REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DYNAMIC ENVIRONMENT AWARE AUTONOMOUS MOBILE ROBOT NAVIGATION

Mustafa Burak DİLAVER

MASTER OF SCIENCE THESIS
Department of Computer Engineering
Program of Computer Engineering

Advisor
Asst. Prof. Erkan USLU

July, 2020

REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DYNAMIC ENVIRONMENT AWARE AUTONOMOUS MOBILE ROBOT NAVIGATION

A thesis submitted by Mustafa Burak DİLAVER in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 21.07.2020 in Department of Computer Engineering, Program of Computer Engineering.

Asst. Prof. Erkan USLU
Yildiz Technical University
Advisor

Approved By the Examining Committee

Asst. Prof. Erkan USLU, Advisor
Yildiz Technical University ______________________

Assoc. Prof. Sırma YAVUZ, Member
Yildiz Technical University ______________________

Asst. Prof. Gökhan İNCE, Member
Istanbul Technical University ______________________

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of Dynamic Environment Aware Autonomous Mobile Robot Navigation supervised by my supervisor, Asst. Prof. Erkan USLU. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Mustafa Burak DİLAVER

Signature

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) Grant No: ARDEB 1001 118E215

This study was supported by the Yildiz Techical University Scientific Research Projects Coordination Unit (YTU-BAPK) Grant No: FDK-2017-3044

Dedicated to my family

ACKNOWLEDGEMENTS

First and foremost, my deep gratitude goes to my family, for their continuous and uncountable support.

I would like to express the deepest appreciation to Asst. Prof. Erkan USLU, who has been my supervisor for this and many other previous studies, and has shown his ultimate support throughout my educational career. His encouragement and positive attitude towards all the work I've done, whether its good enough or not, makes him unique.

Besides my advisor, my sincere thanks goes to Lect. Furkan ÇAKMAK who has been one of my greatest supporters. Without his tremendous support and advice I would not be where I am now. I'll always envy his dynamism and persistent work.

I would like to thank Assoc. Prof. Sırma YAVUZ and Assoc. Prof. M. Fatih AMASYALI for their best efforts to make Yildiz Technical University Probabilistic Robotics Group a fabulous place for the students interested in robotics for more than a decade. Without their passion and tireless work this group couldn't have become its current perfection.

I thank all of the great people working as both academic and administrative staff in Yildiz Technical University's Computer Engineering Department, where I've spent 7 years of my life.

Finally, I would like to thank all of my friends who have shared tiny bits of their precious lives with me and supported me with their wonderful friendship.

Mustafa Burak DİLAVER

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

A*	A-star
DARPA	The Defense Advanced Research Projects Agency
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LIDAR	Light Detection and Ranging
LPC	Lee Position Controller
MAV	Micro Aerial Vehicle
RADAR	Radio Detection and Ranging
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle
VSLAM	Visual Simultaneous Localization and Mapping

LIST OF FIGURES

ABSTRACT

Dynamic Environment Aware Autonomous Mobile Robot Navigation

Mustafa Burak DİLAVER

Department of Computer Engineering
Master of Science Thesis

Advisor: Asst. Prof. Erkan USLU

Robotic technologies are getting popular in today's world. Robots are used in a wide range of areas like defense industry, military, agriculture, research and rescue, logistics, healthcare etc. Tasks that are monotone, hard or dangerous for humans can be done by robots more efficiently and robustly. Different types of robots can be used depend on complexity of a task and technical knowledge of involved engineers.

There are lots of unique problems in many of application areas, each of which requires a specific solution in terms of robot platforms and software. Today, mobile ground robots are widely used in both academic and commercial projects. However, some applications such as exploration can be implemented more efficiently due to the mobility of unmanned aerial vehicles (UAVs). Although there are different types of UAVs in the market, multi-propeller UAVs are preferred due to their low costs, better mobility and easier dynamics compared to other UAVs.

When multiple UAVs are used together, a task can be accomplished even much more quickly and effectively. However, especially when working with more than one multirotor UAV in indoor environments, it can cause problems even if the environment is static. The problem is that UAVs moving indoors may crash into each other. In the path planning phase of navigation, possible collisions can be eliminated between UAVs, which are dynamic obstacles in this case.

In this context, a system that creates a common 3D map using sensor data from multiple UAVs and navigates these UAVs using 3D costmaps has been developed. In order to prevent UAVs from hitting each other while navigating, a method that is based on costmaps has been implemented using Robot Operation System (ROS). For multi mapping, a customized version of OctoMap is used. Path planning is done by using A* and Lee Position Controller is used for path tracking. Experiments have been done in different simulation environments in Gazebo using RotorS simulator.

Keywords: Multi robot systems, unmanned aerial vehicles, multi navigation, multi mapping, path planning

---

ÖZET

Dinamik Ortamda Otonom Mobil Robot Navigasyonu

Mustafa Burak DİLAVER

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Dr.Öğr.Üyesi Erkan USLU

Günümüz dünyasında robotik teknolojileri giderek yaygınlaşmaktadır. Robotlar savunma sanayi, askeriye, tarım, arama ve kurtarma, ulaşım, sağlık gibi birçok alanda kullanılmaktadır. Tekdüze, insanlar için zor ve tehlikeli olan görevler robotlar tarafından daha randımanlı ve sağlam şekilde yerine getirilebilmektedir. Bir görevin karmaşıklığına ve ve ilgili mühendislerin teknik yetkinliğine bağlı olarak farklı tür robotlar kullanılabilir.

Robot platformu ve yazılımı açısından özgün çözümler gerektiren uygulama alanlarının çoğunun kendine has problemleri bulunmaktadır. Günümüzde mobil yer robotları hem akademik hem ticari projelerde yaygın bir şekilde kullanılmaktadır. Fakat keşif gibi bazı uygulamalar insansız hava araçlarının (İHA) hareket kabiliyetleri nedeniyle daha verimli bir şekilde yerine getirilebilir. Piyasada farklı tip İHA'lar bulunmakla beraber çok pervaneli İHA'lar düşük maliyetleri, diğer İHA'lara göre daha iyi hareket kabiliyetleri ve daha kolay dinamikleri sebebiyle tercih edilmektedir.

Birden fazla İHA birlikte kullanıldığında, bir görev daha da hızlı ve etkili bir şekilde başarılabilir. Ancak birden fazla çok pervaneli İHA ile çalışıldığında özellikle iç ortamlarda ortam statik olsa bile bu problemlere sebebiyet verebilmektedir. Buradaki problem iç ortamda hareket eden İHA'ların birbirine çarpabilmesidir. İHA navigasyonun yol planlama kısmında dinamik engeller olan İHA'ların arasındaki olası çarpışmalar önlenebilir.

Bu kapsamda çoklu İHA'lardan gelen sensör verilerini kullanıp ortak bir 3 boyutlu harita oluşturan ve bu İHA'ları 3 boyutlu bir maliyet haritası kullanarak otonom hareket ettiren bir sistem geliştirilmiştir. İHA'ların otonom hareket ederken birbirine çarpmaması için maliyet haritaları temelli bir yöntem Robot Operating System (ROS) kullanarak gerçeklenmiştir. Çoklu haritalama için OctoMap'in özelleştirilmiş bir hali kullanılmıştır. Yol planlama için A* ve yol takibi için Lee Pozisyon Kontrolcüsü kullanılmıştır. Deneyler Gazebo içerisinde farklı simülasyon ortamlarında RotorS simülatörü kullanılarak yapılmıştır.

Anahtar Kelimeler: Çoklu robot sistemleri, insansız hava araçları, çoklu navigasyon, çoklu haritalama, yol planlama

1
INTRODUCTION

1.1 Literature Review

An autonomous UAV needs to have many capabilities for doing tasks in an unknown/known environment. A UAV which is also a kind of mobile robot, may need to localize itself, map and navigate in its environment and explore its surroundings, allowing it to move autonomously.

