Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı, S. 466-472, Eylül 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue, pp. 466-472, September 2020

Research Article

ORB-SLAM-based 2D Reconstruction of Environment for Indoor Autonomous Navigation of UAVs

Abdullah Yusefi[1*], Akif Durdu[2] and Cemil Sungur[2]

[1] Computer Engineering Department, Konya Technical University, Turkey
[2] Electrical-Electronics Engineering Department, Konya Technical University, Turkey

(1st International Conference on Computer, Electrical and Electronic Sciences ICCEES 2020 – 8-10 October 2020)

(**DOI:** 10.31590/ejosat.819620)

ATIF/REFERENCE: Yusefi, A., Durdu, A. & Sungur, C. (2020). ORB-SLAM-based 2D Reconstruction of Environment for Indoor Autonomous Navigation of UAVs. *European Journal of Science and Technology*, (Special Issue), 466-472.

Abstract

In this paper, a simple and economic yet efficient autonomous mapping and navigation system for unmanned aerial vehicles is presented. In order to realize this system, three modules have been implemented. First module constructs a 3D model of the environment while autonomously navigating the drone and is based on one of the top monocular SLAM algorithms called ORB-SLAM. For the autonomous navigation of the system a visual-based line tracking method is proposed. Afterwards, the second module performs a real time transformation of the 3D map to 2D grid map. While most of the 3D to 2D map conversion studies use octomaps in the middle of two, we present a threshold-based method that directly converts the 3D map to 2D without need for any middle component. Finally, third module uses A* path planning algorithm to navigate the drone to the goal pose in the constructed 2D grid map. This module uses only IMU-aided Adaptive Monte Carlo localization (AMCL) combined with monocular camera information to complete this task. The experimentation results indicate that the proposed system is adequately efficient to be used in the low-cost drones that have only a monocular camera and limited processing resources on them.

Keywords: ORB-SLAM, Line-Tracking, Map Conversion, AMCL, Autonomous Navigation

1. Introduction

Obtaining a physical model and mapping the environment is one of the most important and fundamental issues in robotics and has many applications, such as autonomous robotic navigation, tracking and detecting objects and people, and search and rescue operations. Significant progress has been made over the last ten years in the field of autonomous robot guidance, in particular the Simultaneous Localization and Mapping (SLAM) of the environment.

In 1985, one of the first attempts to perform SLAM using a laser scanner sensor and an encoder shaft was made by Chatila et al. considering the uncertainties in the problem [5]. In 1990, Smith and colleagues first mapped using environmental characteristics and used the Kalman filter to solve the problem [6].

Over the past decade, many researchers have focused on finding suitable solutions for simultaneous localization and mapping in real time. Among these, the most popular methods for locating and mapping at the same time have been the Extended Kalman filter [7] and the Rao-Blackwellized particle filter [8]. The Extended Kalman filter uses a linear approximation, assuming that the density function has a Gaussian probability distribution. However, a number of studies have shown that the map obtained in this method is not very accurate due to the error caused by linearization and the computational complexity of this method is high and is not applicable in real time. Much research has been undertaken to address these issues.

* Corresponding Author: Konya Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Konya, Türkiye, ORCID: 0000-0001-7557-8526, e168129001005@ktun.edu.tr

In an effort to increase the accuracy of the simultaneous location and mapping, a new Kalman filter was introduced called the Unscented Kalman Filter, which did not use linear approximations [9]. The Unscented Kalman Filter uses a definitive sampling method to calculate mean and covariance estimates. The Unscented Kalman filter, instead of the nonlinear function of the system, estimates the probability density function and the higher the nonlinearity of the system, the more reliable the estimation than the Extended Kalman filter, but its computational complexity is similar to that of the Extended Kalman filter. On the other hand, the Fast Simultaneous Localization and Mapping Algorithm (FastSLAM) uses the Particle Filter and the Extended Kalman Filter simultaneously to greatly reduce the computational volume compared to the Extended Kalman Filter and the Unscented Kalman Filter [10]. Finally, the Unscented Kalman Filter-based FastSLAM algorithm is called Unscented FastSLAM as one of the new SLAM methods, using accurate Unscented Kalman Filter estimates to improve precision and minimize computational time [11]. It has been shown that Unscented FastSLAM has a better performance in terms of accuracy and quality of mapping compared to FastSLAM methods based on Extended Kalman filter [12]. Despite this advantage, there are still many untapped potentials for FastSLAM algorithms.

