Modelling mobile robot navigation in 3D environments: camera-based stairs recognition in Gazebo

Maksim Mustafin
Intelligent Robotics Department
Kazan Federal University
Kazan, Russia
0000-0002-2229-0104

Tatyana Tsoy
Intelligent Robotics Department
Kazan Federal University
Kazan, Russia
0000-0002-5715-7768

Edgar A. Martínez-García
Industrial Engineering and Manufacturing Dept.
Autonomous University of Ciudad Juarez
Chihuahua, Mexico
0000-0001-9163-8285

Roman Meshcheryakov
Cyber-physical Lab
V. A. Trapeznikov Institute of Control Sciences
of Russian Academy of Sciences
Moscow, Russia
0000-0002-1129-8434

Evgeni Magid
Intelligent Robotics Department
Kazan Federal University
Kazan, Russia
HSE University
Moscow, Russia
0000-0001-7316-5664

2022 Moscow Workshop on Electronic and Networking Technologies (MWENT) | 978-1-6654-9666-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/MWENT55238.2022.9802368

***Abstract*—The task of sensory-based autonomous navigation of mobile robots requires data fusion from multiple sources in order to properly detect and recognize environmental obstacles. One of important issues mobile robots deal in a typical multi-level indoor environment is a stair well detection and negotiation. This paper presents ROS-based stairs detection implementation using onboard cameras of the Russian mobile crawler robot Servosila Engineer. Virtual experiments were performed in Gazebo environment with a single camera and a stereo camera.**

***Index Terms*—Computer vision; Object recognition; ROS; Gazebo; Mobile crawler robot**

I. Introduction

Mobile robots are gradually getting more integrated in various aspects of human life and activities, from participation in one-time dangerous operations to social interaction with people on a daily basis. The dangerous operations include fire-fighting services [1], hazardous industries and production [2], urban search and rescue operations (USAR [3]), and special military operations [4]. Human-robot social interaction provides new concepts and improves existing processes in modern education [5]–[7], manufacturing [8], medical industry [9] and pandemic negotiation [10], human support and assistance with daily tasks [11], [12], restaurant services [13], and entertainment [14]). All these tasks require advanced technologies of sensory-based autonomous navigation within a compound environment.

Autonomous navigation is a robot ability to independently make decisions about its motion and actions, based on its onboard sensors performance and data quality that they provide, including internal states, external environmental information and location. The need for robot autonomy arises when it is assumed to operate in conditions of variable or poor communication with a teleoperation mode control system. The teleoperation mode implies that a robot is controlled by a human operator who remotely sets the robot's motions via control devices (e.g., a control panel or a computer), which could be further improved using intelligent assistance advises and services [15]. While controlled in the teleoperation mode, the robot does not require special algorithms and software for object recognition since a human operator uses information sensory data (e.g., cameras, LIDARs, sonars, etc.) for decision making and manual robot control. Yet, in this mode a human factor might result into inaccurate robot motion or even lead to a complete loss of the robot due to the operator mistakes, distractions or poor attention to details.

In turn, in the case of autonomous or semi-autonomous navigation, the dependency on the human factor drastically decreases. While onboard sensors and processing algorithms might fall behind a human brain in terms of cognition and recognition, they have significantly better perception range and quality than a human vision and audition capabilities. Moreover, data fusion from multiple onboard sensors allow to increase perception capabilities far beyond human senses. This emphasizes an importance of properly designed and tested systems for gathering and processing of incoming data from robot's onboard sensors. Cameras of different types are probably the most common and widely used range sensors in robotics. A broad variety of computer vision algorithms, which allow environment perception [16], object recognition [17], cognitive analysis of a scene [18] and many other functions, are implemented with cameras.

In search and rescue operations, especially in an urban environment, there is a high probability that a robot might encounter complicated yet traversable obstacles on its way. These could be piles and debris, which appeared as a result of of destructive forces, as well as ordinary fully functional constructions that are often found in everyday life. In both cases the robot should be capable to autonomously and successfully overcome these obstacles. One of such typical for a human-populated environment constructions that turn into obstacles for robots during operation is a stair well or a staircase.