The UAV platform should be selected according to the application environment. There are many different types of UAVs available for different types of applications. Fixed wings, helicopters, airships, tilt-rotors and multicopters, which are also called drones, are examples of UAV types [1] [2]. Two most commonly used type of UAVs are fixed wings and rotary wings, i.e. multicopters. Fixed wing UAVs have longer flight time and the ability of flying on higher altitudes due to their aerodynamic structure, while multirotors have the ability to hover vertically and a better ability to stay in position in the air, which brings complex control to the former and easier control to the latter [3]. Maneuvering ability and smaller size of multirotors make them a better alternative when it comes to indoor applications [4]. It is possible to use pre-built multirotor platforms for academic studies [5] [6].

Localization is a crucial step for autonomous mobile robots. While drones could localize themselves using GPS (Global Positioning System) in outdoor environments, they also need to exploit different sensor data, for example data obtained from Inertial Measurement Units (IMU) and stereo vision cameras, for indoor environments [7] [8] [9]. For outdoor applications, even the wide availability of Global Navigation Satellite Systems (GNSS) of today, they may provide noisy data and not be available for some reason [10]. Mapping is a concept of creating meaningful representations of unknown environments [11] [12]. Maps are used as an input for motion planning and exploration. Simultaneous localization and mapping (SLAM) algorithms gives robots the ability of localizing themselves and creating maps of their surrounding environments [13] [14] [15]. SLAM is a very important concept and tool for mobile ground robots therefore there's been many studies done on this subject [16] [17]. SLAM makes it possible creating advanced systems with mobile robots. It's possible for mobile ground robots to use range finders such as 2D light detection and ranging (LIDAR) and radio detection and ranging (RADAR) sensors data to perform SLAM because of their mostly planar movement space in applications. 2D LIDARs also can be used to navigate drones in indoor environments [18]. There are also studies which try to take advantage of deep learning methods for doing SLAM with 2D LIDAR data on UAVs [19].

Movement ability in 3D space of drones and huge amount of dense data capturing capability of cameras, make visual data a good option for doing SLAM on UAVs. Visual simultaneous localization and mapping (VSLAM) is a SLAM variant which takes the advantage of visual data obtained from today's highly improved and affordable cameras [20]. There are many VSLAM studies done on UAV platforms. Indoor UAVs can be smaller than outdoor ones because of the frequent obstacles and limited altitude in an indoor environment. Smaller UAVs have smaller motors which means they can carry less weight, thus having compact sensors like cameras gives a great advantage to indoor UAVs. There are different type of cameras used in different studies [21].

Monocular cameras are widely used in VSLAM studies on UAVs because they are common, inexpensive and available on many commercial drone platforms [22] [23] [24]. Simulation environments are also used in VSLAM studies that use monocular camera data [25] [26]. It is possible to do autonomous landing using monocular cameras [27]. Multiple UAVs with monocular cameras on them can accomplish tasks while doing VSLAM [28]. UAVs with monocular camera can also do obstacle avoidance with VSLAM [29].

Stereo cameras are another type of sensor used on UAVs. Stereo cameras offer data that can achieve better results in odometry calculation than monocular cameras. Autonomous take-off, flight and landing can be done with odometry information produced using stereo camera data [30] [31]. Odometry data can be combined with different information, e.g. beacon-based localization data [32]. Most of the stereo and RGBD cameras on the market provide depth images and such cameras, which provide data in RGBD format, are highly preferred in robotic studies. Since RGBD data, which is generally obtained at 30Hz from commercial stereo cameras, includes point cloud data, the size of the image is greater than plain RGB data. However, VSLAM can be performed in real time on UAVs using stereo cameras [33] [34] [35]. While doing VSLAM with RGBD data, UAVs are able to do tasks like assisting teleop driving [36], doing obstacle avoidance [37] and fully autonomous navigation [38]. Localization, mapping and navigation operations can be done on-board or off-board [39]. Two stereo cameras can be placed in different positions on a UAV, one facing towards the front and one facing towards the bottom of the UAV, and their pose estimation can be merged to create a more accurate odometry of the UAV [40]. Optical flow sensors are also a good odometry source and they can be use in a combination with stereo cameras [41].

After having a decent localization and mapping information, drones need solutions to move autonomously in unknown environments. One of the fundamental tasks of robotics is called navigation. Navigation can be separate into two levels, path planning and path tracking. Path planning involves calculating a collision free path from a destination position to a target position using sensor data and pre-created maps also optimality and calculation time of a path finding algorithm are important factors [42] [43]. There are different approaches in literature about path planning [44]. Finding obstacle free safe corridors using A* is one option to find paths in point cloud data [45]. It is possible to use pruning in this safe corridor finding process to decrease the time path planning takes for a UAV [46]. Rapidly-Exploring Random Tree (RRT) based algorithms are widely used in UAV path planning [47] [48]. Also there are hybrid path finding methods [49] [50].

Second step of navigation is path tracking. Producing accurate movement commands for a robot to make it follow a pre-defined path is what makes a robot move. There are algorithms which outputs required motor commands for a given set of a destination and a target point [51].

Collision avoidance is another important part of navigation topic. Sampling based methods try to simulate the drone movement and choose best local path to avoid dynamic obstacles [52]. Rapid response to obstacles is important due to the nature of UAVs. Increasing the processing speed by using the depth data with basic and minimal feature extraction can enable obstacle avoidance with low latency [53].

As robotics applications get more popular, studies on multi robot systems have been increasing. Multiple robots could done a task more effectively than a single robot in terms of speed, power usage and quality etc. [54]. Using multiple robots comes with a requirement for more efficient and complex control of all robots [55]. UAVs and mobile ground robots can be used in co-operation to do delivery related tasks [56]. Determination of which UAV should done which task is an important problem in multi UAV studies, e.g. exploration [57] [58] [59].

Having multiple UAVs in an environment increases possibility of UAVs getting crashed into each other. Increased movement ability over ground robots of UAVs makes them more vulnerable to crashes. When a ground robot crashed into a collision it’s likely that robot can continue to operate but when a UAV hit a collision it's very likely that it will make it incapable of working right thus collision avoidance is also important in multi navigation. Taking other UAVs into account while path planning is a way to avoid possible crashes in the literature [60].

1.2 Objective of the Thesis

Localization, mapping and navigation are essential topics for doing exploration and many other robotics tasks. Having multiple UAVs in an unknown indoor environment, coordinating them and making them accomplish meaningful tasks is a challenging problem.

In this study, a system has been developed in which multiple UAVs can navigate without interfering each other in a static indoor environment. In the second section, information is given about the ROS used as a framework, Gazebo used for simulations and the platform used for UAVs. In the third section, information about mapping and multi mapping output is given. The fourth section focuses on navigation. In the fifth section, a summary of the publication of the thesis is given. In the sixth section, the thesis is completed by making comments about the study.

1.3 Hypothesis

Within the scope of the thesis, a system has been created for multiple UAVs to navigate in indoor environments without hitting each other. While doing this, a multiple 3D mapping method was used. A costmap was created from this 3D map. A costmap was used during road planning. While the cells that should not be planned over have high cost, the cells that can be planned over are considered cost-free. As a result of the study, it was seen that multi mapping can be done in various simulation environments without problems.

The hypothesis can be expressed as follows: "Multi UAV navigation is possible in indoor environments using only costmaps in simulation."

2

INFRASTRUCTURE

Information about the ROS framework, Gazebo simulator and UAV platform used in the study is given in this section.

2.1 ROS

ROS is a framework for robotics projects [61]. ROS serves its features in four different categories.

The first is referred as plumbing, which means data communication between processes either on the same machine (computer) or on different machines. Because each robotics project involves data flowing from sensors to computers to actuators, there was a need to program each layer between these three main components. This creates a repetition for each new robot platform, sensor and actuator combination. Rather than programming new drivers between these layers, ROS lets developers use pre-written ones and if there is no already written driver for a given component, provides a set of communication protocols to ease the development process.

