JOURNAL OF SCIENTIFIC, TECHNOLOGY AND ENGINEERING RESEARCH (JSTER)
Bilim, Teknoloji ve Mühendislik Araştırmaları Dergisi, (2021) - 2(1): 11-22
ISSN : 2717-8404 https://dergipark.org.tr/en/pub/jster

RESEARCH ARTICLE

A Mobile Robot Application for Constructing Semantic and Metric Maps of Search and Rescue Arenas with Point-Based Deep Learning

[1] Muhammed KOCAOĞLU, [2] Yunus Emre IŞIKDEMİR, [3] Muhammed Ali UZUN, [4] Kaya TURGUT
[5] Muhammed OĞUZ TAŞ, and [6,*] Burak KALECİ

[1,2,3,4,5,6] Eskisehir Osmangazi University, Electrical and Electronics Engineering, Eskisehir, Turkey.
[6] burakaleci@gmail.com, 0000-0002-2001-3381
[1]mdkocaoglu@gmail.com, 0000-0003-0956-1210
[2]yemreisikdemir@gmail.com, 0000-0001-7022-2854
[3]mdaliuzun@gmail.com, 0000-0002-2679-3980
[4]kayaturgut@hotmail.com, 0000-0003-3345-9339
[5]motas@ogu.edu.tr, 0000-0001-5689-8786

HIGHLIGHTS

- *The semantic and metric maps are generated that first-responders can easily read in post-disaster indoor environments.*
- *A point-based deep learning architecture is employed to produce the semantic map.*
- *Octree-based 3D metric map composes voxels not only occupied and free but also walls, terrain, and ramps.*
- *The experimental results show that the proposed method can produce accurate maps.*

Keywords :
- Search and Rescue,
- Mobile Robot,
- 3D Semantic Map,
- 3D Metric Map,
- Point Cloud,
- Point-Based Deep Learning

Article Info :
- Received: *16.02.2021*
- Accepted: *08.03.2021*
- Published: *21.06.2021*

DOI:10.5281/zenodo.4589489

***Correspondence:**
Burak KALECİ,
burakaleci@gmail.com,
+90 555 7375966.

GRAPHICAL ABSTRACT

This study aims to create semantic and metric maps of a post-disaster indoor environment similar to standard the National Institute of Standards and Technology (NIST) search and rescue test arenas that first-responders can easily read. We prefer to use point cloud data acquired with an RGB-D camera since it does not be affected by post-disaster environments' dusty and dull nature. Besides, each point cloud data is processed separately so that the semantic and metric maps grow incrementally. The Dynamic Graph Convolutional Neural Network (DGCNN) is used to classify points as sematic categories such as walls, terrain, and inclined and straight ramps. RTAB-Map and the semantic map are utilized to generate the octree-based 3D metric map. The experiments are conducted in a simulated environment modelled with Gazebo similar to NIST test arenas to show the effectiveness of the proposed method.

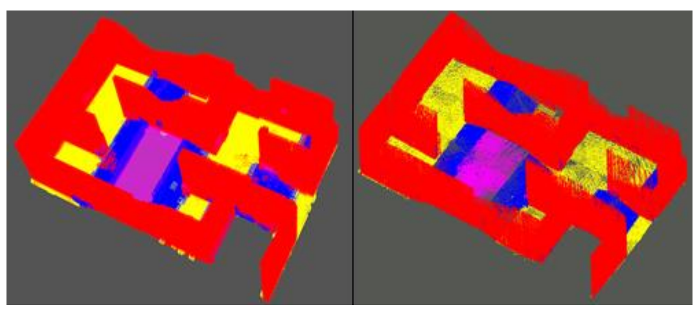


Figure A. The metric(left) and semantic(right) maps of the environment

Aim of Article: *This study aims to construct semantic and metric maps of a search and rescue test arena with a mobile robot.*

Theory and Methodology: *The point cloud data is used to generate semantic and metric maps. DGCNN architecture is applied to determine the semantic class of points. The RTAB-Map and semantic map are utilized to generate an octree-based 3D metric map.*

Findings and Results: *Figure A shows our experimental results. As seen from the figure, the proposed method produced accurate semantic and metric maps.*

Conclusion : *The proposed method process each point cloud data separately and grows the semantic and metric maps incrementally so that it decreases computational complexity.*

RESEARCH ARTICLE / ARAŞTIRMA MAKALESİ

Arama Kurtarma Alanlarında Metrik ve Anlamsal Harita Üretmek için Nokta Tabanlı Derin Öğrenme ile Bir Gezgin Robot Uygulaması

[1] Muhammed KOCAOĞLU, [2] Yunus Emre IŞIKDEMİR, [3] Muhammed Ali UZUN, [4] Kaya TURGUT
[5] Muhammed OĞUZ TAŞ, [*] Burak KALECİ

[1,2,3,4,5,6] Eskişehir Osmangazi Üniversitesi, Elektrik-Elektronik Mühendisliği, Eskişehir, Turkiye.
[6] burakaleci@gmail.com, 0000-0002-2001-3381
[1]mdkocaoglu@gmail.com, 0000-0003-0956-1210
[2]yemreisikdemir@gmail.com, 0000-0001-7022-2854
[3]mdaliuzun@gmail.com, 0000-0002-2679-3980
[4]kayaturgut@hotmail.com, 0000-0003-3345-9339
[5]motas@ogu.edu.tr, 0000-0001-5689-8786

Citation / Alıntı :
Kocaoglu, M, Y.E Işıkdemir, M.A. Uzun, K. Turgut, M. Oğuztaş, B. Kaleci. (2021). A Mobile Robot Application for Constructing Semantic and Metric Maps of Search and Rescue Arenas with Point-Based Deep Learning, *Journal of Scientific Technology and Engineering Research*, 2(1) : 11-22. DOI:10.5281/zenodo.4589489

HIGHLIGHTS/ÖNE ÇIKANLAR

- *Afet sonrası bina içi ortamlarda ilk yardım ekiplerinin kolaylıkla kullanabileceği anlamsal ve metik harita üretilmiştir.*
- *Anlamsal haritanın çıkarılması için nokta tabanlı derin öğrenme mimarisinden faydalanılmıştır.*
- *8-li ağaç yapısında 3B metrik haritada sadece dolu ve boş vokseller değil duvar, zemin ve rampalara ait olan vokseller de yer almaktadır.*
- *Test sonuçları önerilen yöntemin doğruluğu yüksek haritalar üretebileceğini göstermiştir.*