This paper presents robot operating system (ROS [19]) based stairs detection implementation using onboard cameras of the Russian mobile crawler robot Servosila Engineer [20]. An existing OpenCV library for recognizing stairs in the real world [21] was ported into ROS via our intermediate link and integrated with corresponding to the real robot types of cameras. Virtual experiments were performed in Gazebo environment with a single camera and a stereo camera.

II. Mobile robot Servosila Engineer

Servosila Engineer (Fig. 1) is a mobile crawler-type robot that is manufactured by the Russian company Servosila [20]. The robot is designed to perform tasks in difficult and dangerous conditions, including hazardous production, engineering services, fire fighting services, search and rescue operations and others.

Fig. 1: Crawler Robot Servosila Engineer

The robot could be employed in indoor environments as well as outdoors, in snow and dust. The robot has a relatively small weight that allows to carry it in a backpack by a single person. This eases the robot use in hardly reachable for a human locations and natural disasters' sites, where the robot transportation by a truck is difficult or completely impossible. Thanks to its design, the robot is capable to climb stairs in the teleoperation mode as well as inspect car interiors. The robot mobility was studied empirically by the authors during stairs climbing experiment (Fig. 2) while its usage for a car inspection was studied in Gazebo simulation (Fig. 3) with a model of the robot that had been created in our laboratory [22], [23].

Fig. 2: Servosila Engineer robot climbs stairs in the teleoperation mode during our experiment in Kazan Federal University, Kremlevskaya street 35

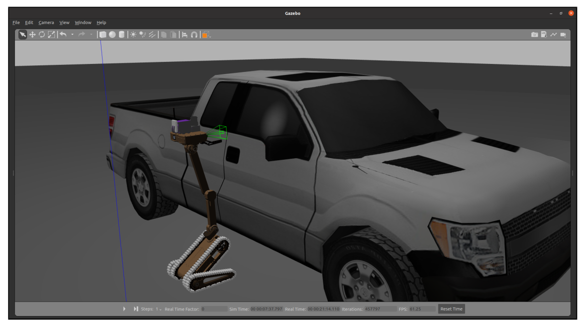

Fig. 3: Servosila Engineer inspects the interior of a car model in Gazebo simulation

The head of the robot Servosila Engineer (Fig. 4) contains an on-board computer that is responsible for all calculations. The on-board computer has Intel Core i7 3517UE 1.70 GHz CPU, 4 GB DDR3 1333 MHz RAM, 32 GB InnoDisk SSD storage and supports Intel HD Graphics 4000. Currently it uses Linux Ubuntu 20.04 LTS operating system. In particular configurations, the head could be equipped with a LIDAR, accelerometer, gyroscope and radio transmitter.

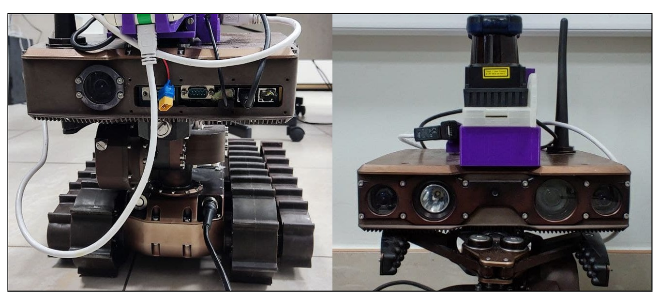

Fig. 4: Servosila Engineer's head. Left image: a monocular rear view camera. Right image (from left to right): a left monocular camera of a stereo vision pair, a flashlight, a camera with variable optical magnification, a right monocular camera of the stereo pair, on top is LIDAR.

