

The Architecture of the Robot-Finder Based on SLAM and Neural Network

Iakushkin Oleg(✉), Ruslan Sevostyanov, Alexander Degtyarev, Egor Krasilnikov, Maria Mingazova, Alexey Rusakov, Olesya Kondratieva, and Andrey Bobryshev

Saint-Petersburg University, 7/9 Universitetskaya nab., St. Petersburg 199034, Russia
o.yakushkin@spbu.ru

Abstract. The task of this paper is to find lost or frozen people in the wood. That takes accurate exploration of a large space with a minimum time duration. This work is dedicated to the architecture part of the assigned task.

We give an architecture for robot-finder capable to find a human being in the wood or in the snow area. For us, the robot is blending of two elements, which we can develop independently. That are a wheeled platform and an operating module. In this task, we look at the second one. During that way we assume that the first one is developed, therefore the robot is driving upon the airbag or wheeled platform.

Our solution to this task is architecture and algorithm. These two are made for and directed to learn robot follow the map and detect human being alongside. We use computer vision, neural network and GPS technologies. In the end, we have a theoretical basis for developing robot-finder.

1 Introduction

In our thought, the robot is final equipment which capable and aimed to complete the task provided by a human being. The robot includes in itself required equipment and program.

In this paper, we solve the problem dedicated to missing human. Young and old can lost in the wood or in unknown areas without an opportunity to find a way back. This is the reason why we create a robot-finder to be competent to help search and rescue teams.

The task of this robot is to find a human being in the bounded area for a finite period of time. It must be sustained for the environment condition and work autonomously. We assume to use our robot in the unknown, dynamically changed terrain. Also, we consider obstacles avoiding in an aggressive environment such as animals.

This research was partially supported by Russian Foundation for Basic Research grant (project no. 17-29-04288) and by SPbU (Saint Petersburg State University) grant no. AAAAA18-118071790047-9 (id: 28612502).

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11622, pp. 761–771, 2019.
https://doi.org/10.1007/978-3-030-24305-0_57

At the finish of this paper, we will provide a theoretical basis for a robot that can find a human being and deliver water or medicines.

1.1 Functional Subsystems and Their Tasks

To give a proper solution we work on the five subsystems. They belong to the main controller. The controller process an intermediate results from these subsystems. It can be a microcontroller or computer. The subsystems that are needed for full functionality robot are provided in the Table 1 below.

Table 1. Robot-finder subsystems

Subsystem	Task
SLAM algorithm	Control logical moving by changing local route
Neural network	Detect human in a visible area
Wheeled platform/airbag	Perform physical moving
GSM-modem	Send information
GPS-tracker	Give global coordinates

2 Related Works

There were many explorations about robots moving in the unknown hazards and dynamically changed areas. Classification of the existing approaches was given in [15]. Mathematical model of the maneuvering between moving obstacles was given in [4,7,14]. In the [5] authors gave a motion algorithm for self-driving on the bumpy road using ultrasonic sensor and gyroscope. In the [9] fuzzy neural network for obstacle avoiding is provided.

For the navigation task, there were suggested several different approaches. First is completed library for perceiving the environment using mesh estimation [10]. Second is capable to detect obstacles using a neural network that accept RGB image [9]. The third solution uses accurate stereo-vision technology [2,3]. Fourth suggest using sensors to estimate the distance to the object [6,8]. Fifth paper [12] describes a navigation algorithm. Its authors give a clear algorithm for obstacle avoidance. All these approaches take a different amount of resources and give different benefits.

The third part of our robot that takes the proper solution is architecture. In the paper [1] 2d navigating system architecture is suggested. There are the odometry and location information let compute proper robot velocity. An interesting approach is also presented in [1,11], where sensors are placed on the board, information flows from which are processed by the computing module. These works have affected to our robot. But there is still no out of the box solution that can solve our problem. In particular, there is no complete obstacle avoidance algorithm.