Article Info / Makale Bilgi
Geliş Tarihi : 16.02.2021
Kabul Tarihi : 08.03.2021
Yayım Tarihi: *21.06.2021*

DOI:10.5281/zenodo.4589489

***Corresponding Author / Sorumlu Yazar:**
Burak KALECİ,
burakaleci@gmail.com,
Tel: +90 555 7375966

ABSTRACT / ÖZET

Bina içi ortamlarda zehirli madde yayılımı, sel, yangın ve deprem gibi afetlerden sonra robotlar kullanılarak arama ve kurtarma yapılmasına yönelik çalışmalar son yıllarda hız kazanmıştır. Bu çalışmanın ana motivasyonu, ilk yardım ekiplerinin kolaylıkla kullanabileceği afet sonrası bina içi ortamın metrik ve anlamsal haritalarını oluşturmaktır. Bu çalışmada, afet ortamında karşılaşılabilecek toz, duman ve yetersiz ışıklandırma gibi faktörlerden etkilenmeyen ve nesnelerin geometrik yapısını yüksek doğrulukta temsil edebilen nokta bulutu verilerinin kullanılmasına karar verilmiştir. Her bir adımda alınan nokta bulutu ayrı ayrı işlenerek önerilen yöntemin hesaplama karmaşıklığının düşürülmesi amaçlanmıştır. Anlamsal haritanın üretilmesi aşamasında geçmiş çalışmalardan farklı olarak nokta tabanlı derin öğrenme mimarisi DGCNN kullanılmıştır. Böylece nokta bulutunda yer alan her noktanın anlamsal sınıfı (duvar, zemin, eğimli ve düz rampa) belirlenmiştir. 3B metrik haritanın oluşturulması için RTAB-Map ve anlamsal harita birlikte kullanılarak 8-li ağaç yapısında bir gösterim elde edilmiştir. Bu haritada önceki çalışmalardan farklı olarak sadece dolu ve boş vokseller değil, aynı zamanda duvar, zemin ve rampa sınıflarına ait olan vokseller de yer almaktadır. Önerilen yöntemin test edilmesi için Gazebo benzetim ortamında NIST ortamlarına benzer bir test alanı modellenmiş ve bir Pionner 3-AT gezgin robot teleoperasyon yöntemi ile gezdirilmiştir. Test sonuçları önerilen yöntemin başarılı bir şekilde anlamsal ve metrik harita üretebildiğini göstermiştir.

Anahtar Kelimeler: *Arama ve Kurtarma, Gezgin Robot, 3B Anlamsal Harita, 3B Metrik Harita, Nokta Bulutu, Nokta Tabanlı Derin Öğrenme*

I. INTRODUCTION

After disasters such as fire, earthquakes, floods, and toxic substances, post-disaster indoor environments could be hazardous for search and rescue teams that include humans and animals. The main risks in these environments are the possibility of spreading dangerous matters and collapsing due to the structural breaking down. For these reasons, the studies that addressed search and rescue tasks exploiting robots in these environments have been gained popularity. Although using robots in search and rescue tasks could appear an appropriate solution to avoid risks for humans and animals, post-disaster environments would be challenging even for robots to achieve the tasks they are expected to perform. The primary difficulties in post-disaster environments that the robot must cope with are uneven terrain and poorly lightened circumstances due to these environments' dusty and dull nature. To deal with these challenges, the robots that operate in post-disaster environments must have advanced capabilities such as interpreting raw data, producing semantic information, and being aware of circumstances. Thanks to the improvements in perception technologies and corresponding algorithms and software, robots approach to reach these abilities. However, it is necessary to observe steadily positive and negative aspects of improved methods for giving direction to future works. Unfortunately, the researchers generally may not have the opportunity to test their works since post-disaster indoor environments are rarely faced, and building these environments is complicated and expensive. In order to overcome that problem, DARPA and RoboCup organizations regularly constituted competitions for search and rescue missions.

The RoboCup rescue competitions have been conducted since 2001. The main goal of these competitions is to increase the performance of the robots in search and rescue missions. After the first competition, Kitano and Tadokoro [1] revealed challenges about these missions and introduced the first standards and evaluation metrics. Then, Jacoff et al. proposed reference test arenas for autonomous mobile robots developed by NIST [2]. An example reference test arena is shown in Fig. 1. Also, they defined objective performance evaluation criteria such as the number of locating victims and producing accurate maps that first-responders can easily read. In 2006, the RoboCup rescue competitions were separated into two categories: Agent and the virtual robot competitions. While the agent competitions aimed to coordinate multi-agents systems that include police officers, firefighters, and first-responders to handle disasters in urban scenarios, virtual robot competitions focused on navigation, mapping, and victim detection [3]. After the first virtual robot competition, Balakirsky et al. [4] assessed the performance of the participant teams under specific standards and criteria. In order to evaluate the maps, they used metrics such as attributions that indicate crucial points such as victims and obstacles, accuracy, skeleton and metric quality, and utility that provides cleared regions, locations that the victims are trapped for first responders. The participant teams generally preferred to generate topological maps with 2D lasers. Over the years, the researchers developed new methods to improve the mapping capabilities of the robots. For example, in 2009, teams preferred to use image processing approaches to produce the environments' metric map [5]. These improvements promoted the administration of the competitions to introduce more challenging environments for robots. In 2010, active elements such as smoke, elevator, and the ferry was integrated into the environments. The champion team at RoboCup 2012 used the simultaneous localization and mapping (SLAM) approach to generate geometric map of the environment. The SLAM approach segmented the laser scans into lines depending on the distance between successive points. Besides, the produced line segments were used to construct a semantic map that separates the environment into spaces such as small rooms, large rooms, and corridors divided by doorways. This was the first attempt to construct semantic maps in the search and rescue domain [6]. Sheh et al. [7] overviewed the 16 years of Robocup rescue competitions. The authors emphasized the robots' progress in terms of mobility, autonomy, perception, and adaptation to inhospitable environments such as poor lighting and piles of rubbles. Then, they announced novelties in the competition for the coming years. Probably, the most important one was using Robot Operating System (ROS) together with Gazebo simulation environment. ROS [8] is a commonly used framework to perform robotic applications. It contains libraries (in other words, packages) for a variety of purposes, from mapping to manipulation of a robot arm. ROS generally operates with Gazebo [9] simulation environment, which provides high-performance physics engines and 2D/3D sensors.

Figure. 1. An example reference test arena [5].