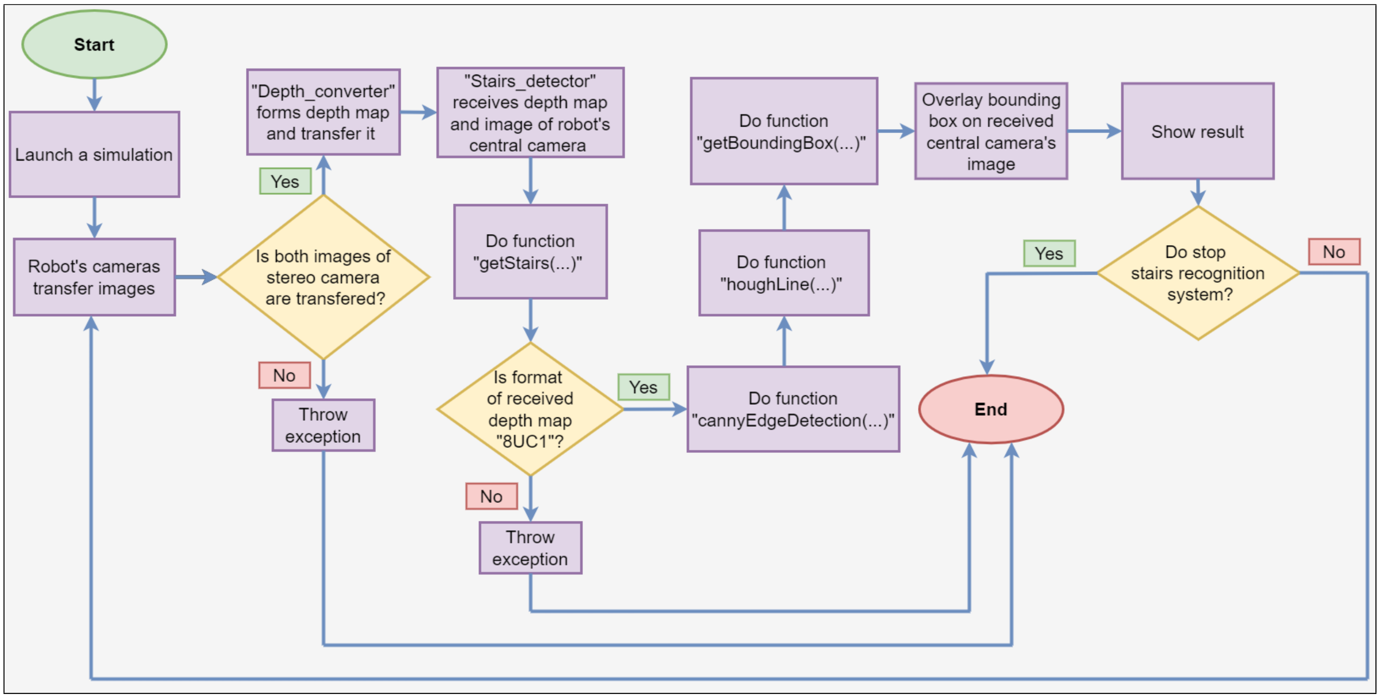

Fig. 5: Block scheme of stairs recognition algorithm in Gazebo

III. Stairs recognition algorithm

The recognition of stairs and their overcoming is an important topic for research in the field of robotics. In [24] authors proposed a staircase recognition and localization for a cleaning robot application using a deep learning model. The article [25] suggested a new algorithm for identifying stairs based on a depth camera. The paper [26] described a staircase recognition system for a UAV that was implemented with a CNN-based imaging process for stair recognition and distance measurements with LIDAR. In [27] Hirasawa proposed to improve stair climbing capabilities by employing passive crawlers. Fukuda et.al. presented involute curve shaped mechanism that concentrates on stair climbing [28].

The stairs recognition system in Gazebo is written using an open source computer vision and machine learning library OpenCV C++. Figure 5 presents a block scheme of the implemented in Gazebo stairs recognition system, which employs Delmerico et.al. package [21].