In addition to the above filter-based SLAM methods, several other methods with different meanings for the results have also been proposed in recent years. The global optimization method, focused on conservation of some key frames within the setting and bundle adjustment in order to estimate movement. This is a common approach at present for SLAMs based on vision such as ORB-SLAM. ORB-SLAM is a flexible, precise SLAM solution capable of retrieving the camera trajectory and a fragmented 3D scene reconstruction from small handheld devices to a vehicle that moves across multiple urban blocks in real-time. It is capable of closing large loops and relocalizing itself globally in real time.

This paper presents a simple and economical yet efficient autonomous mapping and navigation system for unmanned aerial vehicles. Three modules have been implemented in order to realize this system. The first module builds an environment 3D model while autonomously navigating the drone and is based on one of the top monocular SLAM algorithms called ORB SLAM2. A visual-based line tracking method is proposed for the autonomous navigation of the system. The second module then performs a real-time transformation of the 3D map to the 2D grid map.

While most 3D to 2D map conversion studies use octomaps in the middle of two, we present a threshold-based method that directly converts the 3D map to 2D without the need for any middle component. Finally, the third module uses the A* path planning algorithm to navigate the drone to the goal pose in the built 2D grid map. This module only uses the IMU-aided Adaptive Monte Carlo (AMCL) localization to complete this task.

There are many research findings in the literature in the case of the generation of the grid map and the identification of obstacles from a point cloud. Goeddel et al.'s latest work [14] implemented a 3D LiDAR data extraction method for localization performing a 2D map. Huesman [15] proposed a point cloud into a 2D occupants map conversion with the simple concept of using slope thresholding to assess obstacles.

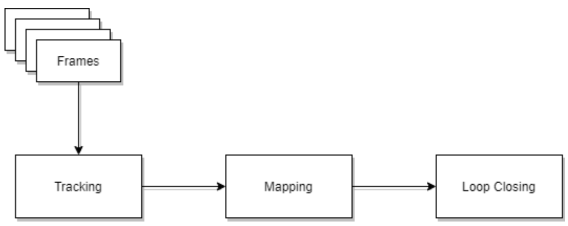
Beutel et al. [16] used a somewhat different approach based on nueral neighbor interpolation for making maps for both 2D and 3D. The ROS grid mapping library [17] also offers features to create a variety of different types of 2D grid maps from different input sources. Thrun's approach to learning [18] used maximization of expectations to produce occupancy maps from sensor data directly.

In comparison to the aforementioned techniques, which use only 3D data, Santana et al. [19] took advantage of image data by separating the scene into floor and non-floor areas by colour-based visual segmentation. Another way of resolving this issue is by using point cloud to laser scanning [20], and then by using the resulting data as input to an algorithm for mapping like the ROS Gmapping [21] that can generate occupancy maps on LiDAR scans. But the mapping algorithm still considers the LiDAR data to be noisy, and we know that the point cloud created by ORB-SLAM contains only accurate data. This approach may generate additional uncertainty in the map.

The paper is organized as follows. The first section concerns the introduction and brief review of the types of SLAM specifically ORB-SLAM. This is further discussed in Section II. Section III describes the line tracking-based navigation and mapping of the system and section IV presents the transformation method of 3D to 2D grid map. Section V concerns the general architecture of the proposed navigation system the experimental results are described in section VI. A final discussion and conclusions are presented in section VII.

2. ORB-SLAM

ORB-SLAM is a robust visual SLAM method and benefits from the speed and rotation invariance advantage, which allows the features to be extracted from images in real time. It is designed for real time processing and is based on a visual search of ORB features in the image surface space which allows for more efficient detection and tracking of features [1, 13]. Fig. 1 shows an overview of the data flow in ORB-SLAM.



The three main threads, Tracking, Local Mapping and Loop Closing, are the basis of this algorithm that obtain the frames sequentially and operate in parallel for camera motion estimation and environment mapping.