Our decision is to use computer vision libraries with open-source code. These packages are using depth map data. This choice was accepted because of our needs in free and reusable solutions. This library builds an analysis-friendly field that shows the location of the obstacle polygon. It gives the distance to the obstacle and its form. It helps to accurately estimate the environment in the robot sight. The benefit of this library in comparison to others is open-source. Thanks to it we are able to support real-time processing of incoming data from the camera.

3 Task

Our main task was to build robot-finder architecture using open-source and the cheapest software and hardware. Therefore we've learned technologies in chosen areas, compare them and choose the most proper. Besides architecture, we purpose motion algorithm. This algorithm requires an only RGB-d camera, network adapter, and microcontroller or small computer. It is possible because we introduce sufficient math model.

3.1 Mathematical Model and Task

Let robot's world model is an unchangeable environment with the only dynamic robot. The motion is performing iteratively, where one iteration includes sufficient actions in Fig. 1. For the easiness we can let that world is 2d.

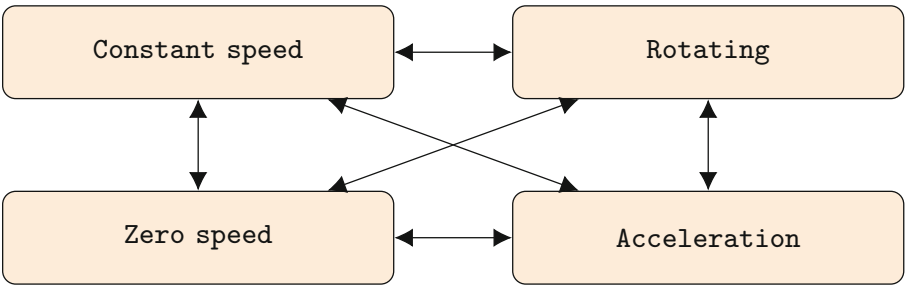

Fig. 1. Robot states.

The robot can be in four states:

- Constant speed
- Acceleration
- Rotating
- Zero speed

The robot includes in itself compute module which gives final commands to motion platform API. We claim that every state is the result of previous states. Therefore the motion model is: robot moves forward in bounded coordinate system, follows a closed graph which vertices are GPS-coordinates. There is a finite number of these vertices. These assumptions allow us to reduce task from moving along the graph to moving along the edge.

