazebo simulator.

Keywords: Mobile robot · Gazebo simulation · ROS · Dense clutter environment · 3D environment reconstruction · Autonomous exploration and coverage algorithm · Next-best-view

1 Introduction

Modern 3D reconstruction systems have developed methods that are able to produce highly-detailed and largely accurate reconstructions of real environments using online processing even with monocular cameras [8]. 3D Reconstruction systems have found their applications in autonomous navigation of mobile

A. Ronzhin et al. (Eds.): ICR 2019, LNAI 11659, pp. 216–225, 2019.
https://doi.org/10.1007/978-3-030-26118-4_21

robots [19], 3D scanning [25] and augmented reality [27], that include consumer software. However, some issues have not been yet resolved, e.g. operating in various dynamic environments, small range of sensing, operation in a featureless monotone environments, presence of reflective surfaces, dynamic lighting conditions, etc. Efforts that amend those issues lead to the fully autonomous 3D scanning and possibility learning and reasoning about of 3D space. Complex environments are especially important for evaluation of autonomous 3D exploration algorithms that are used in urban search and rescue robotics (USAR), which is often constrained to non-typical environments, such as tunnels, caves, forests, mountains, mines, construction sites, collapsed buildings, junkyards, etc [16]. Even for applications within collaborative robotics [21], the task of finding objects with known geometry specified by human in a house environments represents an exploration and coverage task [9], where the robot indicates successful discovery. Motivation for this work is the lack of general evaluation methodology for existing algorithms in dense cluttered environments. Majority of development and testing work for 3D exploration and coverage planning algorithms are undertaken in well-controlled environments, majority of researchers test their algorithms in their laboratory rooms that have simple underlying 3D structure [4,15,20]. In general, behavior of an algorithm is not validated against a complex geometry environment. Creating a dense clutter environment for real experiments is a challenging task, especially if we are required to have fragile objects that are coming out from walls or hanging down from a ceiling [14,26]. We propose an algorithm for testing environment generation for evaluation of existing and our own exploration and coverage planning algorithms.

In this paper, we present the first version of DCEGen - an environment generation algorithm that produces random dense clutter unstructured environments as a 3D model with information of possibly visible voxels from a given robot configuration. DCEGen is designed to aid development of new algorithms for tasks of autonomous exploration and coverage in dense clutter 3D environments. DCEGen was developed using Python programming language and is targeting for usage within Robot Operating System (ROS) framework [3].

The rest of this paper is organized as follows. Section 2 overviews related work and highlights limitations of existing solutions. Section 3 describes problem definition, system setup and our proposed solution approach. Finally, we conclude in Sect. 4 and discuss our future work plans.

2 Related Work

The problem of autonomous exploration and coverage in 3D space appeared fairly recently as a scientific research, but there already significant progress has been made. This research area have started and still mostly concentrate on algorithms for UAVs, as UAVs usually have limited teleoperation control and low capacity batteries that require more automation in area coverage jobs. Exploration and coverage path-planning problem for 3D and 2D (see [10]) spaces have similar goal of developing a globally optimal solution that provides total coverage of all visible environment in free configuration space without a-priori knowledge.

One of the first research groups who have successfully combined exploration and coverage goals for 3D space were [11]. They indicated that previous research on exploration were ignorant of 3D space, works on coverage were based on a-priori knowledge of the entire environment map, while next-best-view algorithms assumed only single objects of known size (further discussed in [11]). They presented an information gain-based heuristic solution for unmanned aerial vehicles (UAVs), which relies on selection of closest frontiers with high information gain (called next-best-view gain) (see Sect. 4.2). Next-best-view gain remains highly popular measure in existing solutions for 3D space exploration and coverage problems. Another early take on this problem was made by [2] with a similar next-best-view solution, although addressing 3-DoF unmanned ground vehicle (UGV) with omnidirectional depth sensor.

Another notable solution to the problem was proposed by [5] using RRT* path-planning algorithm [13] replacing typical frontier-based algorithms. They improved RRT* algorithm with execution of the current best branch that stops after one node, which have improved the result and resilience to changes compared to existing solutions. Authors augmented their approach in [6] with the ability to explore visible space by surfaces instead of voxels.