At initialization step, the simulation of robot with a central camera and a stereo camera in Gazebo world which contains models of stairs is launched. Next, the robot's cameras start publishing the captured images of the surrounding environment to ROS topics. It is important to check if both images of the stereo camera are being published to ROS topics; otherwise, the program throws an exception. Then special ROS node $depth\_converter$ receives stereo camera's images, forms a depth map and publishes it to the ROS topic; at the same time the depth map is demonstrated on the screen within the Gazebo simulation. The heart of the system is $stairs\_detector$ ROS node which is responsible for the stairs detection. After the depth map is formed, $stairs\_detector$ node receives this map and an image from the robot's central camera from ROS topics. Then $stairs\_detector$ node employs $getStairs$, which passes depth map and vector of $cv :: Point$ that will contain a resulting bounding box. But before getting stairs it is necessary to check if a format of the received depth map is $8UC1$ (which means 8-bit single-channel array); otherwise, the program throws an exception. After this verification, $stairs\_detector$ runs $cannyEdgeDetection$ function, which passes the results to $houghLine$ function. The later passes its output to $getBoundingBox$ function that forms a bounding box around potential staircases, which serces as a final output of $getStairs$ function. Finally, the program overlays this bounding box on an image that is obtained by the central camera of the robot.

As a result, if stairs are recognized based in images from the stereo camera and the generated depth map, then a resulting bounding box is placed on an incoming image from the central camera, which shows where a structure similar to stairs is detected. After overlaying a bounding box, an image is demonstrated in a separate window of the Gazebo simulation.

The robot cameras operate at 60 frames per second rate and a result of a stairs recognition may change from one frame to another. This is caused by a formation of a depth map and is also affected by a number of separate objects that have similar to staircases structures. For example, Figures 5, 6 contain the same environment with three staircases structures and there was a slight shift of a bounding box location even though in both frames the staircase model in the left side of the environment was recognized.

Figure 7 demonstrates an example of the right side staircase model recognition with a small offset. When there is a single staircase instance in an environment, the recognition algorithm places a bounding box in a such way that the box's frame is settled in the location of the staircase model's frame (Fig. 8).

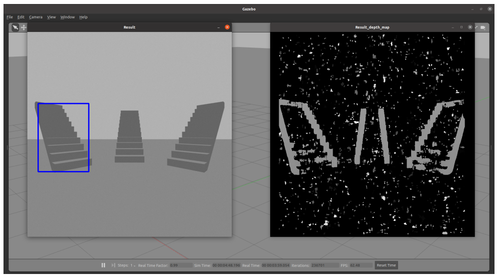

Fig. 5: The left stairs model was recognised

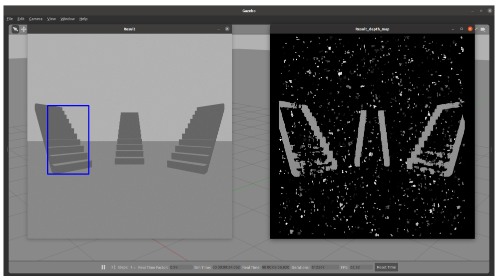

Fig. 6: The left stairs model was recognised

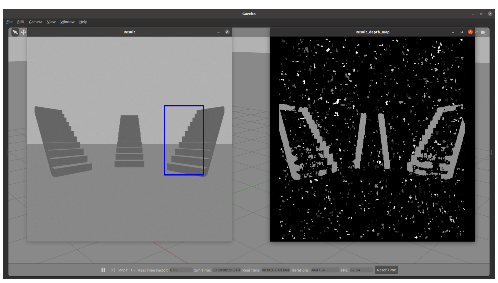

Fig. 7: The right stairs model was recognised

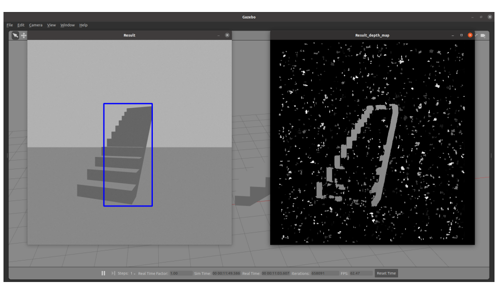

Fig. 8: The model of the stairs located at an angle was recognised

IV. Modeling in Gazebo

A. Getting images from cameras in Gazebo

Various Gazebo plugins are used to create cameras in the Gazebo simulator. To create a regular monocular camera we used $libgazebo\_ros\_camera.so$ plugin; for a stereo camera $libgazebo\_ros\_multicamera.so$ plugin was used [29].