After ROS and Gazebo's introducing, participants of the virtual robot competition began to use ROS packages together with Gazebo. For example, in 2017, Chukyo Rescue A Team [10] employed GMapping [11] and Hector SLAM [12] packages for mapping. GMapping [11] is a well-known mapping approach since it was introduced because it could be employed in both indoor and outdoor applications. The main advantage of that method is the ability to create accurate maps with low computational complexity. However, GMapping approach accepts raw 2D laser range data and odometry to generate a 2D occupancy grid map. Similarly, the Hector SLAM [12] approach was applied to produce 2D metric maps. The Hector SLAM's positive aspects are it does not need to use odometry data and has a high update rate. YILDIZ Team [13] utilized Octomap [14] to describe 3D environments. OctoMap was proposed to build a representation (in other words, map) of 3D environments based on octree data structure [14]. First, the data is represented with only one voxel that contains all points. Then, it splits into eight voxels of the same size. The process is repeated until the predefined depth, or voxel size is accomplished. The main contribution of OctoMap representations is using a probabilistic occupancy estimation approach to determine free, occupied, and even unknown voxels. Besides, OctoMap is a memory-efficient representation when it is compared to previous approaches. However, the computational complexity of OctoMap is not appropriate to produce fine-detailed maps. For example, as the voxel size decreases, especially less than 0.05 meters, to describe details of scenes, OctoMap requires significant durations for generating maps. Also, it does not take into account the semantic clues of the scenes. In 2018, SOS RS Team [15] exploited FastSLAM algorithm [16] for mapping. FastSLAM algorithm uses the 2D laser range data as input and generates the geometric map of the environment. This algorithm decreases computational complexity of SLAM approaches significantly. In the same year, AutonOHM Team [17] employed one of the ROS packages, which is called ohm_tsd_slam [18] to generate a 2D map of the environment. The main advantage of this package is to integrate data gathered by RGB-D cameras and 2D/3D laser range finders within the same representation. Similar to the OctoMap approach, ohm_tsd_slam package cannot interpret the scenes in terms of semantic information. In 2019, ATR Team [19] utilized Real-Time Appearance-Based Mapping (RTAB-Map) [20] to create a 2D occupancy grid of the rescue environment. Besides, RTAB-Map also provides 3D point cloud data. RTAB-Map ROS package is integrated with OctoMap so that it can generate the voxel representation of the environment and separate voxels as occupied, free, and unknown [21]. Besides, RTAB-Map contains many feature extractor algorithms such as SIFT, SURF, BRIEF, FAST etc. to recognize some objects such as walls, terrain, and other everyday objects from the visual data. Therefore, it can produce semantic information. Unfortunately, RTAB-Map cannot identify ramps that placed NIST's test arenas since it did not design for search and rescue missions.

As mentioned above, the participant teams of virtual robot competitions generally focused on mapping approaches that generate metric, topologic, and geometric maps. Extracting semantic information or producing semantic maps were considered from only a few teams. However, some previous studies that interested in the semantic classification of walls, terrain, and ramps were proposed. These studies are generally cast into two groups. In the first group, the well-known segmentation methods such as region growing [22] and RANSAC [23] are applied to obtain planar surfaces. Region growing uses a predetermined number of neighbors or search radius to determine the points that belong to the same planar surface. Therefore, it may not be appropriate for real-time applications due to its high computational complexity. On the other hand, RANSAC is a fast and accurate segmentation method aiming to determine a mathematical model for planes. However, it can clusters points that have a similar mathematical model into a plane. This could be problematic since RANSAC does not take into account the neighboring relationship. After planar surfaces are specified, segmented planes are classified depending on geometric features of planes such as normals of points. In the second group, learning approaches are employed for semantic segmentation. These studies generally prefer to apply the Convolutional Neural Networks (CNN) approach to visual data. A recent study proposed by Deng et al. [24] uses CNN to determine point-wise semantic labels with RGB and depth images in NIST test arenas. On the other hand, Turgut and Kaleci [25] concentrated on directly using point cloud data instead of visual data. For that reason, they examined point-based deep learning architectures and they made a comprehensive comparison for PointNet [26], PointNet++ [27], PointCNN [28], and DGCNN [29] architectures that are classified walls, terrain, and ramps in a simulated environment similar to NIST test arenas.

In this study, we aim to produce the 3D metric and semantic maps of a simulated environment similar to NIST reference test arenas with a mobile robot. In this way, we can provide an accurate semantic map, which describes walls, terrain, and ramps, that first-responders can easily read. It is important to note that generating accurate maps is one of the crucial evaluation criteria in virtual robot competitions because an accurate map can significantly decrease searching victim duration and protect first-responders from accidents. Besides, producing a semantic map can contribute robot's autonomous navigation capability. For example, the robot can navigate more reliable by adjusting its velocity when it knows the slope of a ramp. Also, the robot can generate suitable waypoints while traversing ramps to keep its balance. In fact, the robot can consider ramps to enhance its path plan.

The previous studies that address the semantic classification of walls, terrain, and ramps are rare, and they used generally visual and 2D range data. However, the visual data may not be appropriate for dusty, dull, and poorly lightened post-disaster environments. On the other hand, 2D range data cannot be affected by these situations. Nevertheless, its capability to describe the 3D characteristics of the scene is insufficient. At that point, using point cloud data can be a favorable solution to cope with the drawbacks of visual and 2D range data. Therefore, we placed an RGB-D camera on a P3-AT robot and utilized RTAB-Map ROS package to gather point cloud data of a scene. One of the contributions of this study is processing a single scene, which is captured after each predetermined time interval while the robot navigates in the environment by teleoperation. In this way, the semantic and metric maps grow step-by-step, and the computational complexity of the proposed method is decreased. The second contribution is applying a point-based deep learning architecture DGCNN, which receives a single scene and determines point-wise semantic classes, instead of using visual data. The weight of DGNN model generated by Turgut and Kaleci [25] with data acquired in a different simulated environment is used to classify each point. The last contribution is creating a 3D metric map that the robot needs to navigate. Apart from the previous studies, we utilize RTAB-Map and the semantic map while producing the octree-based 3D metric map. In the map, each voxel has a semantic label so that robot can plan its path more reliable.