Other works in this area also contains some notable ideas: [22] use search for edges of known surfaces instead of unobserved frontiers, which is a good idea for orthogonal environment; [17] take into account the paths that provide the gain in quality to model results inside voxels; [7] implement human-inspired visual attention model that plan the exploration towards visually salient parts of RGB image. [18] use genetics algorithms as an extra step to refine the movement of the robot between selected frontier viewpoints.

We assuming that most of these algorithms are not well suited for a goal of getting total coverage in dense clutter environments due to their stochastic based approach. Althrough most algorithms in this area are suitable for any robot configurations, we designed DCEGen to use with basic 3-DoF UGV, without considering applicability to other configurations.

3 Overview

This section describes our approach in detail. After formal problem definition, an overview for random environment generation algorithms is presented.

3.1 Formal Definition of Autonomous Exploration and Coverage Problem in 3D

Usually the problem's goal in this research field is to autonomously explore a bounded 3D space V $\subset R^3$ while minimizing the time to achieve total coverage. Volume V is partitioned into voxels, each categorized as free, occupied, non-observed or residual (not observable) type. The goal of exploration is considered achieved when $V_{free} \cup V_{occ} = V \backslash V_{res}$.

We start addressing specifics goal of dense clutter environment exploration with formal definition by adding extra "hard to observe" voxel type, indicated as V_{hto},. We consider environment as densely cluttered if it contains a large portion of voxels that are only observable from limited view points in free configuration space.

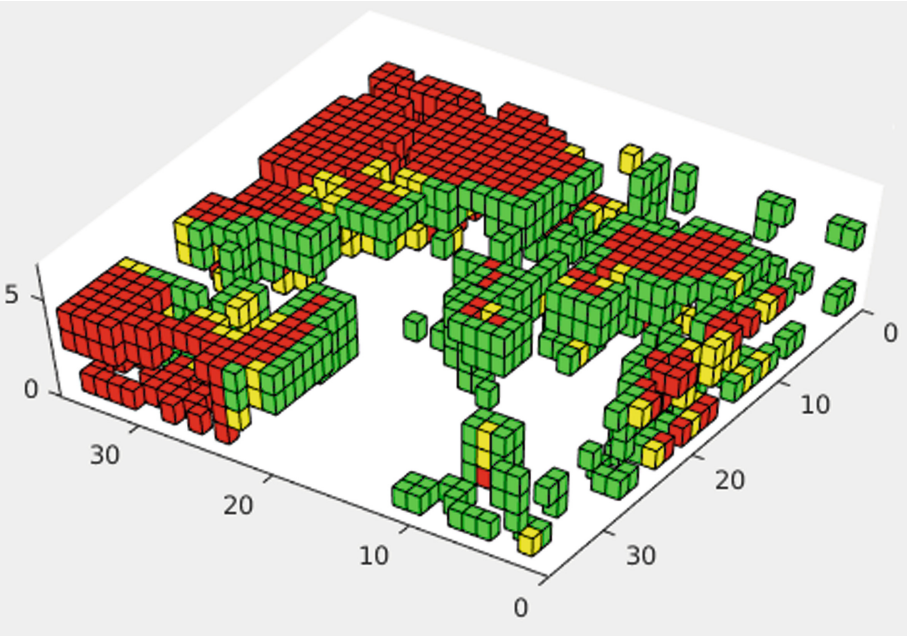

Fig. 1. Labeled generated environment. Green voxels are observable voxels, red are non-observable voxels and yellow are "hard to observe" voxels (V_{hto}). Walls and voxels above the robot head are omitted for better visualization. Visualized in MATLAB environment using [23]. (Color figure online)

In addition, V_{hto} can be used as a performance indicator for more successful algorithms during early iterations, as top performing algorithms must not skip close V_{hto} for later. We also suggest using V_{hto} for real environment experiment quality assessment by juxtaposing automatic and manual coverage for those areas, as an alternative to entirety of the environment.

3.2 Random Dense Clutter Environment Generation

The core contribution of this paper is description of algorithm DCEGen used for generation of cluttered environment for 3-DoF UGV robot configuration. Our algorithm composed of two steps. First, sample a random set of point tuples defining lines of random width in 2D space that are joined together in a single

figure. Resulted figure is defines as collision-free path in space. Second, placing random voxels onto 3D space outside the collision-free space with the addition of random cuboids. Example of a generated environment is presented in Fig. 1.