The tracking thread localizes the camera and determines when a new keyframe is inserted. Features correlate with the previous frame and the position is optimized by motion-based bundle adjustment. FAST corners are detected as features and are described by the ORB. The local mapping thread uses the concept of covisibility graph of keyframes to obtain a local visible map. The ORB features are triangulated and matched in connected keyframes in the covisibility graph. If the matched map point is found in more than 25% of the frames in which it is predicted to be visible and is observed by at least three keyframes, it will be added to the local map. The loop closing thread uses a bag of words principle to identify possible loops within the system and to adapt the global optimization. It searches the bag of words [2] in the covisibitlity graph of the current keyframe and its vicinity. If three clear loop candidates are successively found, this loop is known to be a serious candidate. Afterwards, a series of optimizations and a RANSAC are applied to these loop candidates to remove the noise and accept one if necessary. In order to handle the scale and final optimization the pose and map points of current keyframe and its neighbours are corrected and fused respectively.

Fig. 1 Overall Data Flow of ORB-SLAM

3. Line Tracking-Based Navigation and Mapping

Ubuntu 16.04 operating system equipped with ROS Kinetic is the software platform used in this study. ROS makes it easy for robotic researchers to develop and deploy autonomous robots that follow their own programmed algorithms, or those of a human operator. The C++ and python language programming are supported [3] by the implementation of complex, highly-scalable and modular packages for robotic behaviour. In addition, Gazebo simulation software allows for fast-testing algorithms, design robots, regression tests and the use of practical scenarios to train artificial intelligence.

For autonomous navigation and mapping of the environment an architecture for house simulation has been developed. The lines are used to differentiate the rooms or corridors (Fig. 2). A pure line follower's algorithm is most likely to result in unpredictable outcomes, as the robot has to turn when faced with rotation points or endpoints of lines. To address this problem, a right-oriented line follower algorithm was developed and robot motions were provided in this framework.

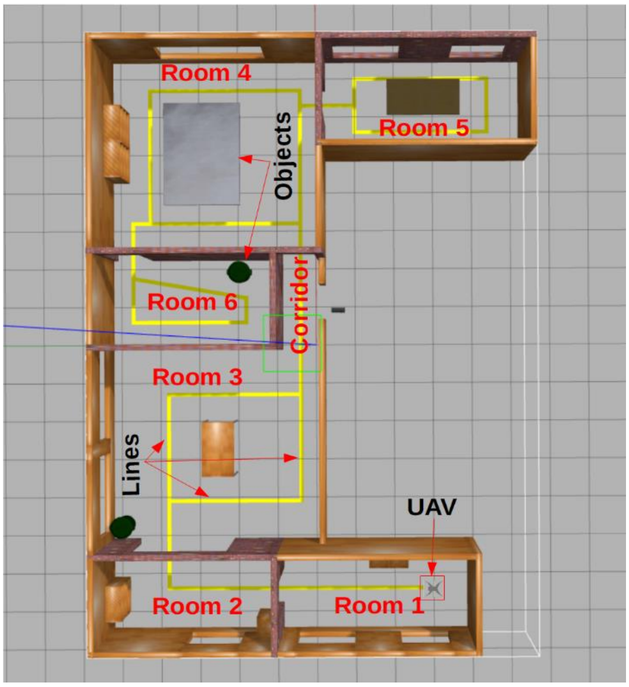


Fig. 2 View of the environment and tracking lines

The Dji Tello was implemented in a simulated indoor environment for collecting training data and experimental testing. The Dji Tello is a low-cost UAV with a top velocity up to 8 m / s and a maximum height of 30 m. In addition, it weighs around 80 g and is fitted with a 5MP camera capable of capturing pictures and videos of 2592 x 1936 and HD 7200P30 quality, respectively [101]. Such capabilities allow UAV robotic researchers a suitable forum for testing their visual based algorithms.

The robot takes a picture of the environment (Fig 3. a) from a camera so that the navigation path of the robot can be calculated using image processing algorithms with these images. Sequentially processing the pictures taken from the robot camera, multiple types of frames, including HSV and masked are made. Segmented images are used for the foreground evaluation. The photos would then be transferred to HSV files. The purpose of this transformation is to provide a more precise outcome in the image assessment, as HSV images are more stable with respect to light, shadow effects, etc. In [4] the left-oriented autonomous navigation of the robot was suggested to take a new approach in a factory-like environment which inspires our process. Here, when the HSV transformation was applied to the picture as the evaluation of the picture as a whole will also be complicated and would increase the time for computation. The image is split in four sub-regions: a1, a2, a3 in the process (Fig. 3 a). The a1 and a3 pieces are multiplied by 0 to exclude areas of irrelevance for the proper orientation such as up, down and left. The area a2 is left alone and decision-making in this area is made upon it. It just removes the regions to be tracked from the entire image and leads the robot in the true direction (Fig. 3. b, c).