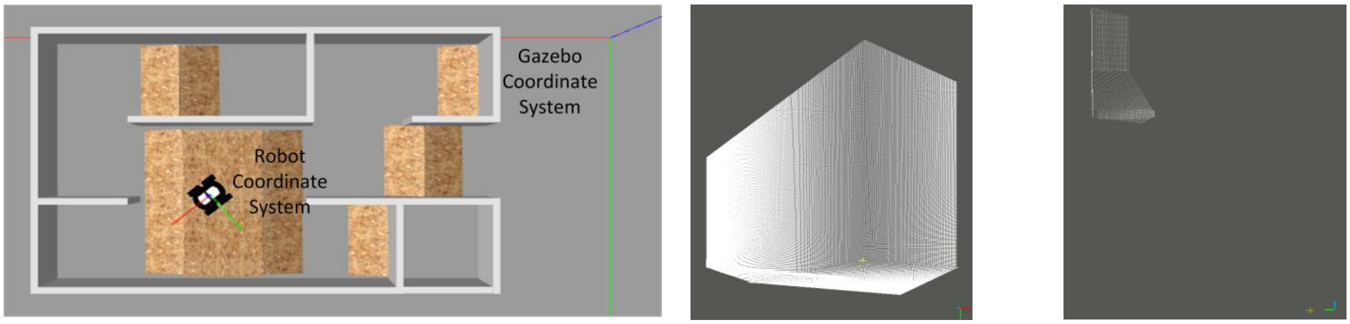
The rest of the paper is organized as follows: In Section 2, the proposed method is explained in detail. The experimental setup and experiments are presented in Section 3. The conclusion and future works are given in Section 4.

II. METHOD

The proposed method consists of three stages. In the first stage, RTAB-Map is employed to gather point cloud data. Besides, we made some modifications to obtain point cloud data for each scene. In the second stage, we construct a semantic map of each separate scene with the aid of point-based deep learning architecture DGCNN. Then, we merge the current map with the global semantic map. Lastly, similar to the semantic map, we build a metric map (in other words, octree map) of the current scene and merge it with the previously generated metric map. In this stage, we obtain free and occupied voxels from RTAB-Map, and then we utilize the semantic map to classify voxels in terms of wall, terrain, inclined and straight ramps.

A. Gathering Point Cloud Data

We used the RTAB-Map ROS package to gather point cloud data. RTAB-Map receives raw point cloud data acquired with the RGB-D sensor of the robot (Fig. 2(a)). It is important to note that the raw point cloud data is obtained according to the robot's local coordinate system (Fig. 2(b)). Hence, RTAB-Map automatically transforms the raw point cloud data into the global coordinate system of Gazebo simulation environment with the aid of the robot's position and orientation. Then, RTAB-Map applies the voxel filter depending on the GridCellSize parameter to downsample the transformed point cloud data (Fig. 2(c)). In this way, computational complexity decreases because the number of samples is reduced without losing the general characteristics of the data. Another crucial issue about the gathering point cloud data is that RTAB-Map accumulates point cloud data during the map producing process as a default property. After this process is completed, it reveals an entire map of the environment. However, this default property is not appropriate for constructing the semantic and metric maps of a search and rescue environment. The main



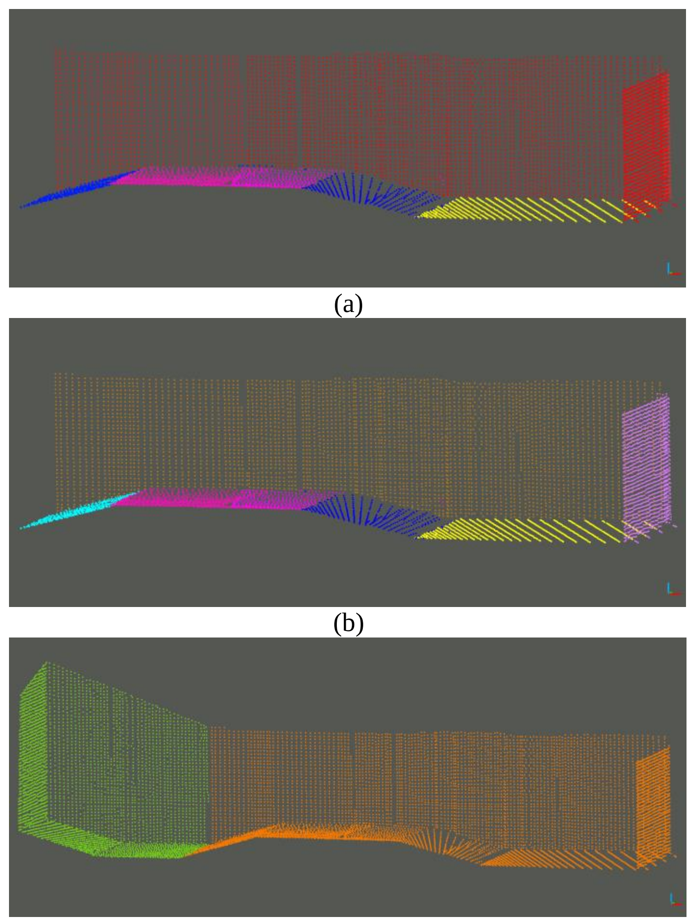
(a) (b) (c)

Figure. 2. Gathering point cloud data. (a) The red, green, and blue lines at the upper-right corner of the image indicate Gazebo's global coordinate system. The same colored lines on the robot show robot's local coordinate system. (b) The raw point cloud data according to the robot's local coordinate system. (c) The downsampled point cloud data according to the Gazebo's global coordinate system.

reason for that is decreasing computational complexity while processing point cloud data separately for each scene. In order to achieve this, we adjusted the MaxNodes parameter.

B. Constructing Semantic Map

After we obtain point cloud data for a scene, the duplicated points are removed from the point cloud data to diminish the computational complexity of the approach. Then, the DGCNN architecture is used to classify points as sematic categories such as wall, terrain inclined, and straight ramps. It is a graph-based architecture that creates local regions for each point in the point cloud. In these local regions, K neighbors of a point (P_c) are found. In order to determine these K neighbors of P_c, if the point features are exist, the distance in feature space is used, otherwise spatial distance is used. DGCNN builds a graph for each local region, and the points (P_c and its K neighbors) that belong to the local region are considered nodes of the graph. The edges of the graph are defined only between P_c and its K neighbors. The weights of the edges are x, y, and z coordinates of neighbors relative to P_c in the first layer. In the successive layers, the weights of edges are features of points relative to the previous layer. After local regions and corresponding graphs are constructed, Multi-Layer Perceptron (MLP) is applied to edges for extracting features of points. The features of a local region are extracted by applying the maximum pooling method to the features of all points situated in the local region. In other words, points are evaluated by considering K neighbors in local regions instead of evaluating each point independently. The process steps mentioned above are called EdgeConv operator, and the operator can easily integrate into any architecture. The DGCNN architecture was created by combining the PointNet architecture and EdgeConv operator. In contrast to architectures that process edges of the graph of local regions, the neighborhood relationship between points is dynamically updated according to feature space. Besides, the local regions are not expanded hierarchically, unlike other architectures.