Callback functions for the right and left images were integrated into depth map construction code. In the constructor these functions were tied to the corresponding ROS topics of the stereo camera. The program starts using these functions when new data are transmitted into ROS topics. It is critical that the images from the right and left cameras are transmitted at the same moment of time. To boolean variables $left\_camera\_msg\_updated$ and $right\_camera\_msg\_updated$ serve as a lock, and only when both become $true$ (this means an image from the corresponding camera was received) the algorithm proceeds to $getDepthMap$ function.

B. Creating a depth map

After $depth\_converter$ ROS node receives images from the stereo camera, it is verified that they are available and have a value in $getDepthMap$ function, the algorithm starts forming a depth map. A depth map is an image where each pixel contains depth information (not RGB data) and it is usually represented as a grayscale image [30]. It is for this purpose that the $getStairs$ function checks that the resulting depth map has $8UC1$ format. The depth map contains information about a distance between surfaces of objects from a given point of view and has a broad usage.

To form an appropriate depth map, a thorough selection of parameters configuration is critical. For our algorithm the following settings were used in Gazebo:

```
NUM_DISPARITIES = 0
BLOCK_SIZE = 21
SPECKLE_RANGE = 8
SPECKLE_WINDOW_SIZE = 9
UNIQUENESS_RATIO = 5
TEXTURE_THRESHOLD = 507
MIN_DISPARITY = -39
PRE_FILTER_CAP = 61
PRE_FILTER_SIZE = 5
```

where

- NUM_DISPARITIES is a number of disparities, i.e., a number of pixels to slide a window over as this value increases the range of visible depths, but more calculation is required.
- BLOCK_SIZE larger blocks create smoother images, smaller blocks create noisy images.
- SPECKLE_RANGE controls proximity in value disparities must be to be considered a part of a same blob.
- SPECKLE_WINDOW_SIZE is a number of pixels below which a disparity blob is dismissed as "speckle."
- UNIQUENESS_RATIO if a best matching disparity is not sufficiently better than every other disparity in a search range, a pixel is filtered out.
- TEXTURE_THRESHOLD: filters areas that do not have texture for a reliable matching.
- MIN_DISPARITY: an offset from x-coordinate of a left pixel at which to begin searching.

- PRE_FILTER_CAP and PRE_FILTER_SIZE: normalize an image brightness, enhance texture in a preparation for a block matching.

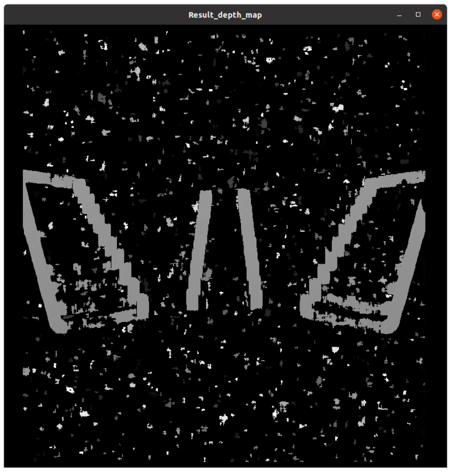

Fig. 10: Example of a good depth map used in stairs recognition system

Fig. 11: Example of a poor depth map for stairs recognition system

Fig. 12: Example of a poor depth map for stairs recognition system

The values of these parameters were set as a result of an empirical research.