The second is the tools ROS gives to the use developers. Robotics studies involves the usage of many different sensors. There are many tools ROS provides; visualization tools for visualizing different types of data acquired from different sensors like camera, LIDAR, IMU, GPS etc., simulation tools, design tools and many. Using these tools creates a better environment for researchers and developers.

In third category ROS serves algorithm implementations which gives capabilities of localization, mapping, navigation, exploration etc. to robots. A robot system needs wide range of skills to overcome complex problems of the real world. Researchers might not have deep interest in all different topics of robotics. They can use the capabilities they need to build up a complete robot system using pre-built ROS packages. Then researchers could focus on their main work area and interest. For example a researcher who wants to work on navigation, can use SLAM packages ROS serves and focus on his/her main working field. Naturally ROS creates an ecosystem created by developers around the world and this is the fourth category of ROS's features.

2.1.1 Key Concepts

ROS has several key concepts that are used in the rest of this study so it's important to know about them.

Programs in ROS are called nodes. Each ROS node is nothing but a program that exploit ROS client libraries. Nodes are essential blocks of ROS. They receive and/or send data to other nodes in the same network. A ROS node may be responsible for reading sensor data and sending them to other ROS nodes, where as another ROS node may be responsible for processing data it receives from another node. Writing robotics programs in many ROS nodes gives a modular development experience to researchers/developers.

Nodes receive/send data through ROS topics. A ROS topic is an abstraction for establishing a modular communication between nodes. A ROS node needs to know nothing about another node it wants to communicate with but a common topic between them. There is a publish-subscribe concept in ROS topics. By publishing over a topic, a node can make data accessible to other nodes that subscribe to that topic. Topics can work in many to many way.

In every ROS network, there needs to be a ROS master node. All of the nodes needs to know where master is in the network. A node needs to get information on a topic before using it by asking master. Master node always keeps the information of which node publishes/subscribes to which topic and needs to run in a ROS system.

2.2 Gazebo

Gazebo is a simulation environment that targets robotics development. Doing tests in robotics studies is harder and more challenging than most of the other fields of computer engineering. Robots are built for doing tasks in real world, which requires them to be tested for the real world factors. Testing could be costly in terms of labour, time and price. Simulation environments try to make this testing phase shorter and more efficient. For this work, Gazebo was selected as a simulator for several reasons, mainly its ROS capabilities. There are other simulators that can work together with ROS but they are not mature and widely used as much as Gazebo [62].

2.2.1 History

Gazebo first appeared as project in 2002 at the University of Southern California. Several academicians continued its development but it's more commonly use became possible after it's been integrated with ROS by Willow Garage, same company who created ROS, in 2012. It's used as a simulation environment in a virtual robotics competition held by The Defense Advanced Research Projects Agency (DARPA), in 2013. Another competition called the Subterranean Challenge, which held by DARPA began in 2018 and is scheduled to finish in 2021, is also doing its virtual tasks on Gazebo. It's clear that government agencies contribute to the development of robotics by organising competitions where the famous DARPA Grand Challenge held in 2005 can be seen as another example [63] [64].

2.2.2 UAV Platform

Many different robot platforms such as ground robots, robotic arms, humanoid robots and air robots are simulated in Gazebo. These robots can be used in many different indoor / outdoor environments designed. There are several different UAV platforms that can be used in the Gazebo environment [65] [66]. The RotorS Simulator was selected to be used within the scope of the thesis study. There are several reasons behind this, such as the fact that drone models can move realistically on the Gazebo physics engine and a built-in local planner is provided to use. Simulation model used in the study is provided in Figure 2.1.

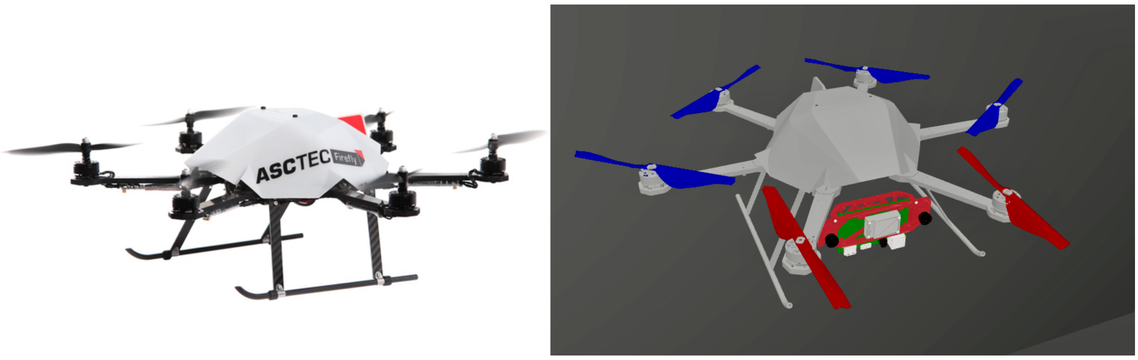

Figure 2.1 Real AscTec Firefly on the left and Gazebo model of it with a stereo camera attached to it on the right

3
MAPPING

Map representation of an environment is an important element in a robotics application. Using maps, operations such as path planning, localization, exploration and obstacle avoidance can be done. Different mapping methods examined are given in this chapter.

3.1 Depth Data

Depth data can be taken from the stereo cameras in real and simulation environments. The (x, y, z) coordinate of each pixel relative to the camera is stored in the depth data. This data is used as input for mapping. The visualization of the depth data taken in a Gazebo environment is given in the figure.

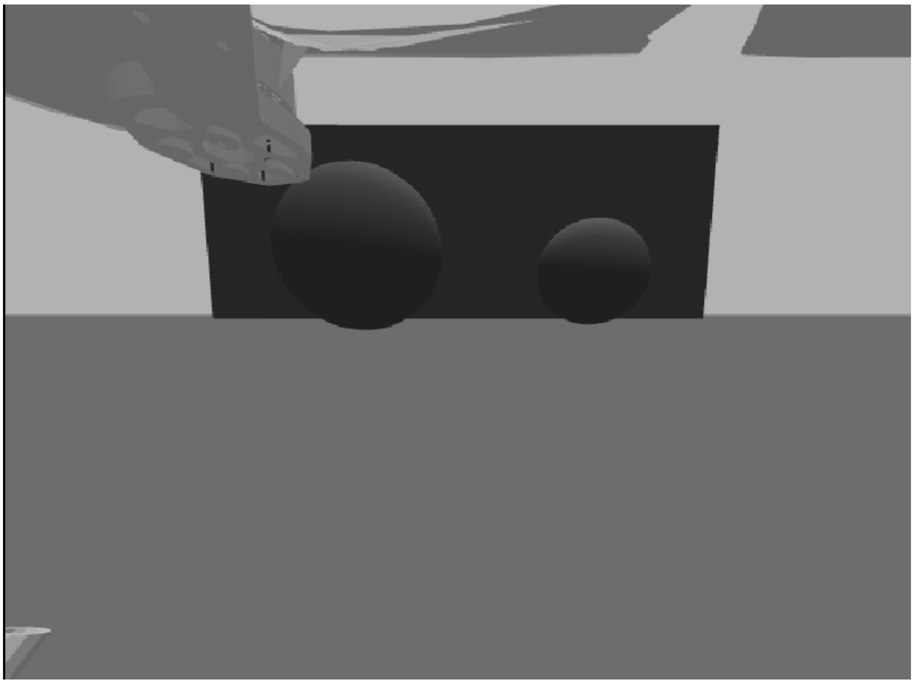

Figure 3.1 Image data taken from the stereo camera on the drone in Gazebo

RGB data of stereo camera is given in Figure 3.1. Depth data taken from stereo camera is given in Figure 3.3 for simulation environment given in Figure 3.2.