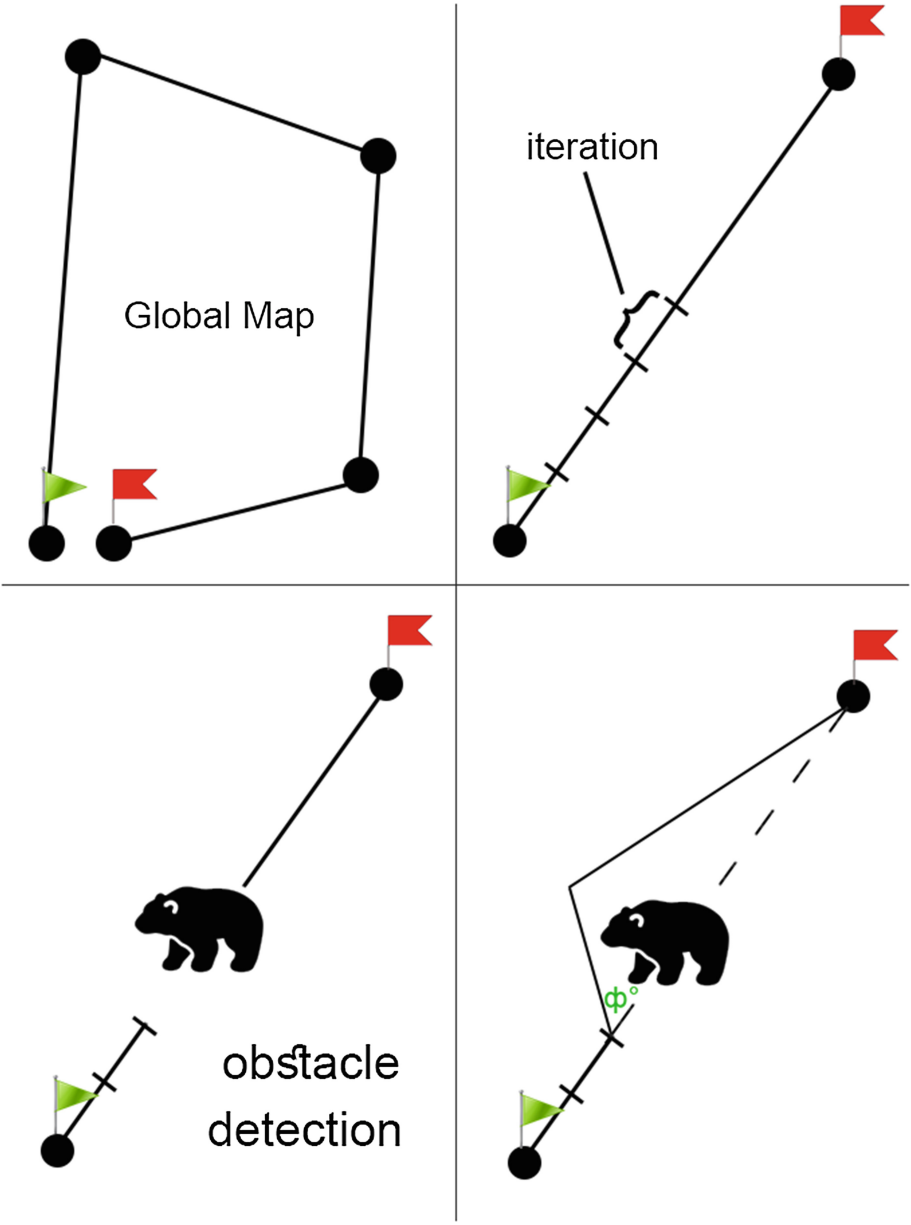

Fig. 2. Changing local route according to the obstacle.

The robot drives straight ahead iteratively in Sect. 4.2. On every iteration, the robot scans terrain to detect human and obstacles. We assume that the algorithm can't handle the situation when it bumps into an animal. It is because of ambiguous behavior. Therefore we can formulate subtask: change the local route according to the new circumstances in Fig. 2. Let us have several parameters:

- Orientation (O)
- Obstacle parameters (Π)
- Target (P)
- Rotating angle (Φ)

Hence we have a formula:

$$\widetilde{\Phi} = \min \Phi(O, \Pi, P) \tag{1}$$

Here Φ is function of three arguments. Minimization is conducted for all possible rotating angles:

$$\Phi \in [O - \frac{\pi}{2}, O + \frac{\pi}{2}]. \tag{2}$$

After minimization and rotating we compute orientation again and proceed iterative algorithm. If the robot detects human being, he goes back to the base station follows the shortest way in Fig. 3.