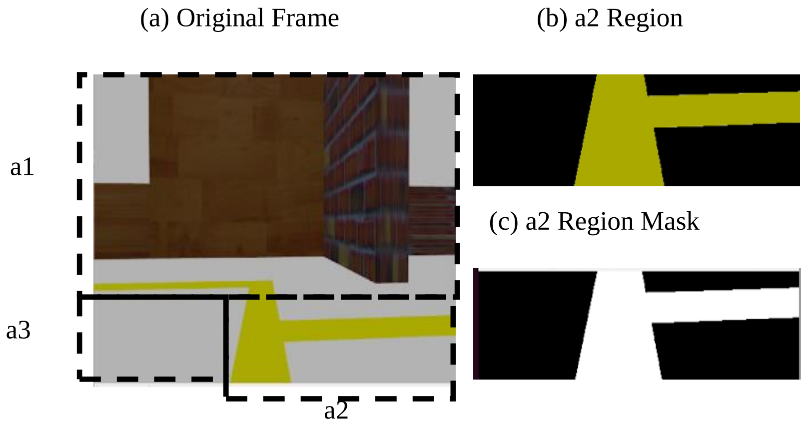


Fig. 3 Camera View of Robot

According to the algorithm provided, we have controlled the robot in three directions: forward and right and reverse.

1. The robot continues forward at 5 mph speed and height of 1 meters and matches the middle of the line without any orientation decision.
2. The robot turns 90 degrees right while making a decision on right-orientation. The forward speed is then 0 mph, with the robot turning 90 degrees to the right.
3. The robot sets the forward speed to 0 mph and turns 180 degrees if it reaches the endpoint.

In order to obtain the 3D map, the ORB-SLAM runs in parallel while navigating the environment. It creates a map of the point clouds built by the ORB features detected and therefore, the performance of the mapping depends on the high-texture structure of the environment.

4. 3D to 2D Grid Map Conversion

Point clouds in robotic applications are a growing form of data that allow the robot to see the world. Unfortunately, it is somewhat restricted in usage for autonomous navigation. There is just so much data to handle and determine the direction of a robot, in particular for a low-cost drone with limited processing resources. This section proposes a method that receives a 3D point cloud data set and converts it to a 2D occupancy grid, a much more popular browsing / path algorithm data type. The algorithm can convert point clouds from any source, which in the case of this study is the ORB-SLAM. Since the ORB-SLAM builds the map incrementally in real-time, the point cloud map is partially updated in a particular environment. Accordingly, the 3D to 2D map conversion module also is capable of processing the partially obtained point clouds and converting them to 2D grid maps in real-time.

In order to process the point clouds obtained from ORB-SLAM in real-time, we take the keyframe into account at each step. When a keyframe is received along with its camera position and all the map points visible to the keyframe. Since the map point clouds are in 3D pose (x, y, z) axis, we convert them to 2D pose by removing the height axes and only take the horizontal axis into account. Afterwards, a threshold based method is applied to the map point along the camera pose and each map point. If the number of map points along these two are above a threshold then that map point is considered to be occupied. Since the 2D grid map is represented by 0s and 1s, indicating free cells and occupied cells respectively, the whole map points in a keyframe is represented either by 1 or 0 (Eq. 1). There, $pc_{c,m}$ is the number of point clouds along the line between camera and current map point and δ represents the threshold size. The grid cells with no map points are also considered to be free. In order to handle the loop closing and scale variation, scale factor is defined and after every loop detection a recalculation of the whole map points is done correspondingly.

$$Grid\_Cell_{i,j} = \begin{cases} 1, & pc_{c,m} \geq \delta \\ 0, & pc_{c,m} < \delta \end{cases} \quad (1)$$

5. Autonomous Navigation

In this module the algorithm A* is used to guide the drone towards the target location in the 2D grid map. To perish this function, this module uses only the Adaptive Monte-Carlo (AMCL) location supported by IMU in combination with monocular camera information.