(a)

(b)

(c)

Figure. 3. An example for construction of semantic map. (a) Clustering of point depending on class labels. (b) Segmentation of planar surfaces of class. (c) Merging current map with global one.

The DGCNN architecture determines the semantic label of each point in the point cloud data. After that point, we cluster the points in terms of their labels. An example point cloud is shown in Fig. 3(a). In this figure, red, yellow, blue, and purple indicate wall, terrain, inclined, and straight ramps classes, respectively. As seen from the figure, each class could contain different planar surfaces, just like walls orthogonal to each other. Therefore, we apply RANSAC [30] segmentation method to segment a class' points that belong to different planar surfaces. This process is repeated for each class so that each planar surface in the scene is determined. The result of the segmentation process is given in Fig. 3(b). We show each planar surface with different colors in the figure. Lastly, segmented planar surfaces are merged with the global semantic map depending on the position and orientation of the planar surfaces of the current and global map. The resultant semantic map is represented in Fig. 3(c). In the figure, green and orange describe the global and current semantic maps, respectively.

C. Constructing Metric Map

Robots generally require an appropriate representation of the environment to achieve the tasks they are expected to perform. One of these representations is the metric map. In previous studies, occupancy grids were frequently applied to obtain 2D metric maps. Occupancy grids describe the environment with a grid composes of equal-size cells. Each cell has a probabilistic value between 0 and 1, depending on its amount of occupancy. Besides, each cell must belong to one of the three states: free, occupied, and unknown. In the beginning, all cells are initialized with 0.5 to indicate the unknown state. Then, as the robot gathers information from the environment, the cells' probabilistic value is updated [31]. In a similar manner, the octree data structure is commonly employed to generate 3D metric maps [14, 21]. An example for the octree data structure is given in Fig. 4. First, the data is represented with only one voxel that contains all points. This is generally called level 0 or root node. Then, it splits into eight voxels of the same size (level 1). Simultaneously, voxels of octree are classified as empty (free) and non-empty (occupied), whether consisting of at least one point or not. The process is repeated until the predefined depth or voxel size (VSIZE) is accomplished.

In this study, we utilized RTAB-Map ROS package to construct an octree. RTAB-Map can also determine the state of the voxels as free and occupied since it integrates with OctoMap. As a default, RTAB-Map ROS package only provides occupied voxels. However, it has RayTracing ability that fills the unknown spaces between

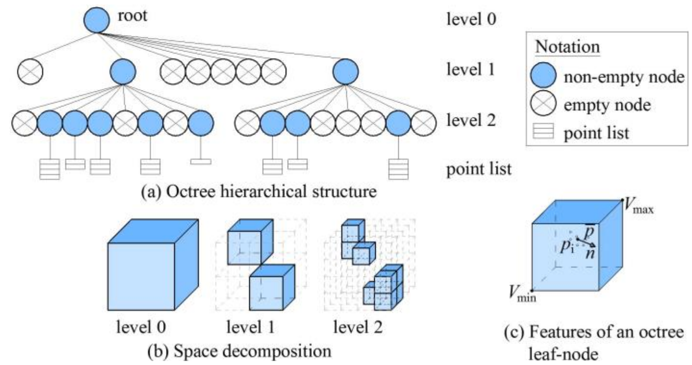


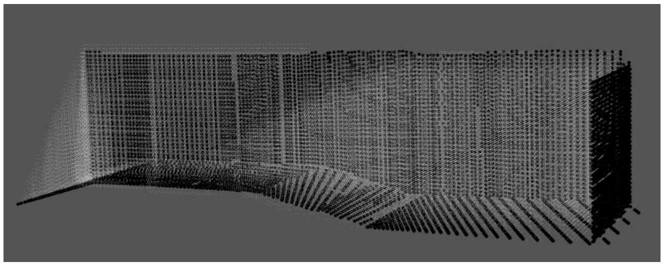
Figure. 4. An example for the octree data structure [32].

the sensor and occupied voxels. Therefore, we enabled RayTracing ability to obtain free voxels. An example for constructing the metric map is shown in Fig. 5. In this example, we used the same scene that is given in Fig. 3. The white and black colors in Fig. 5(a) depict free and occupied points taken from the RTAB-Map. The corresponding metric map is shown in Fig. 5(b). At that point, we used the semantic map to classify occupied voxels into walls, terrain, inclined, and straight ramps. To achieve this, we first identified the points that belong to a voxel. Then, the semantic class of each point in that voxel was specified with the aid of the semantic map. Lastly, we calculated a histogram to count the number of points for each class and determined the dominant semantic class, which has the maximum number of points, of that voxel through the histogram. The semantic map and the corresponding metric map of the scene are illustrated in Fig. 5(c) and Fig. 5(d), respectively. In order to merge the current and global metric maps, we first determined the boundary voxels of both maps. Then, we considered the positions and orientations of boundary voxels. Lastly, we found neighbor voxels and merged the maps.

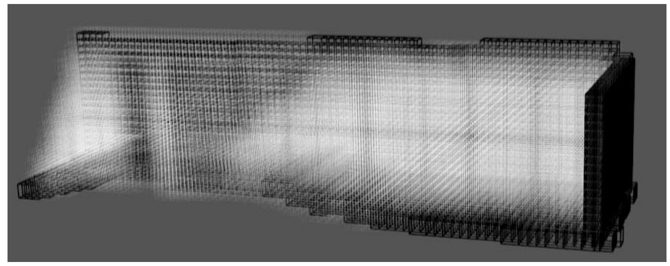
III. EXPERIMENTAL WORKS

A. Experimental Setup

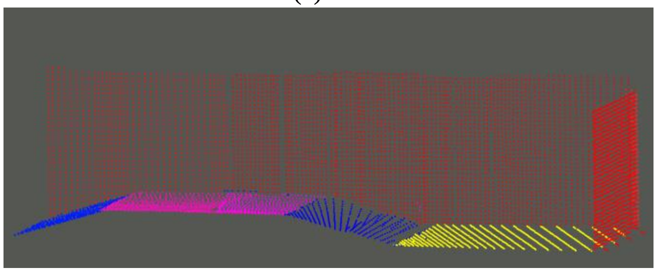
We used ROS and Gazebo to conduct the experiments. First, we modeled ESOGU Artificial Intelligence & Robotic Laboratory Search and Rescue Test Arena in Gazebo simulation environment (Fig. 6(a)). The dimensions of environment are 6 x 4 meters. Then, we utilized hector_nist_arenas_gazebo ROS package [33] to insert ramps in the environment (Fig. 6(b)). A Pioneer 3-AT mobile robot was launched in the modeled environment with an Asus XTion Pro RGB-D sensor to capture point cloud data. We used teleop_twist_keyboard ROS package [34] for the teleoperation of the mobile robot.