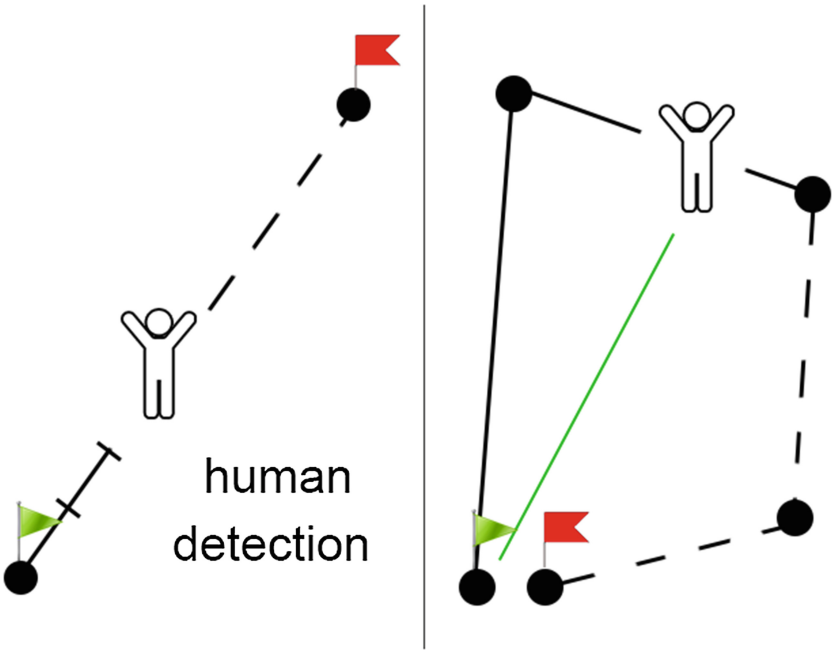

Fig. 3. Changing route after human detected.

4 Positioning Algorithm

Autonomous vehicles require an accurate obstacle detection mechanism for correct navigation in unmonitored environments. During developing the positioning algorithm, the need for its flexibility in the conditions of an uneven road and the possibility of obstacles were taken into account. The basis is the SLAM (simultaneous localization and mapping) method. This approach allows to simultaneously follow the global route and build a local route in accordance with the circumstances of the movement.

To implement this algorithm, we devise a local model of robot behavior. It lies in the fact that the robot recognizes the nearest obstacles and circles them with a minimum deviation from the specified trajectory.

4.1 The Efficiency Criterion of the Iterative Algorithm

For search work in rough terrain, we have developed an algorithm for an effective search for a person. The performance criteria include two conditions:

– The robot must follow the specified route as close as possible.
– In most cases, the robot should recognize a person who was met on the way.

It is worth saying that it is a possible situation when the robot revealed not a lost person, but just a passer-by. In this case, after sending the data to the server, the robot operator sends the task of finding a person again on the remaining route. The block diagram of the algorithm is presented in Fig. 4

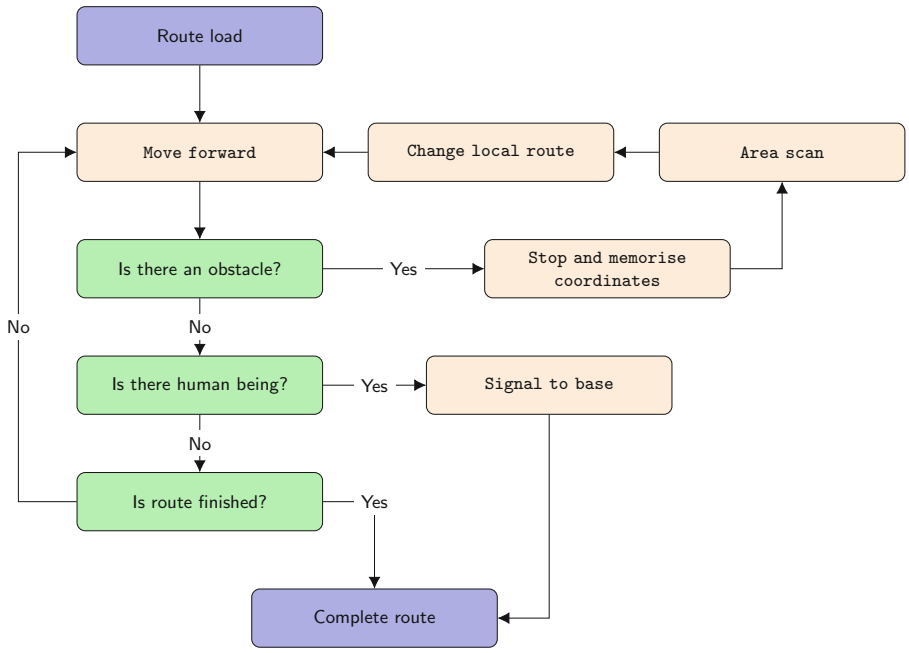

Fig. 4. Iterative algorithm.