Depending on certain values, a depth map may change. For example, the depth map (Fig. 10) was created with the best parameter's, described in a table above. This depth map has a smooth appearance and a small noise value, making it a good selection for the stairs recognition system usage. Another example is a poor depth map (Fig. 11), which is not recommended for usage as it has more noises and is not smooth enough in edges of stairs. This depth map was created with just a single parameter that differed from the recommended setup (the number of disparities was set to 16). Another example of a poor depth map is demonstrated in Fig. 12; it has more noises, which make it less informative for stairs recognition (number of disparities = 16, block size = 15, minimal disparity = -50).

V. Conclusions

An important task of a mobile robot within a typical multi-level indoor environment is a stair well detection and negotiation. This paper presents ROS-based stairs detection implementation using onboard cameras of the Russian mobile crawler robot Servosila Engineer. Virtual experiments were performed in Gazebo environment with a single camera and a stereo camera. We presented an empirically defined selection of configuration parameters that optimize a constructed depth map. As a part of our ongoing work, the algorithm will be transferred onto a real robot and verified in a real world environment. Next, a function of autonomous staircase detection and climbing will be integrated into a control software of Servosila Engineer robot.

Acknowledgment

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program ("PRIORITY-2030").

References

[1] J. Zhu, W. Li, D. Lin, H. Cheng, and G. Zhao, "Intelligent fire monitor for fire robot based on infrared image feedback control," *Fire Technology*, vol. 56, no. 5, pp. 2089–2109, 2020.

[2] J. Petereit, J. Beyerer, T. Asfour, S. Gentes, B. Hein, U. D. Hanebeck, F. Kirchner, R. Dillmann, H. H. Götting, M. Weiser, *et al.*, "Robdekon: Robotic systems for decontamination in hazardous environments," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 249–255, IEEE, 2019.

[3] A. Watanabe, H. Miura, M. Okugawa, and K. Hatanaka, "Verification of scenario for robot-assisted fire-fighting and rescue operations," in *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pp. 106–107, IEEE, 2019.

[4] C. Chen, S. Wang, L. Li, S. Ke, C. Wang, and X. Bu, "Intelligent covert satellite communication for military robot swarm," *IEEE Access*, vol. 8, pp. 5363–5382, 2019.

[5] C. Connolly, J. C. Walsh, H. Worlikar, L. Ryan, A. Murray, S. O'Connor, J. Kelly, S. Coleman, V. Vyas Vadhira, E. Newell, *et al.*, "Exploring new frontiers of education using humanoid robots–a case study of patient centred innovation in digital health education," *Irish Educational Studies*, vol. 41, no. 1, pp. 107–115, 2022.

[6] T. Tsoy, L. Sabirova, M. Abramsky, and E. Magid, "Establishing effective teaching for robotics: a comparison study of bachelor students participated in introduction to robotics course," in *Int. Conf. on Artificial ALife and Robotics*, pp. 212–215, 2018.

[7] L. Gavrilova, V. Petrov, A. Kotik, A. Sagitov, L. Khalitova, and T. Tsoy, "Pilot study of teaching english language for preschool children with a small-size humanoid robot assistant," in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, pp. 253–260, IEEE, 2019.

[8] V. Voronin, M. Zhdanova, E. Semenishchev, A. Zelenskii, Y. Cen, and S. Agaian, "Action recognition for the robotics and manufacturing automation using 3-d binary micro-block difference," *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 7, pp. 2319–2330, 2021.

[9] I. Karabegović, E. Husak, S. Isić, E. Karabegović, and M. Mahmić, "Service robots and artificial intelligence for faster diagnostics and treatment in medicine," in *International Conference "New Technologies, Development and Applications"*, pp. 3–20, Springer, 2021.

[10] E. Magid, A. Zakiev, T. Tsoy, R. Lavrenov, and A. Rizvanov, "Automating pandemic mitigation," *Advanced Robotics*, vol. 35, no. 9, pp. 572–589, 2021.