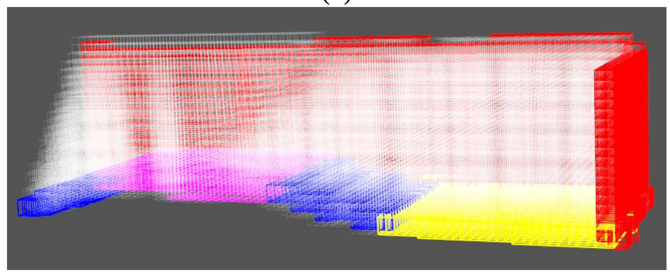
(a)



(b)



(c)

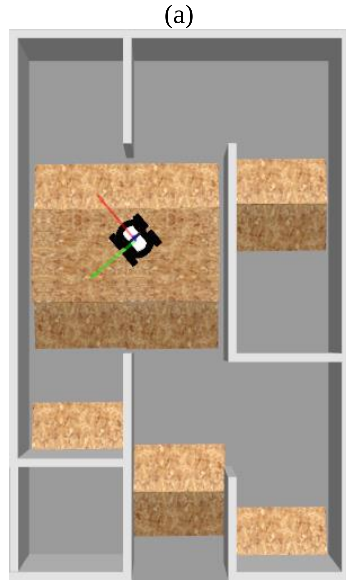


(d)

Figure. 5. An example for construction of metric map. (a) Free (white) and occupied (black) points taken from the RTAB-Map. (b) Corresponding metric map of (a). (c) Semantic map of the scene. (d) Corresponding metric map of (c).

After each predetermined time interval (TimeInterval), point cloud data was gathered with RTAB-Map ROS package while the robot operates in the simulation environment. The TimeInterval parameter is selected 1 second in this experiment. Also, GridCellSize and MaxNodes parameters of RTAB-Map ROS package are determined as 0.025 and 1, respectively. A preprocessing step must be applied to the point cloud data for the scene classification problem of point-based deep learning architectures. Therefore, the scene is divided into blocks instead of using the entire scene to avoid losing data and detect local features. As a result, we separated a point cloud data into 1 m^2 blocks in the xy plane independent from the points' z coordinates. Deep learning architectures accept a fixed number of points. In this study, we specified the number of points in a block as 4096. We applied random upsampling or downsampling to the blocks that contain less than or greater than 4096 points, respectively. Besides, we removed the blocks that have less than 500 points. DGCNN architecture can receive coordinates, normalized coordinates, and color information of points as an input. In this study, we did not use color information, and points were presented with 6D features (x, y, and z coordinates and normalized x, y, and z coordinates). We used the default parameters for scene segmentation of DGCNN architecture.

(a)



(b)

Figure. 6. (a) ESOGU AIRLAB search and rescue test arena, (b) Gazebo model of the test arena.

The DGCNN architecture is implemented in Python programming language using TensorFlow library [35]. However, the remaining parts of the method are realized with C++ programming language. For that reason, we used pybind11 wrapper [36]. After the points belong to each class were determined, we clustered the points depending on the class labels. Besides, we applied RANSAC segmentation method to segment different planar surfaces. The DistanceThreshold parameter of RANSAC is selected 0.02 meters. We used octree_viewer module [37] of the Point Cloud Library (PCL) [38] to visualize the metric map. In this step, VSIZE parameter is chosen 0.05 meters. In order to generate semantic and metric maps of the environment given in Fig. 6(b), the robot navigated for 160 seconds. Therefore, 160 scenes were captured during that process. The experiments were carried out on a PC with Intel i7 processor with 2.8 Ghz, 16 GB RAM, and operating system Ubuntu 20.04.

B. Experimental Results

The experimental results are shown in Fig. 7. In the figure, the left and right columns illustrate the metric and semantic map of the environment, respectively. The rows of the figure represent the results at some steps. The mapping process was completed at 160 steps, and we preferred to give results at 40, 80, 120, and 160 steps. In the figure, red, yellow, purple, and blue colors represent the wall, terrain, straight, and inclined ramp classes, respectively. We did not visualize points and voxels belong to the free semantic class to clarify figures. As seen from the results, our semantic and metric maps grow incrementally. In this way, the computational complexity of the proposed method was decreased.

The results for semantic maps indicate that the proposed method successfully classifies walls, terrain, and ramps for each scene although the DGCNN model was trained with data gathered in a different simulated environment. Besides, the proposed method calculates and stores properties such as orientation, maximum, and minimum coordinates of each planar surface even though we did not visualize these properties. Then, the method utilizes these properties to merge the current scene and the global semantic map. The experimental results indicate that our method successfully merges the maps to generate an accurate semantic map. In our method, the success of producing a metric map of a scene highly relies on the semantic map's accuracy. As seen from the figures, the metric map of the environment is generated successfully since the semantic map is accurately created. Then, the current and global maps are integrated carefully with determining boundary voxels. The experimental results for metric maps show that the proposed method merges the maps successfully.

IV. CONCLUSIONS AND FUTURE WORKS

This study aims to create semantic and metric maps of an environment similar to standard NIST search and rescue test arenas. To do that, we utilized RTAB-Map ROS package and DGCNN architecture. The proposed method grows semantic and metric maps incrementally to decrease the computational complexity. Besides, we prefer to use point cloud data instead of visual data, which many previous studies employed, since point cloud data is more suitable for post-disaster environments. In contrast to previous studies that address producing 3D metric maps, we classified voxels not only occupied and free but also walls, terrain, and ramps. The experimental results indicate that our method successfully generates accurate semantic and metric maps. For future works, we plan to develop a new metric map approach and determine free and occupied voxels without using RTAB-Map since the computational complexity of OctoMap that integrated into RTAP-Map is not appropriate to produce fine-detailed maps. Besides, we aim to create a topological map utilizing the metric map. Thus, the robot can efficiently navigate the environment by preparing the shortest path plan using the topological map.