4.2 Description of the Iterative Algorithm

The robot starts moving from a given point and executes its program iteratively in an infinite loop. During one iteration, the robot scans the obstacles in its path. To do this, it uses a special program based on the OpenCV library, which receives a frame from the RGB-Depth camera. On the basis of the frame, its parameters are calculated: the presence, size, and location of obstacles. Considering the data, the robot changes the route. Based on the orientation and the modified route, the robot decides to move on or stop and make a turn (see Sect. 3.1 Mathematical model and task).

After detecting obstacles, the robot proceeds to detect the person in sight (see Sect. 5.2). The neural network gives an answer about detecting a person or that nobody is detected. In this article, we do not consider meeting with animals and their potential threat to the algorithm operation. This is also stipulated in the task model (see Sect. 3.1). Based on the answer about the presence of a person, the robot either approaches it and checks who it is, and then helps it by explaining the shortest path out of the forest, or continues to move, returning to navigation tasks. If the robot did not find the person, he completes the route, passing it completely.

It should be said that the task is not intended to rotate the camera in order to more likely to detect a person. This approach, however, can be implemented in the following studies.

5 Architecture (System Design)

5.1 Hardware and Software

The diagram in Fig. 5 shows the system architecture.

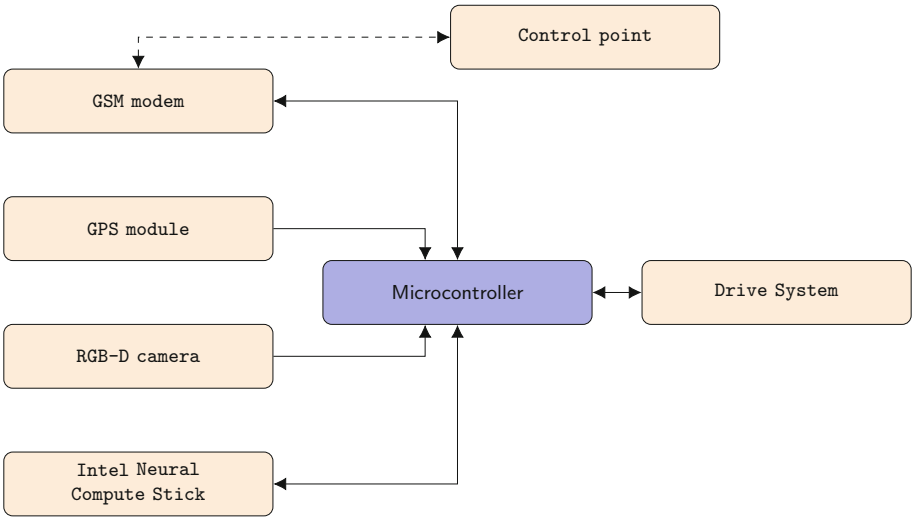

Fig. 5. The architecture of the prototype system architecture.

The robot is controlled by two microcontrollers. One of them combines a set of sensors consisting of a GSM modem, a GPS module, an RGB-Depth camera, and a coprocessor. We use two microcontrollers as an intermediate layer connecting the individual components into a single system. One of them controls the movement, passing commands through the API of the platform to go in a given direction or to stop. The second microcontroller is responsible for choosing one or another command.

It collects environmental information from the sensors, organizes the processing of the received data, and sends the final result of the calculations to another microcontroller as an instruction about changing, continuing the motion path or stopping. Data about the current situation come from the GPS module and RGBD camera. GPS provides the world coordinates of the robot. The RGB-Depth camera sends pictures of the surrounding space to the 2D RGB board and builds a 3D map in the form of a cloud of points [13]. Data from these sensors are needed to solve the issue of navigation.

Communication with the remote control point is carried out using a GSM modem. Using this communication channel, the robot first receives the coordinates of the terrain in which it is necessary to search for a person, and then sends back the search results or notification of the return. The task of pattern recognition is solved by a coprocessor connected to the microcontroller board using a trained neural network.