DCEGen includes option to make environment to look more rugged or more structured like urban environment. Another options controls dilation of the environment from free space as it goes up to form more heap-like structures. See the difference in Fig. 2. For future work we are planning to implement non-planar collision-free space for more complicated mobile robots with moving Z-axis depth sensor.

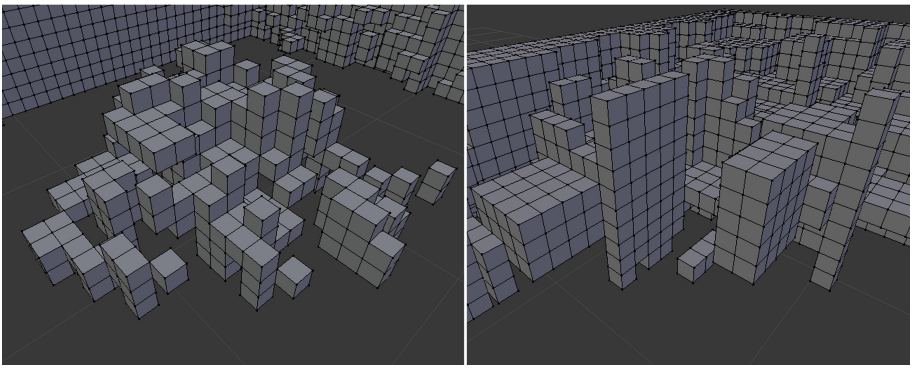

Fig. 2. Heap-like (left) and orthogonal (right) types of environment.

Export as 3D Models. Preceding using generated environments as 3D models in simulation experiments, we have to apply smoothing to those models to reduce voxel-based appearance. Smoothing is done in such a way that the resulting model does not touch 3D grid. This is required to negate noise factor and also to make a dense clutter environment looks more realistic when voxel size is quite high. Our solution for smoothing is to move away voxel faces in the opposite direction to their normals, repeated for several steps in Blender [1]. We subdivide faces by 2, thus every face is now splitted to 4 faces. Smoothing vertices are then applied with user-defined multiplier value. Further inset faces operation are executed with negative depth and resulting model is exported to DAE file format for use in Gazebo simulator. For convenience, we supply a Python script using Blender that make whole process automatic. Figure 3 shows a simple example of a hole that gets transformed after execution of our smoothing process. Depending on your goals, the result can be less or more smoothed. Fully generated environment is presented in Fig. 5.

Count Visible Voxels. The core feature of DCEGen analyzer is the ability to count number of observable and hard to observe voxels of free space in a generated environment. For this process we use octomap_mapping package [12] which

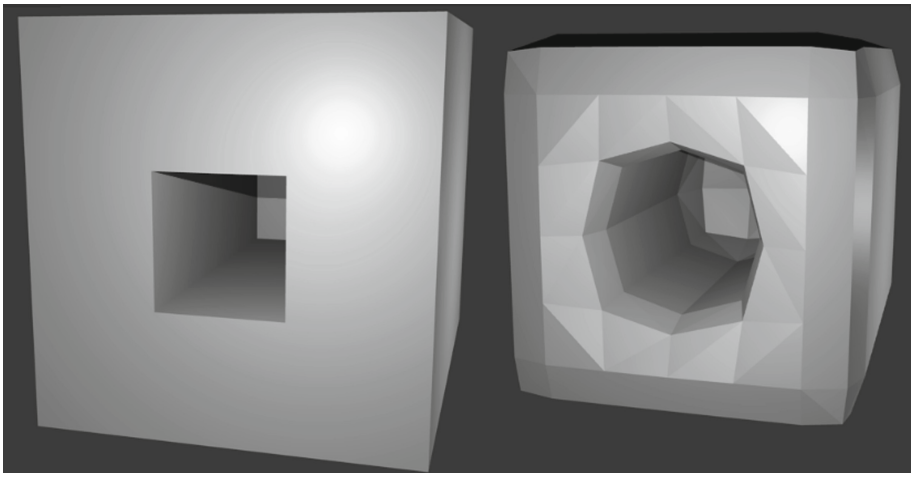

Fig. 3. The proposed solution for smoothing voxels in a simple case of a hole. Note that there are V_{hto} inside the hole (considering that viewpoints are located only around the model).