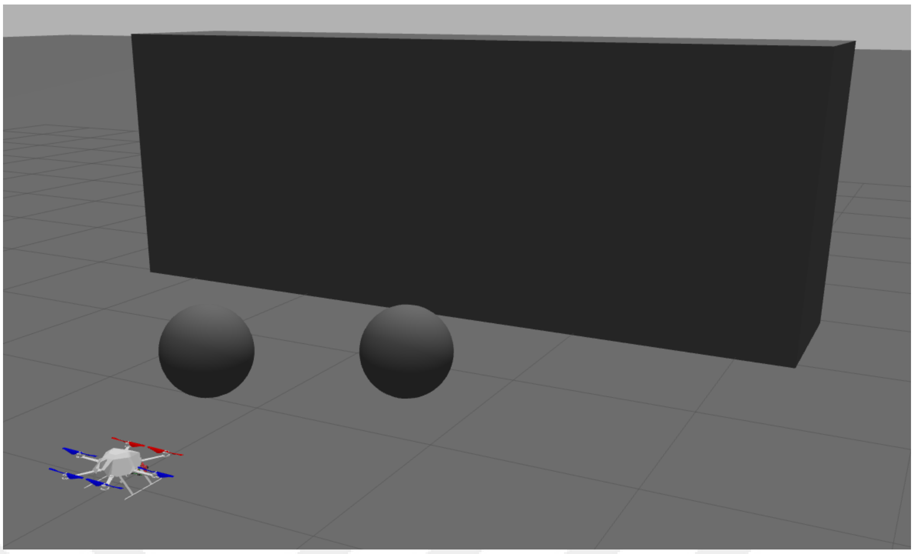

Figure 3.2 Firefly AscTec UAV in Gazebo and three obstacles in front of it.

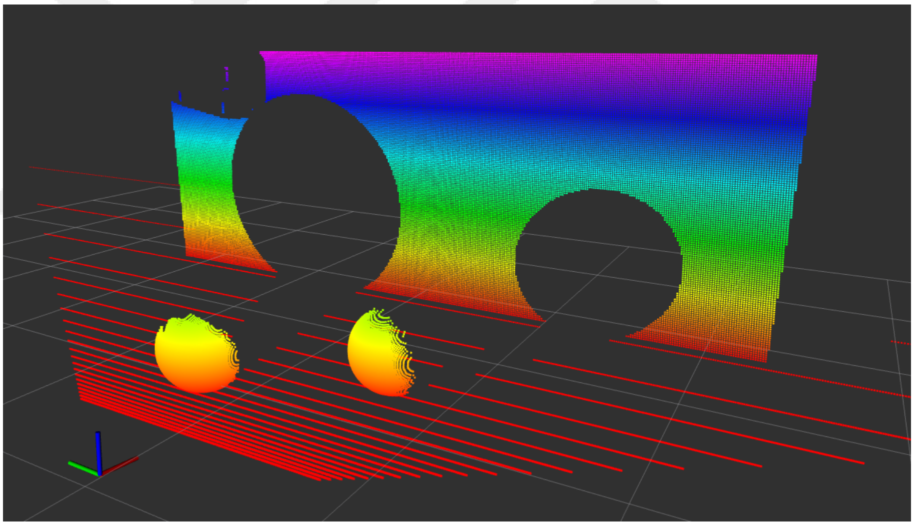

Figure 3.3 Depth data taken from stereo camera is projected in 3D space, camera frame is in the bottom left, different colors represent different vertical coordinate of points

3.2 OctoMap

3D maps can be created using 3D range measurements. The use of 3D maps means creating better representations of the environment. 3D maps consist of cells called grids. Each grid receives a value depending on whether it is empty or full. Cells that have not yet been discovered must be also represented on the map. Grid cells can be thought of as 3D matrices. As the grid size decreases, the resolution of the map increases. As the map resolution increases, the area that the map occupies in memory increases. A probabilistic method should be used to mark whether the grids are empty or full because the 3D measurement data from range sensors will never be ideal. Due to the error of the sensor, probabilistic methods are needed to eliminate the noise in the data [11].

The OctoMap method uses the octree data structure which is a tree representation where each node has eight child nodes to store map cells which is depicted in Figure 3.4. Octree structure decreases the amount of memory required for a map to be stored by pruning the tree when all child nodes of a parent node have the same value. In addition, optimizations were made for better memory usage. This makes it easier to transfer maps between programs in robotic applications. There is also an OctoMap package integrated with ROS [67].

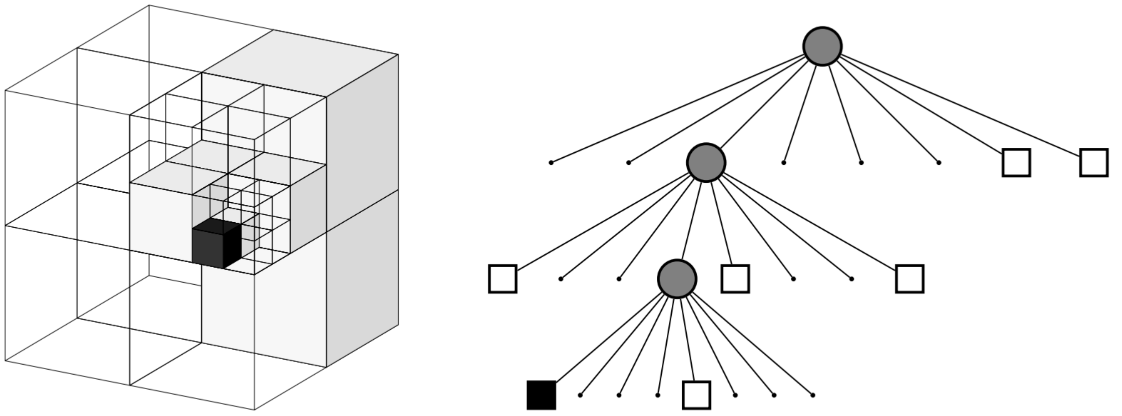

Figure 3.4 Octree representation given in the official paper [11]

In mapping methods, in the simplest sense, cells are marked full or empty by comparing some probabilities. The status of being full or empty of each map node needs to be updated using the new sensor data. In Occupancy grid mapping methods, this is done based on (3.1).

$$L(n|z_{1:t}) = L(n|z_{1:t-1}) + L(n|z_t) \tag{3.1}$$

In the OctoMap method, which is based on mentioned approach, the probability of being occupied of point cloud coordinates taken from the depth sensor is calculated by the probability formula (3.2).

$$L(n|z_{1:t}) = max(min(L(n|z_{1:t-1}) + L(n|z_t), l_{max}), l_{min}) \tag{3.2}$$

The probability of a leaf node n in octree to be filled according to $z_{1:t}$ sensor measurements taken at time t is expressed by $P(n|z_{1:t})$. If the most recent sensor measurement is z_t, the preliminary probability is $P(n)$, the previous being occupied probability of node n is expressed by $P(n|z_{1:t-1})$. According to the last sensor measurement, the probability of occupancy of n is expressed by $P(n|z_t)$. The logarithm of the odds ratio, which is the ratio of a probability to its complement, is called the log-odds ratio, or logit, and is given in (3.3).

$$L(n) = log\left[\frac{P(n)}{1-P(n)}\right] \tag{3.3}$$

Equation (3.1) is obtained when the log-odds ratio of (3.4) is calculated.

$$P(n|z_{1:t}) = \left[1 + \frac{1-P(n|z_t)}{P(n|z_t)}\frac{1-P(n|z_{1:t-1})}{P(n|z_{1:t-1})}\frac{P(n)}{1-P(n)}\right]^{-1} \tag{3.4}$$

The resulting (3.1) needs k measurements in order to change an occupied cell n to be empty. While this approach can be used in static environments, it is insufficient for dynamic applications. For this reason, by using (3.2) in the OctoMap method, changes in the environment are registered to the map more quickly.

3.2.1 Multi OctoMap

OctoMap is a method designed for a single robot by default. The OctoMap ROS package works for point cloud data coming from only a single robot. Point cloud is the name given to the data structure where each point taken from a depth sensor has its own 3D coordinate relative to the sensor. YTU CE Probabilistic Robotics Group is developing a version of OctoMap that can work with more than one robot simultaneously. Multi OctoMap is basically a method that combines Point Cloud data from multiple robots on a single map. The map can then be used by all the robots. With this structure, it enables applications such as navigation and exploration for multiple robots on a single map. Within the scope of the thesis, this multi octomap package was used.