5.2 Convolutional Neural Network for Human Detection

We use the tiny YOLOv3 for searching for a person in sight of the robot. This algorithm is a fully convolutional neural network. The extractor of this network is DarkNet, which highlights the area in the image, where objects are further detected and classified. Then 53 layers are superimposed on top of the extractor. In tiny YOLOv3 object detection is performed by applying detection kernels to feature maps in different layers of the network. Dimensions of kernels are (1×1) and feature maps are presented in two different sizes.



Fig. 6.

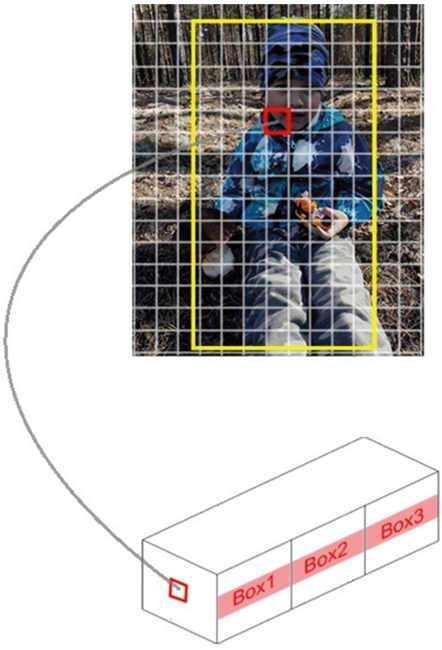

Fig. 7.

This is necessary for detecting small objects more often in Figs. 6 and 7 because the robot has to find a person, who got into an emergency, as quickly as possible. As soon as the robot detects a person, he will be provided with help. Tiny YOLO v3 makes a prediction at two scales which are precisely defined by reducing the size of the input image by 32 and 16 times respectively.

This approach is more resource-intensive and decreases the speed of the algorithm and also allows significantly reduce the number of undetected objects. This is because features maps analysis occurs in different layers of the network which allows detecting objects at different scales of image. Therefore it is possible to detect a person at a greater distance from the robot.

The robot scans the area during the motion, makes RGB-d photo in sight of the camera and transmits RGB image to the neural network processing. Further received image scales and neural network detects objects (humans) in it. If the human is found, the robot will start to approach him and send a GPS signal. It tries to come as close as possible and compare object on RGB image with the point cloud, received from the camera. After that, the robot stops near the person, send detection signal and, if it is possible, photo to headquarters.

6 Conclusions

In this article, we presented the algorithm and architecture of the robot-rescuer. By analyzing the existing research, we came to the conclusion that solutions

can be optimized. Our algorithm of robot behavior was presented on this publications base. SLAM-based algorithms joined with system design, based on two microcontrollers, connecting sensors at one computational space. It was shown above that the proposed design has the potential to help in emergency situations of loss of people in the forest. In the article, we offer a machine that currently has no analogs. The advantages of our model we see at the following points:

- Cheapnest
- Practical significance
- Ability to move around the area with a lot of obstacles
- Compactness
- Autonomy of work

However, in the course of our work we have identified the following disadvantages:

- Energy dependence, limited operating time
- Using a fixed camera
- For a quick search for the missing one robot will not be enough

The practical importance of our robot we see primarily in cost reducing, through the use of cheaper components, combined with the optimization of the algorithm. Secondly, our robot not that big, so it able to pass through small obstructions. With the introduction of this technology in areas where it is expensive or dangerous to search for missing people, it is possible to improve the situation, due to the timely response of rescue robots that do not need rest.

Future research in this area is important, as it will help to optimize the work of rescue services and save human lives. Applications of such machines can be very diverse. One of these cases may be patrolling the borders of countries, states, cities. We believe that this direction has great prospects in improving the quality of life. Thus, our plans for future research are to expand the limits of applicability thanks to a massive wheeled platform that will be able to move over rough terrain. It is also planned to develop an algorithm for detecting obstacles on the basis of the presented mathematical model. Computer vision technologies and patterns will be used for this purpose.