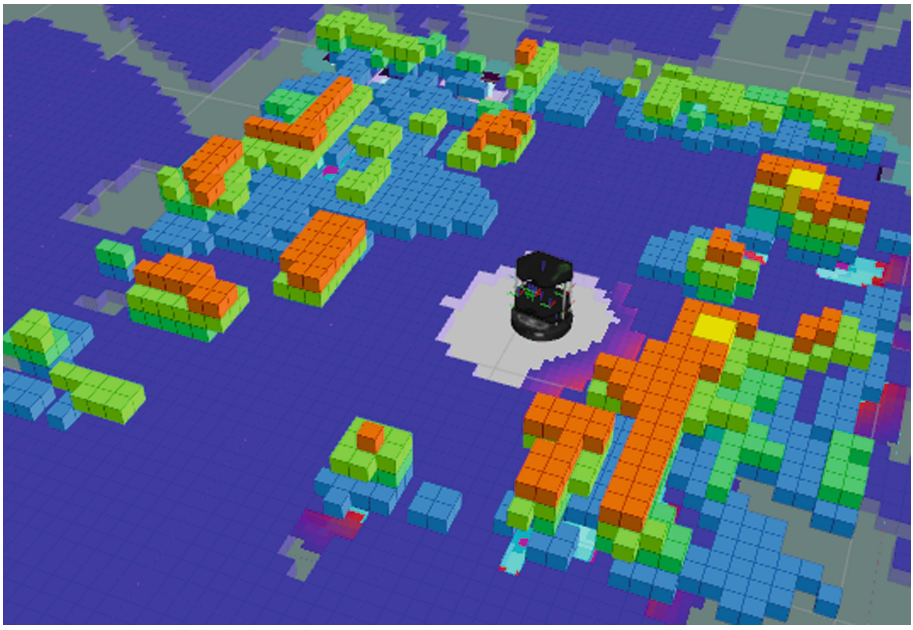

Fig. 4. The results of octomap_mapping package that brings scans back to voxel form. Different colors are used only for demonstration purposes to emphasize voxels' height.

translates scanned 3D reconstruction back to voxel form, usage shown in Fig. 4. Possible coverage for a map is assessed by counting every visible voxel from every view point with detail summarization. If a voxel was seen from limited amount of viewpoints below HTO threshold from the whole environment, then it is considered V_{hto}. Robot is teleported to viewpoints to speed ups observation process and controlled by changing robot model position in Gazebo via pause_physics service that is accessable through ROS interface. After processing a viewpoint the script resets octomap_mapping package but before that it increments the visibility of covered voxels.