ACKNOWLEDGMENT

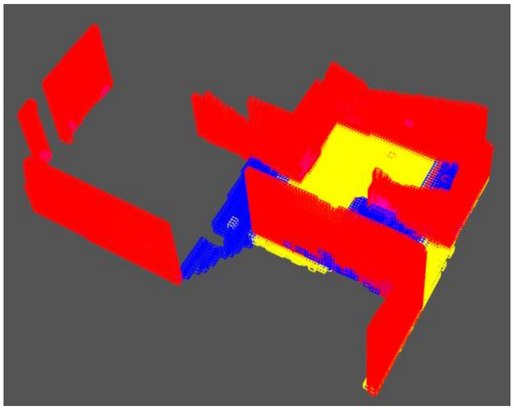
This paper is supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) 2209-B Industrial Oriented License Graduate Thesis Support Program.

CONFLICTS OF INTEREST

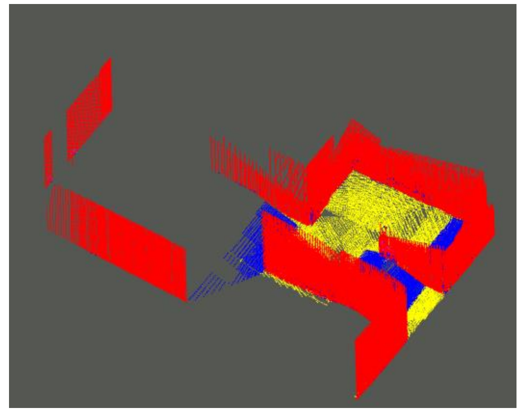
There is no conflict of interest.

RESEARCH AND PUBLICATION ETHICS

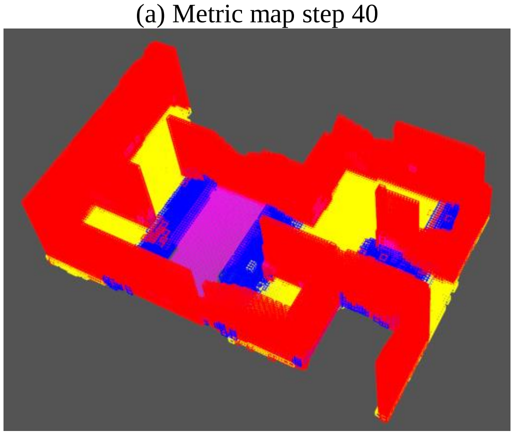
The authors declare that this article does not require ethics committee approval or any special permission.



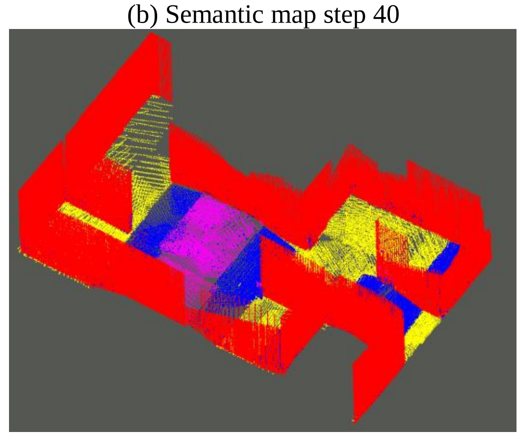
(a) Metric map step 40



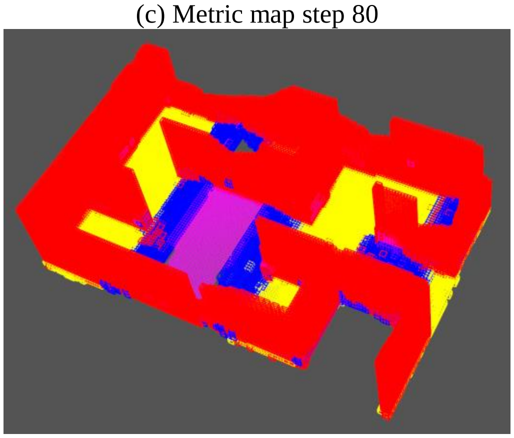
(b) Semantic map step 40



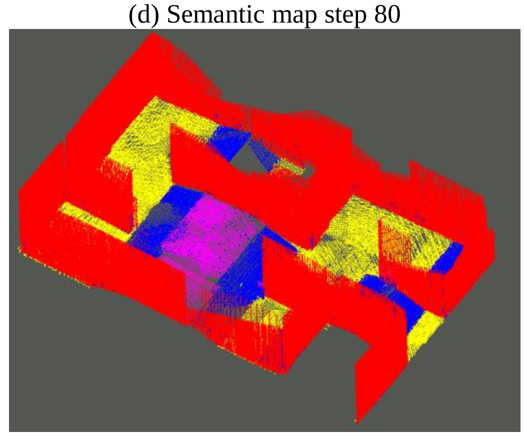
(c) Metric map step 80



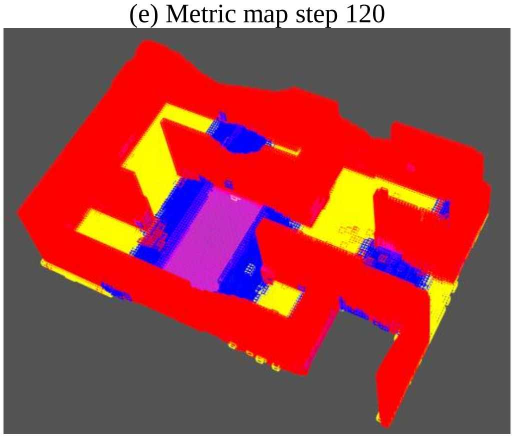
(d) Semantic map step 80



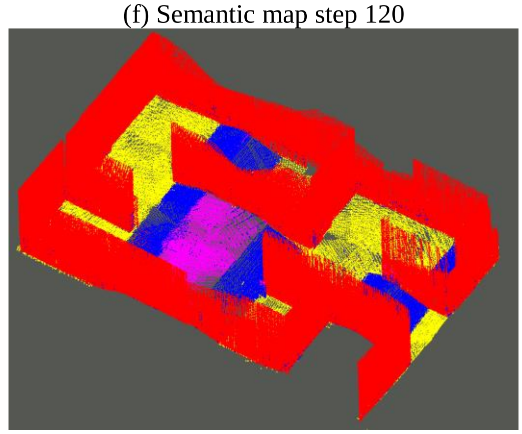
(e) Metric map step 120



(f) Semantic map step 120



(g) Metric map step 160



(h) Semantic map step 160

Figure. 7. Experimental results.