As can be noted in Fig. 5, the system consists of three main modules, mapping, conversion of 3D-to-2D maps and autonomous navigation. Parts III and IV explained the first and second modules.

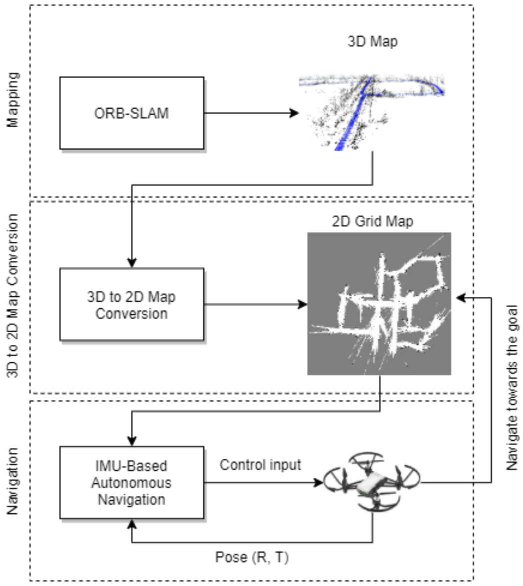


Fig. 4 The general architecture of the system

In the third module, a low-cost drone is equipped with an Inertial Measurement Unit (IMU) and a monocular camera in order to autonomously navigate in the obtained 2D grid map. The IMU is fused with the Adaptive Monte Carlo Localization (AMCL) algorithm to localize itself in the static map and the monocular camera helps the drone to avoid the collision to the static or dynamic obstacles.

6. Results

For this analysis, an indoor environment was developed to execute SLAM independently and to evaluate the algorithm's effectiveness. For this reason, the UAV is requested to start at the point indicated in the map and to follow the line in accordance with the algorithm provided. During the autonomous run, the robot has performed ORB-SLAM and 3D to 2D grid map conversion simultaneously to obtain a light map of the environment. Finally, the low-cost UAV is autonomously navigated using IMU-aided AMCL and camera in the resulting map to demonstrate the efficiency of the 2D grid map.

Fig. 5 displays resulting 2D grid map for our prepared environment and the Kitti data set. Fig. 5 a displays the corresponding ORB-SLAM point map cloud of the corresponding house simulation environment. Fig. 5. b shows the resulting 2D grid diagram of the Kitti dataset. The accuracy of the resulting map depends heavily on the texture and illumination of frames, since all of these are using the monocular ORB-SLAM to create a 3D point cloud map. Therefore, the resulting map in here would be better in well-structured environments particularly in simulation world.

(a) House (b) Kitti Dataset

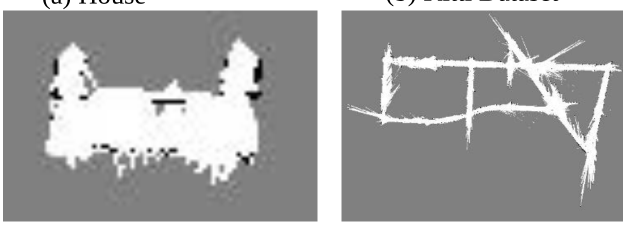


Fig. 5 Resulting 2D grid map of (a) House simulation environment and (b) Kitti dataset

After converting the 3D point cloud map and obtaining the 2D grid map it is possible to set a target for the robot to autonomously navigate in the environment. Fig. 6 displays the low-cost drone autonomously navigating in the obtained 2D grid map of the testing simulation environment.