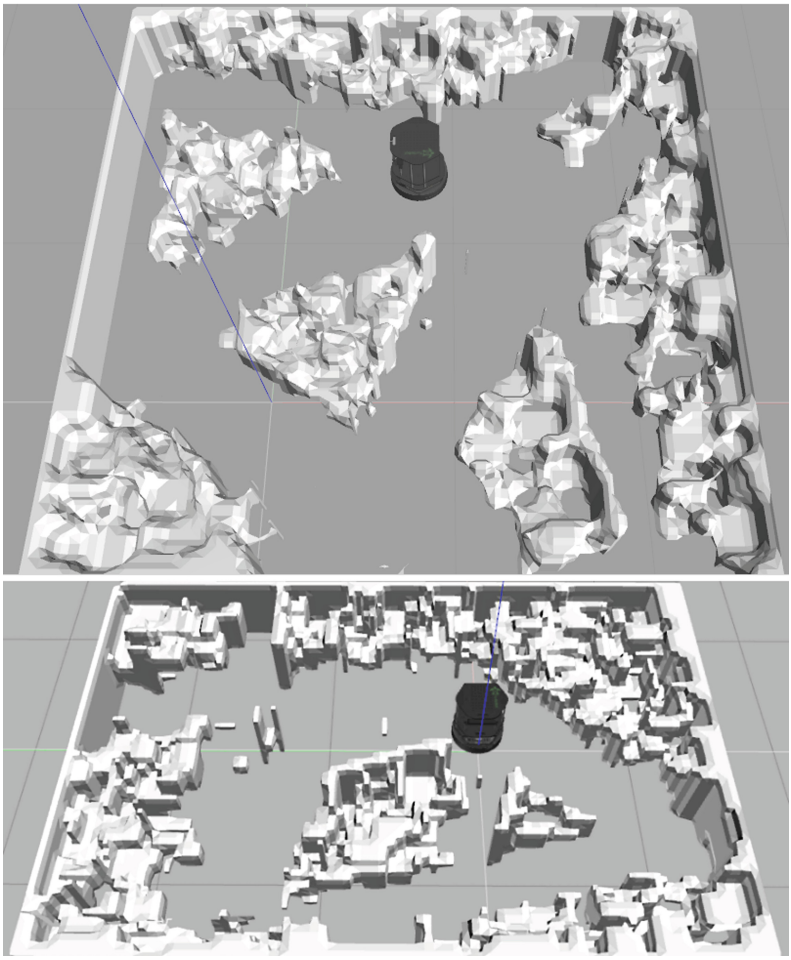

Fig. 5. Example of a generated environment with a robot. A user can define an environment size.

Free space is a continuous space, so it is not possible to sample all free space for viewpoints. Therefore, we propose sampling of viewpoints as more dense 2D grid than initial robot's free space 2D grid with density multiplier parameter. The quality of voxel visibility can be double-checked by projecting covered voxels into a 3D model [20,24].

4 Conclusions and Future Work

In this paper we have presented an algorithm for generating dense clutter environment DCEGen, which was designed for assessment of autonomous exploration and coverage algorithms that are employed for UGVs. We implemented the algorithm in Python programming language and it works within Robot Operating System framework inside Gazebo simulator. We presented a novel metric of space point visibility that is based upon a number of "hard to observe" voxels and will serve as a useful assessment tool in simulation and real world experiments. Our next goal is to create an algorithm that improves upon existing solutions for dense clutter environments. The Python code is available for public use in our Gitlab repository.[1].

Acknowledgments. This work was supported by the Russian Foundation for Basic Research (RFBR), project ID 19-58-70002.

References

1. Blender: Free and open source 3d creation. https://www.blender.org/
2. Adán, A., Quintana, B., Vázquez, A.S., Olivares, A., Parra, E., Prieto, S.: Towards the automatic scanning of indoors with robots. Sensors (Basel) **15**(5), 11551–11574 (2015)
3. Afanasyev, I., Sagitov, A., Magid, E.: ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment. In: Battiato, S., Blanc-Talon, J., Gallo, G., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2015. LNCS, vol. 9386, pp. 273–283. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25903-1_24
4. Andreychuk, A., Yakovlev, K.: Path finding for the coalition of co-operative agents acting in the environment with destructible obstacles. In: Ronzhin, A., Rigoll, G., Meshcheryakov, R. (eds.) ICR 2018. LNCS (LNAI), vol. 11097, pp. 13–22. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99582-3_2
5. Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R.: Receding horizon "next-best-view" planner for 3D exploration. In: IEEE International Conference on Robotics and Automation (ICRA) (2016)
6. Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., Siegwart, R.: Receding horizon path planning for 3D exploration and surface inspection. Auton. Robot. **42**(2), 291–306 (2018)

[1] https://gitlab.com/LIRS_Projects/Simulation-3d-reconstruction/tree/master/autonomous_exploration_and_coverage. Note for reviewers: The access will be opened after the conference if the paper is accepted.