REFERENCES

[1] H. Kitano and S. Tadokoro, "RoboCup Rescue: A Grand Challenge for Multiagent and Intelligent Systems," *AI Magazine*, vol. 22, no. 1, pp. 39-52, 2001.

[2] A. Jacoff, E. Messina, B. A. Weiss, S. Tadokoro, and Y. Nakagawa, "Test arenas and performance metrics for urban search and rescue robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2003.

[3] H. L. Akin, N. Ito, A. Jacoff, A. Kleiner, J. Pellenz, and A. Visser, "Robocup rescue robot and simulation leagues". *AI magazine*, vol. 34, no. 1, pp. 78-86, 2013.

[4] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. A. Ziparo, "Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue", *Journal of Field Robotics*, vol. 24, no. 11, pp. 943-967, 2007.

[5] S. Balakirsky, S. Carpin, and A. Visser, "Evaluating the robocup 2009 virtual robot rescue competition.", *In Proceedings of the 9th Workshop on Performance Metrics for Intelligent Systems*, pp. 109-114, 2009.

[6] F. Amigoni, A. Visser, and M. Tsushima, "RoboCup 2012 Rescue Simulation League Winners.", *In Robocup 2012: Robot Soccer World Cup XVI,Lecture Notes in Artificial Intelligence*, Berlin, Springer, 2013.

[7] R. Sheh, S. Schwertfeger, A. Visser, "16 Years of RoboCup Rescue," *Künstliche Intelligenz*, vol. 30, no.3, pp. 267-277, 2016.

[8] ROS, Available: https://www.ros.org/, Accessed Time: 15.02.2021

[9] GazeboSim, Available: http://gazebosim.org/, Accessed Time: 15.02.2021

[10] K. Ito, K. Yokochi, and Y. Tanaka, "Chukyo Rescue A Team Description Paper for Robocup 2017 Virtual Robot League.", 2017.

[11] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters.", *IEEE transactions on Robotics*, vol. 23 no. 1, pp. 34-46, 2007.

[12] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation.", *IEEE international symposium on safety, security, and rescue robotics*, pp.155-160, 2011.

[13] S. Yavuz, M. F. Amasyalı, E. Uslu, F. Cakmak, N. Altuntas, S. Marangoz, M. B. Dilaver, A. E. Kırlı, "YILDIZ Team Description Paper for Rescue Simulation League Virtual Robots Competition 2017", 2017.

[14] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees.", *Autonomous robots*, vol. 34 no. 3, pp. 189-206, 2013.

[15] S. M. Ahmadi, M. H. G. Nejad, and A. Torabian, "SOS RS Team Description Paper RoboCup 2018 Rescue Virtual Robot League", 2018.

[16] M. Montemerlo, S. Thrun, D. Koller and B.Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem", *Association for the Advancement of Artificial Intelligence (AAAI) Conference*, pp. 593-598, 2002.

[17] J. Ziegler, J. Gleichhauf and C. Pfitzner, "RoboCup Rescue 2018 Team Description Paper AutonOHM", 2018.

[18] S. May, P. Koch, R. Koch, C. Merkl, C. Pfitzner, and A. Nuechter, "A Generalized 2D and 3D Multi-Sensor Data Integration Approach based on Signed Distance Functions for Multi-Modal Robotic Mapping.", *In Vision, Modeling, and Visualization (VMV)*, pp. 95-102, 2014.

[19] J. H. Kim, X. Lin, N. Kanyok, A. Shaker, P. Poudel, I. Cardenas, N. Karina, C. Cabrera, H. Jeong, and G. P. Sharma, "RoboCup Rescue 2019 TDP Virtual Robot Simulation." 2019.

[20] M. Labbé and F. Michaud, "Memory management for real-time appearance-based loop closure detection.", *IEEE/RSJ international conference on intelligent robots and systems*, pp. 1271-1276, 2011.

[21] rtabmap_ros, Available: http://wiki.ros.org/rtabmap_ros. Accessed Time: 15.02.2021.

[22] T. Rabbani, F. A. Van Den Heuvel, G. Vosselman, "Segmentation of point clouds using smoothness constraints," ISPRS commission V symposium: image engineering and vision metrology. *International Society for Photogrammetry and Remote Sensing (ISPRS)*, Dresden, Germany, 2006.

[23] M. A. Fischler, R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

[24] W. Deng, K. Huang, X. Chen, Z. Zhou, C. Shi, R. Guo, and H. Zhang,)., "Semantic RGB-D SLAM for Rescue Robot Navigation.", *IEEE Access*, vol. 8, pp.221320-221329, 2020.

[25] K. Turgut and B. Kaleci, "Comparison of Deep Learning Techniques for Semantic Classification of Ramps in Search and Rescue Arenas," *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Istanbul, Turkey, pp. 1-6, 2020.

[26] C. R. Qi, H. Su, M. Kaichun, L. J. Guibas, ''PointNet: Deep learning on point sets for 3D classification and segmentation,'' *in Proc. IEEE Conf. Comput. Vision. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017.

[27] C. R. Qi, L. Yi, H. Su, L. J. Guibas, ''PointNet++: Deep hierarchical feature learning on point sets in a metric space,'' *Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017.

[28] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, "PointCNN: Convolution on x-transformed points," *Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2018.

[29] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, Article 146, 12 pages, 2019.

[30] Point Cloud Library Plane Segmentation Code. Available: https://pointclouds.org/documentation/tutorials/random_sample_consensus.html, Access date: 15.02.2021.

[31] S. Thrun, "Learning Occupancy Grid Maps with Forward Sensor Models". *Autonomous Robots*, vol. 15, pp. 111–127, 2003.

[32] A. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto, "Octree-based region growing for point cloud segmentation", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88-100, 2015.

[33] Hector_nist_arenas_gazebo, Available: http://wiki.ros.org/hector_nist_arenas_gazebo, Accessed Time: 15.02.2021

[34] teleop_twist_keyboard, Available: http://wiki.ros.org/teleop_twist_keyboard, Accessed Time: 15.02.2021

[35] Tensorflow, Available: https://www.tensorflow.org/, Accessed Time: 15.02.2021

[36] pybind11, Available: https://github.com/pybind/pybind11, Accessed Time: 15.02.2021

[37] octree_viewer, Available: https://github.com/PointCloudLibrary/pcl/blob/master/tools/octree_viewer.cpp, Accessed Time: 15.02.2021

[38] R. B. Rusu, S. Cousins, "3D is here: Point Cloud Library (PCL)," *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.