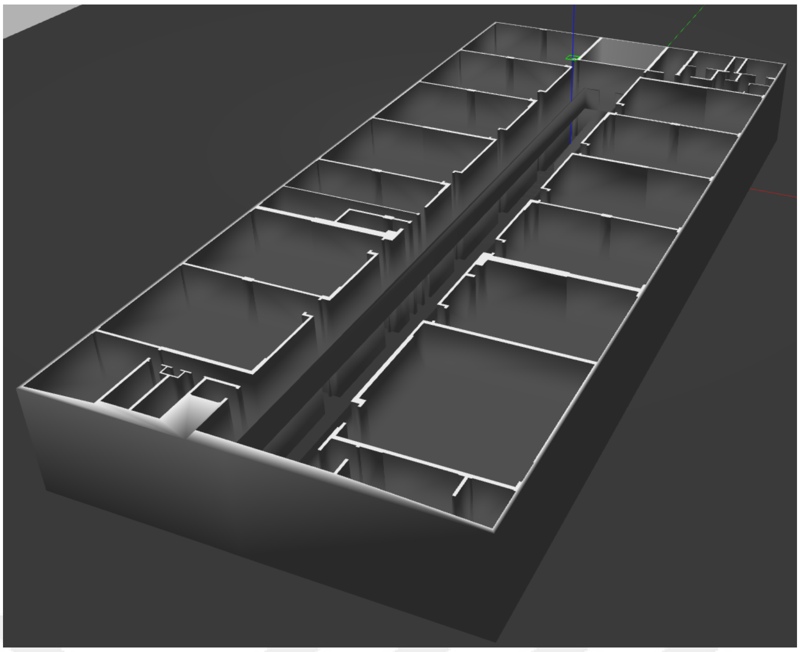

Figure 3.5 Gazebo environment of YTU Computer Engineering Department

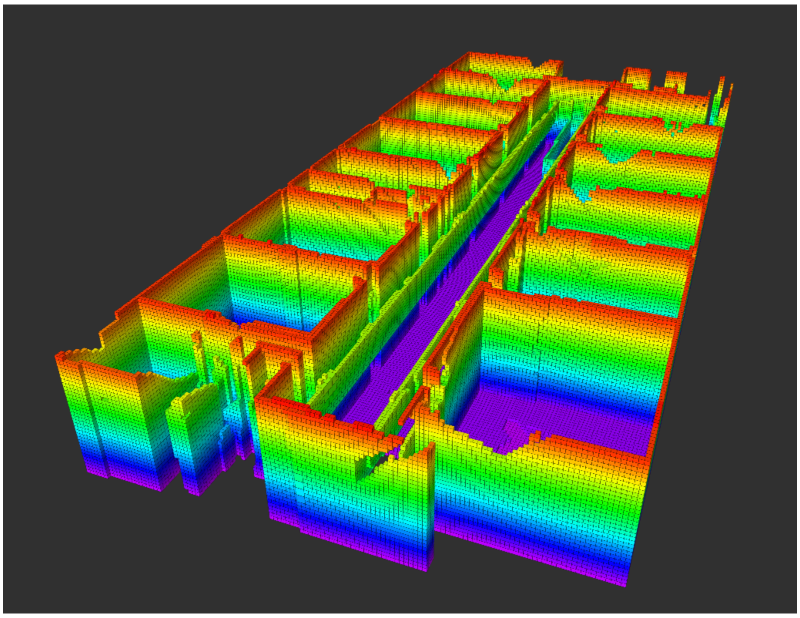

Figure 3.6 3D map of first floor of YTU CE Department model created using OctoMap

4
NAVIGATION

Autonomous robots need to navigate in unknown environments by their own. For a robot, navigation can be defined as the name given to the entire process of sensing its surroundings, determining where it is located in the environment, planning a path from a point to a target point, and tracking that path. In the ROS framework, there is a navigation stack that enables a representation of the environment to perform all operations such as path planning and path tracking. Rather than its broad meaning in the literature, when it comes to navigation in the ROS community, obstacle abstraction, path planning and path tracking processes come to mind. In this section, information about the navigation infrastructure and the different algorithms used are provided.

4.1 Path Planning

One of the most important tasks for a robot's autonomous movement in a real environment is path planning. For the autonomous movement of the robot, it is necessary to calculate an collision free path using pre-acquired data about the environment. Path planning algorithms are classified differently in the literature [44] [68]. Due to its stable, optimal and complete operation, node-based algorithms are widely used in road planning. A-star and its derivatives are called node-based algorithms.

4.1.1 A-star

A-star (A*) is an heuristic based search algorithm widely used in robotics due to its easy implementation steps and optimal operation. It tries to find an optimal path by minimizing $f(n)$ which is given in 4.1.

A* tries to navigate through the nodes adjacent to the target node from the start node. In robotic applications, cells on the map are used as nodes. $g(n)$ represents the cost from the start node to the current node where $h(n)$ is the heuristic distance to the target node. The A* algorithm optimizes the combination of these two, $f(n)$. The A* and derivative algorithms are complete which means if there is a path from the start node to the target node, it will always be found.

$$f(n) = g(n) + h(n) \tag{4.1}$$

4.2 Costmaps

The abstract maps used for robots to calculate collision free paths are called costmaps. Costmaps are used as an input to path planning. It also allows a robot, represented as a single dot, to move without hitting obstacles. Cost maps are used in ground robots for path planning and path tracking with obstacle avoidance. Cost maps are widely used on robots developed especially for environments where people move [69] [70]. By using costmap layers in socially aware robot studies and ROS navigation stack, optimization is taken into account in different ways for mobile and fixed obstacles [71].

Costmaps are used to plan the movement of ground robots. In ground robots, two-dimensional maps are generally created and navigation is done using two-dimensional cost maps created on these maps. Maps created in UAV applications are generally three-dimensional. Creating three-dimensional cost maps on these maps is not a method available in the literature. The map cells were inflated based on a specific multiplier of map resolution. The UAV's ability to move without hitting obstacles has been made possible by planning the path on the costmap.

A small experimental environment on how 3D costmap usage affects path planning is given in Figure 4.1. The path calculation using the costmaps on the 3D map created in this given environment is given in Figure 4.2. The inflation value of the cost map has been experimentally adjusted according to the size of the AscTec Firefly UAV. It is observed on the figure that the path is planned so that the UAV does not hit the obstacle columns.

In Figure 4.3, when the cost maps are not used, a bird's eye view of how a road calculation is made in the same scenario is seen. When cost maps are not used, the UAV will get close to the obstacles when it follows the calculated route, which leads to a crash. In Figure 4.4, the situation depicted in Figure 4.2 is given in bird's eye view.