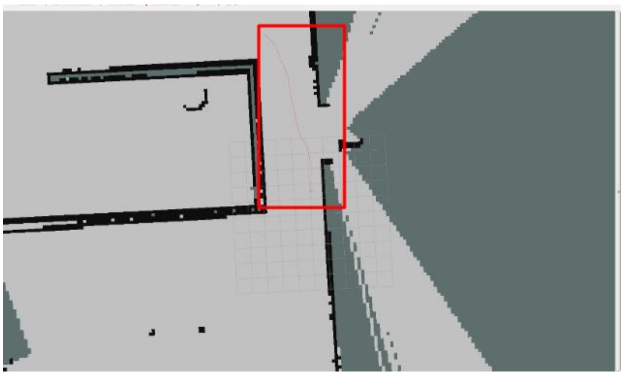


Fig. 5 Drone autonomously navigating in the map

7. Conclusion and Future Works

Within this paper, an autonomous mapping and navigation system for unmanned air vehicles is described as a simple, economical and yet effective method. Three modules have been introduced to execute this system. The first module builds a 3D environmental model when controlling the drone autonomously and is based on an ORB-SLAM monocular algorithm. A visual line tracking approach is suggested for autonomous vehicle navigation. The second module then converts the 3D map into the 2D grid map in real time. Although most 3D to 2D map conversion studies use octomaps in the middle of two, we present a threshold-based approach that directly converts the 3D map to 2D without the need for any middle part. Finally, the third module uses the A* path planning algorithm to guide the drone to the target pose in the constructed 2D grid map. This module uses only the IMU-aided Adaptive Monte Carlo position (AMCL) combined with the information of a single camera to complete this mission. The results of the experiments show that the proposed device is sufficiently powerful to be used in low-cost drones with only a monocular camera and minimal processing resources. The future work can involve the improvement of the autonomous navigation by fusion of the camera and the IMU in the localization part. Moreover, alternative and novel technologies such as deep learning might be able to enhance the capabilities of the system.

References

[1] Mur-Artal, Raul, and Juan D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

[2] Gálvez-López, Dorian, and Juan D. Tardos. "Bags of binary words for fast place recognition in image sequences." IEEE Transactions on Robotics 28.5 (2012): 1188-1197.

[3] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

[4] Durdu, Akif, and Mehmet Korkmaz. "Autonomously simultaneous localization and mapping based on line tracking in a factory-like environment." Advances in Electrical and Electronic Engineering 17.1 (2019): 45-53.

[5] Chatila, Raja, and Jean-Paul Laumond. "Position referencing and consistent world modeling for mobile robots." Proceedings. 1985 IEEE International Conference on Robotics and Automation. Vol. 2. IEEE, 1985.

[6] Harris, Christopher G., and J. M. Pike. "3D positional integration from image sequences." Image and Vision Computing 6.2 (1988): 87-90.

[7] Smith, Randall, Matthew Self, and Peter Cheeseman. "Estimating uncertain spatial relationships in robotics." Autonomous robot vehicles. Springer, New York, NY, 1990. 167-193.

[8] Doucet, Arnaud, et al. "Rao-Blackwellised particle filtering for dynamic Bayesian networks." arXiv preprint arXiv:1301.3853 (2013).

[9] Martinez-Cantin, Ruben, and José A. Castellanos. "Unscented SLAM for large-scale outdoor environments." 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2005.

[10] Montemerlo, M. "A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association." Ph. D. thesis, Carnegie Mellon University (2003).

[11] Kim, Chanki, Rathinasamy Sakthivel, and Wan Kyun Chung. "Unscented FastSLAM: a robust and efficient solution to the SLAM problem." IEEE Transactions on robotics 24.4 (2008): 808-820.

[12] Kurt-Yavuz, Zeyneb, and Sirma Yavuz. "A comparison of EKF, UKF, FastSLAM2. 0, and UKF-based FastSLAM algorithms." 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES). IEEE, 2012.

[13]Mur-Artal, Raul, and Juan D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

[14]Goeddel, Robert, et al. "FLAT2D: Fast localization from approximate transformation into 2D." 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016.

[15]Huesman, Jacob. "Converting 3D Point Cloud Data into 2D Occupancy Grids suitable for Robot Applications." NDSU EXPLORE: Undergraduate Excellence in Research and Scholarly Activity (2015).

[16]Beutel, Alex, Thomas Mølhave, and Pankaj K. Agarwal. "Natural neighbor interpolation based grid DEM construction using a GPU." Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2010.

[17]Fankhauser, Péter, and Marco Hutter. "A universal grid map library: Implementation and use case for rough terrain navigation." Robot Operating System (ROS). Springer, Cham, 2016. 99-120.

[18]Thrun, Sebastian. "Learning occupancy grid maps with forward sensor models." Autonomous robots 15.2 (2003): 111-127.