7. Dang, T., Parachristos, C., Alexis, K.: Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics. In: IEEE International Conference on Robotics and Automation (ICRA) (2018)
8. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: large-scale direct monocular SLAM. In: IEEE European Conference on Computer Vision (ECCV), pp. 834–849 (2014)
9. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robot. Auton. Syst. **61**(12), 1258–1276 (2013)
10. González-Banos, H.H., Latombe, J.C.: Navigation strategies for exploring indoor environments. Int. J. Robot. Res. **21**(10–11), 829–848 (2002)
11. Heng, L., Gotovos, A., Krause, A., Pollefeys, M.: Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments. In: IEEE International Conference on Robotics and Automation (ICRA) (2015)
12. Hornung, A.: Octomap_mapping ros package. wiki.ros.org/octomap_mapping/
13. LaValle, S.M.: Rapidly-exploring random trees a new tool for path planning. Technical report (1998)
14. Lavrenov, R., Matsuno, F., Magid, E.: Modified spline-based navigation: guaranteed safety for obstacle avoidance. In: Ronzhin, A., Rigoll, G., Meshcheryakov, R. (eds.) ICR 2017. LNCS (LNAI), vol. 10459, pp. 123–133. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66471-2_14
15. Lavrenov, R., Zakiev, A.: Tool for 3D Gazebo map construction from arbitrary images and laser scans. In: 2017 10th International Conference on Developments in eSystems Engineering (DeSE), pp. 256–261. IEEE (2017)
16. Magid, E., Tsubouchi, T., Koyanagi, E., Yoshida, T.: Static balance for rescue robot navigation: losing balance on purpose within random step environment. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 349–356. IEEE (2010)
17. Mendez, O., Hadfield, S., Pugeault, N., Bowden, R.: Taking the scenic route to 3D: optimising reconstruction from moving cameras. In: IEEE International Conference on Computer Vision (ICCV) (2017)
18. Meng, Z., Qin, H., Chen, Z., Chen, X., Sun, H., Lin, F., Ang Jr., M.H.: A 2-stage optimized next view planning framework for 3-d unknown environment exploration and structural reconstruction. IEEE Robot. Autom. Lett. **2**(3), p1680–1687 (2017)
19. Panov, A.I., Yakovlev, K.: Behavior and path planning for the coalition of cognitive robots in smart relocation tasks. In: Kim, J.-H., Karray, F., Jo, J., Sincak, P., Myung, H. (eds.) Robot Intelligence Technology and Applications 4. AISC, vol. 447, pp. 3–20. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-31293-4_1
20. Rodríguez-Teiles, F.G., Pérez-Alcocer, R., Maldonado-Ramírez, A., Torres-Méndez, L.A., Dey, B.B., Martínez-García, E.A.: Vision-based reactive autonomous navigation with obstacle avoidance: towards a non-invasive and cautious exploration of marine habitat. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 3813–3818. IEEE (2014)
21. Ronzhin, A., Saveliev, A., Basov, O., Solyonyj, S.: Conceptual model of cyberphysical environment based on collaborative work of distributed means and mobile robots. In: Ronzhin, A., Rigoll, G., Meshcheryakov, R. (eds.) ICR 2016. LNCS (LNAI), vol. 9812, pp. 32–39. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43955-6_5
22. Senarathne, P.G.C.N., Wang, D.: Towards autonomous 3D exploration using surface frontiers. In: IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR) (2016)

23. Shabat, Y.B., Fischer, A.: Design of adaptive porous micro-structures using curvature analysis for additive manufacturing. In: the 25th CIRP Design conference, Haifa, Israel (2015)
24. Shimchik, I., Sagitov, A., Afanasyev, I., Matsuno, F., Magid, E.: Golf cart prototype development and navigation simulation using ROS and Gazebo. In: MATEC Web of Conferences, vol. 75. EDP Sciences (2016)
25. Weise, T., Leibe, B., Van Gool, L.: Fast 3D scanning with automatic motion compensation. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
26. Yakovlev, K., Khithov, V., Loginov, M., Petrov, A.: Distributed control and navigation system for quadrotor UAVs in GPS-denied environments. In: Filev, D., Jabłkowski, J., Kacprzyk, J., Krawczak, M., Popchev, I., Rutkowski, L., Sgurev, V., Sotirova, E., Szynkarczyk, P., Zadrozny, S. (eds.) Intelligent Systems 2014. AISC, vol. 323, pp. 49–56. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-11310-4_5
27. Yang, M.D., Chao, C.F., Huang, K.S., Lu, L.Y., Chen, Y.P.: Image-based 3D scene reconstruction and exploration in augmented reality. Autom. Constr. **33**, 48–60 (2013)