Acknowledgments. This research was partially supported by the Russian Foundation for Basic Research grant (project no. 17-29-04288) and by SPbU (Saint Petersburg State University) grant no. AAAAA18-118071790047-9 (id: 28612502). The authors would like to acknowledge the Reviewers for the valuable recommendations that helped in the improvement of this paper.

References

1. Ball, D., et al.: Vision-based obstacle detection and navigation for an agricultural robot. J. Field Robot. **33**(8), 1107–1130 (2016). https://doi.org/10.1002/rob.21644, https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21644

2. Brand, C., Schuster, M.J., Hirschmüller, H., Suppa, M.: Stereo-vision based obstacle mapping for indoor/outdoor SLAM. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1846–1853, September 2014. https://doi.org/10.1109/IROS.2014.6942805
3. Brand, C., Schuster, M.J., Hirschmüller, H., Suppa, M.: Submap matching for stereo-vision based indoor/outdoor SLAM. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5670–5677, September 2015. https://doi.org/10.1109/IROS.2015.7354182
4. Cui, M., Sun, D., Liu, W., Zhao, M., Liao, X.: Adaptive tracking and obstacle avoidance control for mobile robots with unknown sliding. Int. J. Adv. Rob. Syst. **9**(5), 171 (2012). https://doi.org/10.5772/52077
5. Evstigneev, M., Litvinov, Y., Mazulina, V., Chashchina, M.: Algorithm for mobile robot cross country motion. Nauchno-Tekhnicheskii Vestnik Informatsionnykh Tekhnologii, Mekhaniki i Optiki **17**(3), 393 (2017)
6. Fabrizio, F., De Luca, A.: Real-time computation of distance to dynamic obstacles with multiple depth sensors. IEEE Robot. Autom. Lett. **2**(1), 56–63 (2017). https://doi.org/10.1109/LRA.2016.2535859
7. Ferreira, A., Pereira, F.G., Vassallo, R.F., Bastos Filho, T.F., Sarcinelli Filho, M.: An approach to avoid obstacles in mobile robot navigation: the tangential escape. Sba: Controle & Automação Sociedade Brasileira de Automatica **19**, 395–405 (2008). http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592008000400003&nrm=iso
8. Hernández-Aceituno, J., Arnay, R., Toledo, J., Acosta, L.: Using kinect on an autonomous vehicle for outdoors obstacle detection. IEEE Sens. J. **16**(10), 3603–3610 (2016). https://doi.org/10.1109/JSEN.2016.2531122
9. Kim, C., Chwa, D.: Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network. IEEE Trans. Fuzzy Syst. **23**(3), 677–687 (2015). https://doi.org/10.1109/TFUZZ.2014.2321771
10. Leishman, R.C., McLain, T.W., Beard, R.W.: Relative navigation approach for vision-based aerial GPS-denied navigation. J. Intell. Rob. Syst. **74**(1), 97–111 (2014). https://doi.org/10.1007/s10846-013-9914-7
11. Moniruzzaman, M., Zishan, M.S.R., Rahman, S., Mahmud, S., Shaha, A.: Design and implementation of urban search and rescue robot. Int. J. Eng. Manufact. **8**(2), 12 (2018)
12. Mouad, M., Adouane, L., Khadraoui, D., Martinet, P.: Mobile robot navigation and obstacles avoidance based on planning and re-planning algorithm. IFAC Proc. **45**(22), 622–628 (2012)
13. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans. Rob. **33**(5), 1255–1262 (2017). https://doi.org/10.1109/TRO.2017.2705103
14. Savkin, A.V., Wang, C.: Seeking a path through the crowd: Robot navigation in unknown dynamic environments with moving obstacles based on an integrated environment representation. Robotic. Auton. Syst. **62**(10), 1568 – 1580 (2014). https://doi.org/10.1016/j.robot.2014.05.006, http://www.sciencedirect.com/science/article/pii/S0921889014000955
15. Subhan, M., Bhide, A.: SSGB COE: study of unmanned vehicle (robot) for coal mines. Int. J. Innovative Res. Adv. Eng. (IJIRAE) **1**(10), 116–120 (2014)