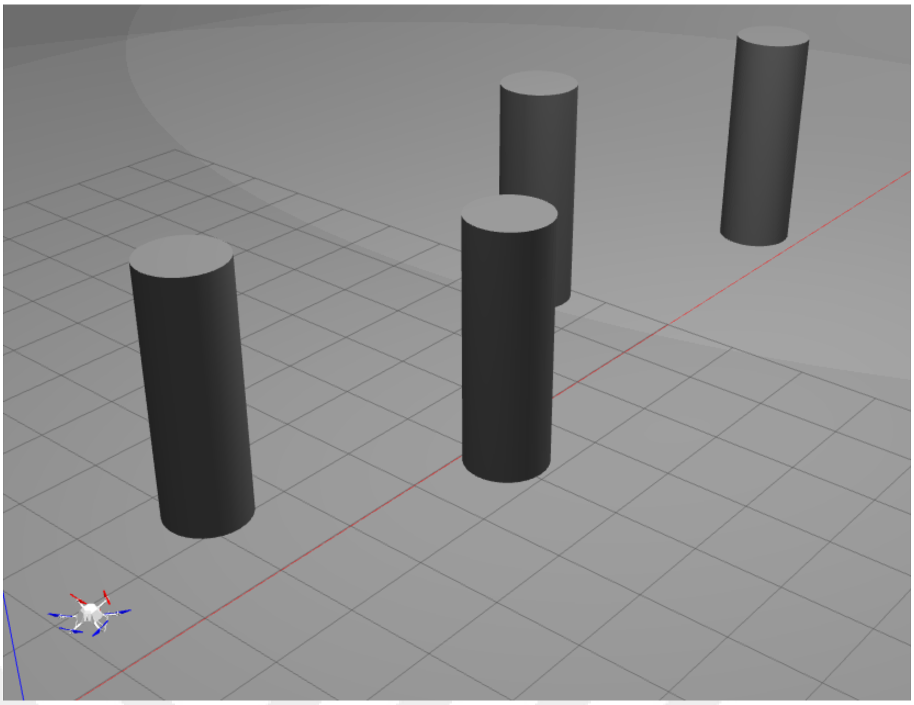

Figure 4.1 A simulation environment for testing path planning with costmaps

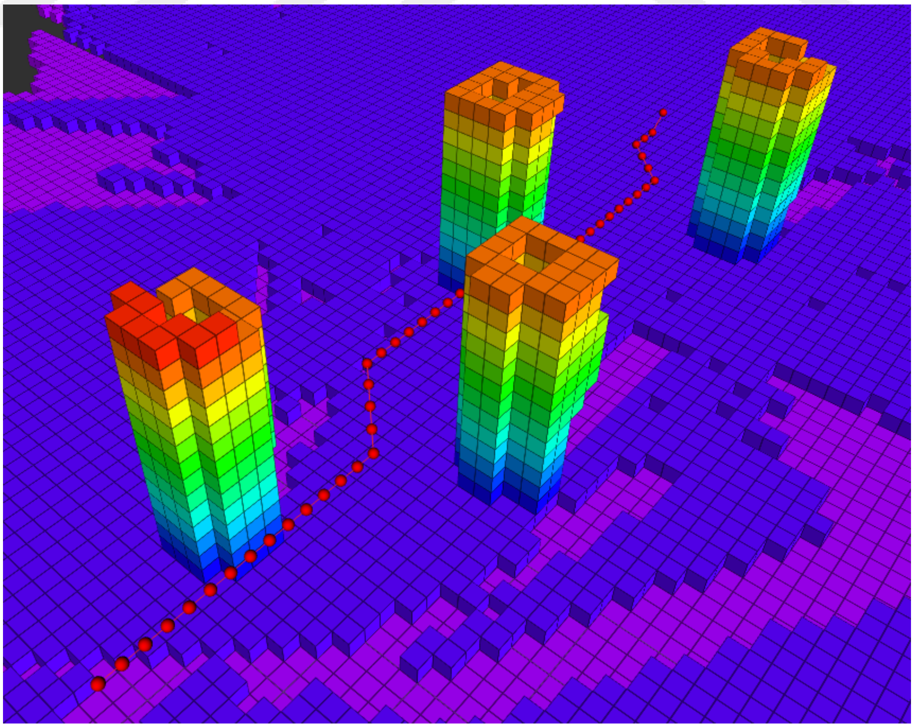

Figure 4.2 Planned path showed on 3D map of the given environment with using costmaps where red dots represents local goals

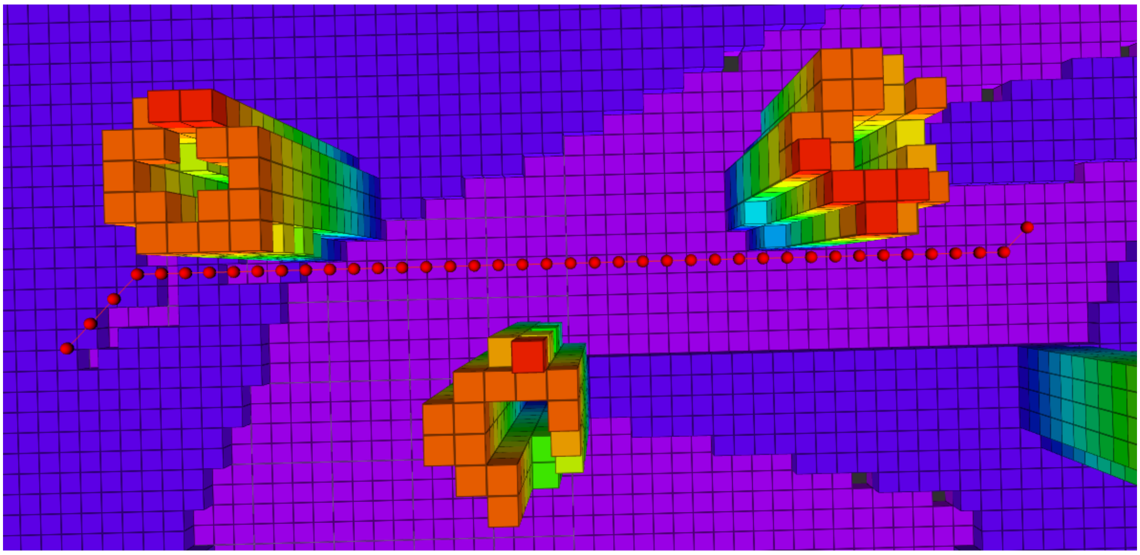

Figure 4.3 Planned path showed in 3D map of the given environment without using costmaps in bird view

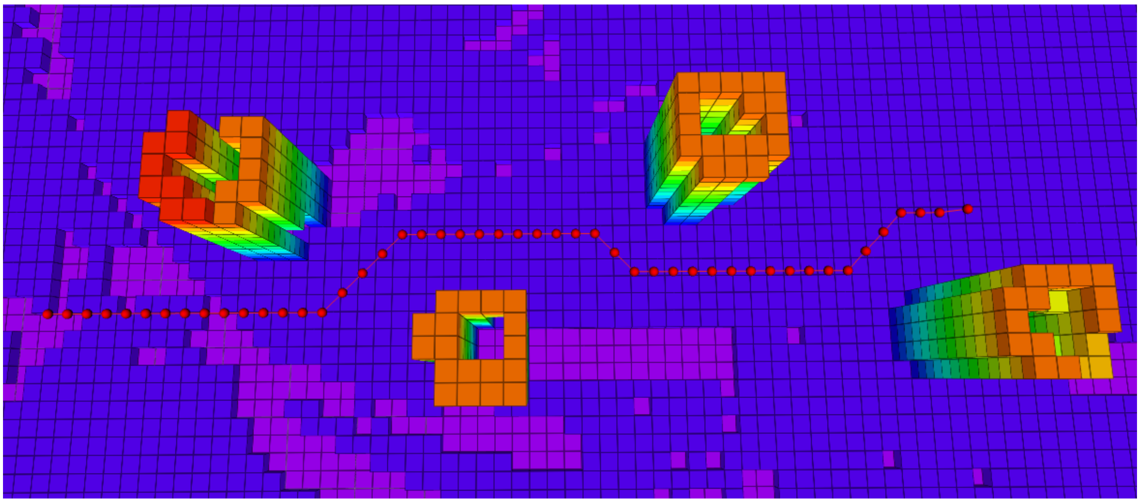

Figure 4.4 Planned path showed in 3D map of the given environment with using costmaps in bird view

4.3 Path Tracking

Path tracking, also called motion control, is a key issue for the autonomous movement of mobile robots. A robot that has calculated a path to a target point on the map of the environment must now generate motion commands that will follow this path in order to reach the target point.

Although UAVs have different approaches for path tracking, geometric approaches can be easier to implement; thus in this study, Lee Position Controller (LPC), a geometric approach, is used [51]. Path points were tracked one-by-one by the LPC to reach the target point via the calculated path.