[11] D. Ryumin, I. Kagirov, A. Axyonov, N. Pavlyuk, A. Saveliev, I. Kipyatkova, M. Zelezny, I. Mporas, and A. Karpov, "A multimodal user interface for an assistive robotic shopping cart," *Electronics*, vol. 9, no. 12, p. 2093, 2020.

[12] E. Chebotareva, R. Safin, K.-H. Hsia, A. Carballo, and E. Magid, "Person-following algorithm based on laser range finder and monocular camera data fusion for a wheeled autonomous mobile robot," in *International Conference on Interactive Collaborative Robotics*, pp. 21–33, Springer, 2020.

[13] J. Yang and E. Chew, "The design model for robotic waitress," *International Journal of Social Robotics*, vol. 13, no. 7, pp. 1541–1551, 2021.

[14] J. Lee, D. Lee, and J.-g. Lee, "Can robots help working parents with childcare? optimizing childcare functions for different parenting characteristics," *International Journal of Social Robotics*, pp. 1–19, 2021.

[15] E. Magid, T. Tsubouchi, E. Koyanagi, T. Yoshida, and S. Tadokoro, "Controlled balance losing in random step environment for path planning of a teleoperated crawler-type vehicle," *Journal of Field Robotics*, vol. 28, no. 6, pp. 932–949, 2011.

[16] J. Zhang, R. Liu, K. Yin, Z. Wang, M. Gui, and S. Chen, "Intelligent collaborative localization among air-ground robots for industrial environment perception," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9673–9681, 2018.

[17] J. Cartucho, R. Ventura, and M. Veloso, "Robust object recognition through symbiotic deep learning in mobile robots," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 2336–2341, IEEE, 2018.

[18] D. Cavaliere, S. Senatore, and V. Loia, "Proactive uavs for cognitive contextual awareness," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3568–3579, 2018.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[20] Servosila, "Mobile robot servosila "engineer"." https://www.servosila.com/en/mobile-robots/index.shtm.

[21] J. A. Delmerico, D. Baran, P. David, J. Ryde, and J. J. Corso, "Ascending stairway modeling from dense depth imagery for traversability analysis," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2283–2290, IEEE, 2013.

[22] A. Dobrokvashina, R. Lavrenov, T. Tsoy, E. A. Martinez-Garcia, and Y. Bai, "Navigation stack for the crawler robot servosila engineer," in *2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1907–1912, IEEE, 2021.

[23] I. Moskvin and R. Lavrenov, "Modeling tracks and controller for servosila engineer robot," in *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*, pp. 411–422, Springer, 2020.

[24] M. Ilyas, A. K. Lakshmanan, A. V. Le, and R. E. Mohan, "Staircase recognition and localization using convolution neural network (cnn) for cleaning robot application," *Preprints*, 2018.

[25] J. sheng Liu, J. po Guo, Z. kai Xiong, and H. Li, "Design of intelligent recognition and positioning algorithm for stairs based on depth camera," in *2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pp. 339–343, IEEE, 2020.

[26] Y. J. Choi, T. Rahim, and S. Y. Shin, "Improved cnn-based path planning so an autonomous uav can climb stairs by using a lidar sensor," *IEIE Transactions on Smart Processing & Computing*, vol. 10, no. 5, pp. 390–397, 2021.

[27] J. Hirasawa, "Improvement of the mobility on the step-field for a stair climbable robot with passive crawlers," *Journal of Robotics and Mechatronics*, vol. 32, no. 4, pp. 780–788, 2020.

[28] R. Fukuda, H. Okamoto, T. Kitahara, and R. Yamashina, "Design and implementation of involute-curve-shaped-mechanism for stair climbing," in *2021 IEEE/SICE International Symposium on System Integration (SII)*, pp. 594–599, IEEE, 2021.

[29] Gazebo, "Gazebo plugins in ros." https://gazebosim.org/tutorials?tut=ros_gzplugins.

[30] OpenCV, "Depth map from stereo images." https://docs.opencv.org/3.4/dd/d53/tutorial_py_depthmap.html.