4.4 Path Planning with Multiple UAVs

In environments where Multi UAVs operate, even if the environment is static, the movement of a UAV makes it a dynamic obstacle for other UAVs. If a UAV tries to plan roads without considering other UAVs moving in the environment, it may crash other UAVs. Path planning can be done by considering the planned paths of other UAVs.

One way to do this is to take advantage of costmaps. Costmaps refer to the costs on the map used in the path finding process. The cost of the regions where the previously planned and ongoing paths pass is increased and added to the cost map. This addition is done by inflating the map cells corresponding to the local points of the ongoing paths. As a result, a UAV that plans paths using the new costmap will calculate paths around the regions where the ongoing paths pass, possible collisions with other UAVs will be prevented. Data flow chart of this approcah for multiple UAVs is given in Figure 4.5. There are only two UAVs depicted in the chart for visualization but more than two UAVs can be used in the developed system. Thanks to modularity provided by ROS, modules with straight borders can be used in real world applications, with replacing some of the dashed modules, without any extra work which minimizes the integration from simulation to real application.

A representation of an inflated path being calculated by a UAV is given in Figure 4.6. In the upper part of Figure 4.6, the path calculated from cell A to cell B on map cells is given, where in the lower part, inflated version of that path is given. The amount of inflation can be changed to be the solid of the map cell. Figure 4.7 gives the path from point C to point D, which can be planned by another UAV considering previously planned path.

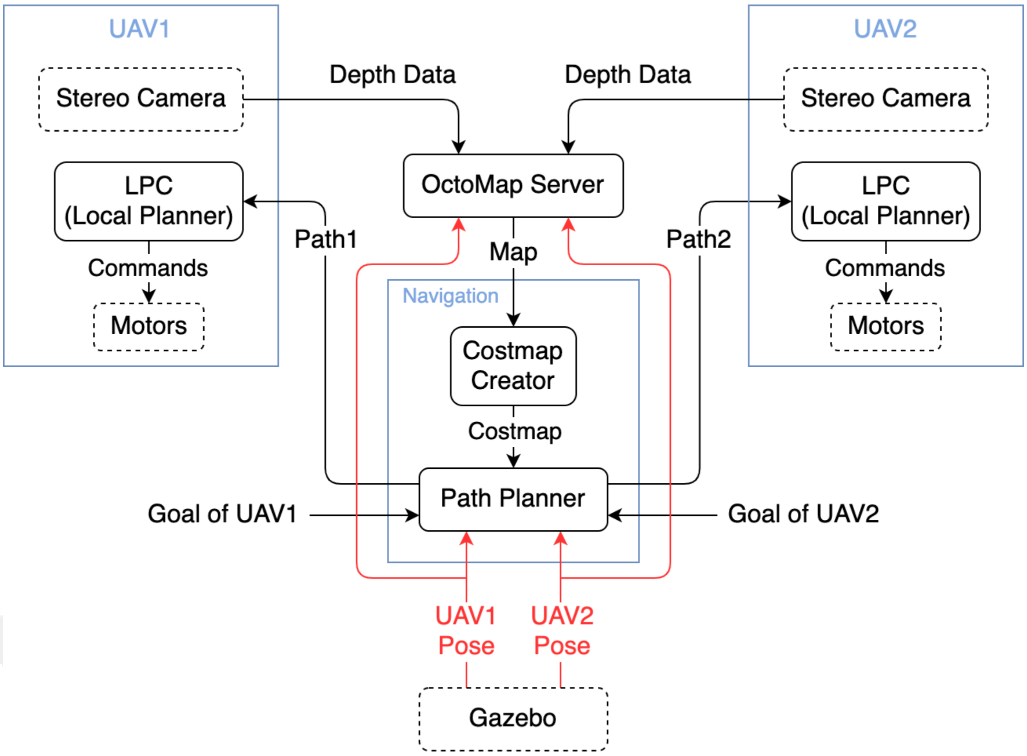

Figure 4.5 Data flow of costmap based multi path planning

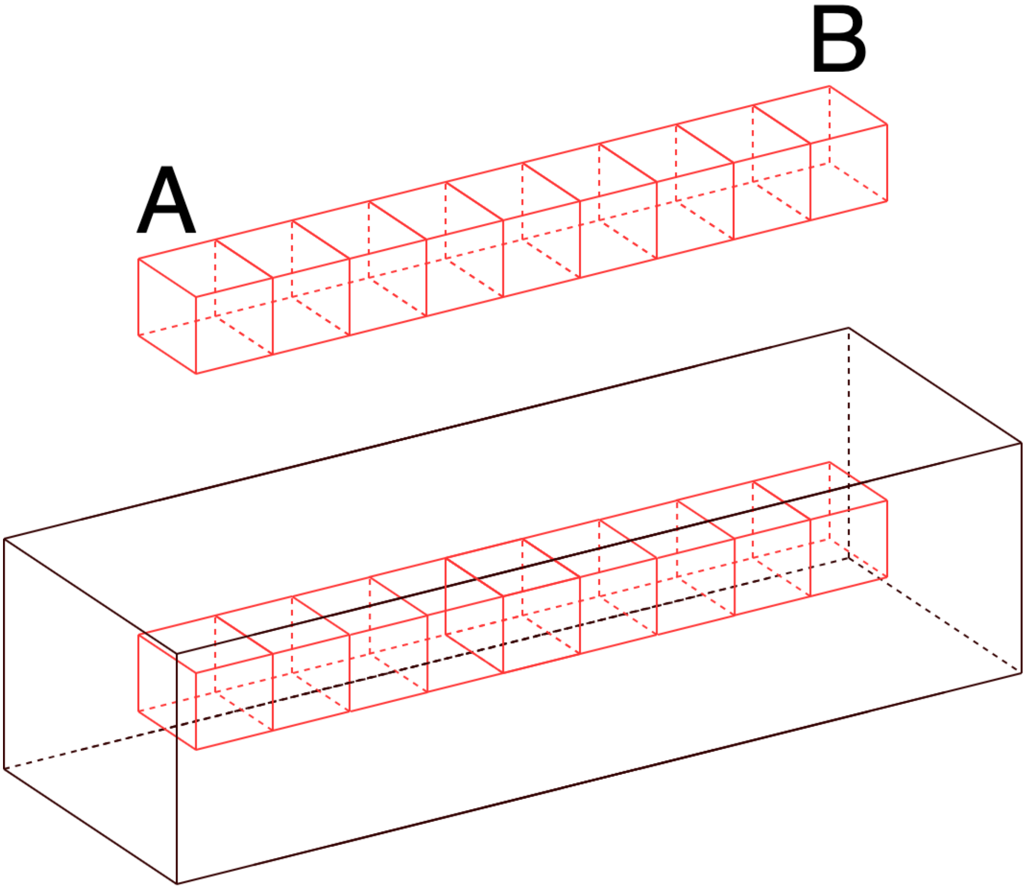

Figure 4.6 A path from cell *A* to cell *B* on the top, inflated version of that path on the bottom

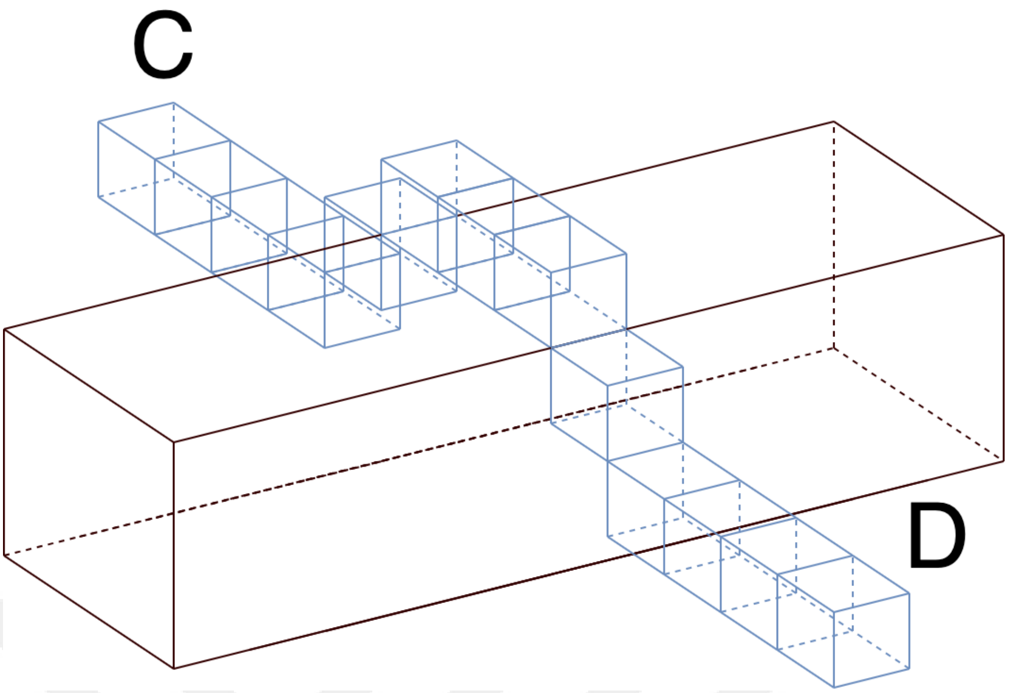

Figure 4.7 A path planned from *C* to *D* considering previously inflated active path

Examples of paths to be calculated when two UAVs in Gazebo need to calculate paths from the same map cells are given in Figure 4.8 and 4.9. In Figure 4.8, the path parallel to the floor calculated by UAV1 is given in blue. The path calculated by UAV2, which is in the same z coordinate as UAV1, is given in red. The ground truth odometry data of the UAVs when tracking these paths using LPC is visualized in green for UAV1 and purple for UAV2.

In Figure 4.9, UAV2 calculated a steep upward path at the point where it is located. It is seen that the road calculated by UAV1 is wrapped around this road. It can be seen that the two UAVs move without touching each other and approaching each other at a risky rate.

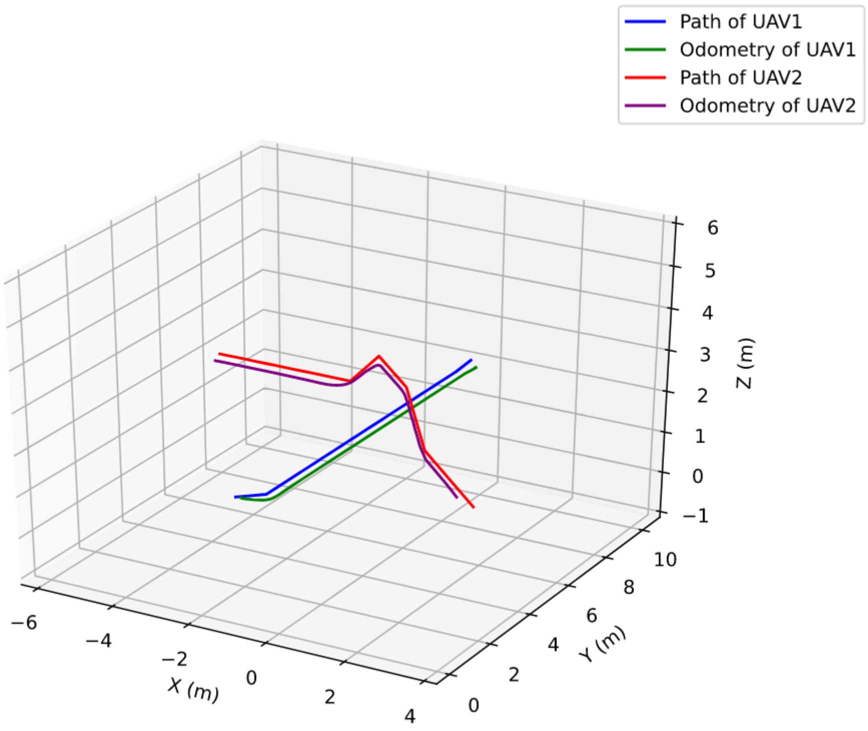

Figure 4.8 Paths for two UAVs and their ground truth odometry while UAV1 is moving in parallel to the floor and UAV2 is planning a path above UAV1's path

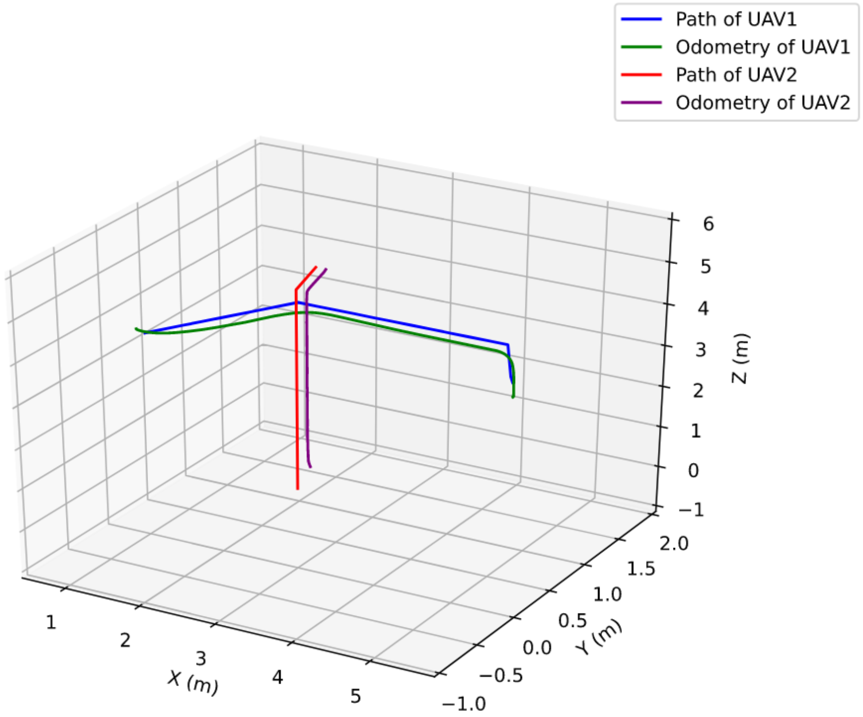

Figure 4.9 Paths for two UAVs and their ground truth odometry while UAV2 is ascending and UAV1 is planning a path around UAV2's path

5
3D MAPPING AND SIMULTANEOUS NAVIGATION FOR MULTI MICRO AERIAL VEHICLES IN INDOOR ENVIRONMENTS

One of the platforms where autonomous exploration applications can be used effectively is micro aerial vehicles (MAV), which are frequently used by robotic researchers due to their high mobility. Sending robots to environments that people cannot enter, observing the environment using data from robot sensors and trying to have an idea about the environment increase the need for exploration studies and accelerates the development of solutions produced in this area. Multiple robots can be used to make the exploration faster and more effectively. The use of multiple UAVs enables faster discovery of the environment. Working with more than one UAV in a closed indoor environment has many challenges.

UAVs need to plan paths to go from one point to another. However, if another UAV will pass over this road, it may be that the UAVs collide with each other. For this reason, each UAV must make its own route calculation, taking into account the routes planned by other UAVs. In this context, a system capable of multiple mapping and simultaneous navigation has been implemented in order to allow multiple UAVs to navigate in the same environment without hitting each other. A road planning technique using cost maps was used to make simultaneous navigation possible.

5.1 Simulation Environment

The trials of the system created in the study were made in an environment created in the Gazebo simulator. The simulation environment is one of the areas used in the Search and Rescue League of the RoboCup competition. This area was chosen because it has a narrow space for MAVs. The model used for MAVs is taken from a package called RotorS [65]. In the RotorS package, there are models of real models of AscTec MAVs that have been transferred to the simulation Gazebo environment. As the parametric adjustment of the MAVs for Gazebo physics engine is done well, their movements are quite realistic. Real and simulation photos of Firefly model can be seen in Figure 5.2.

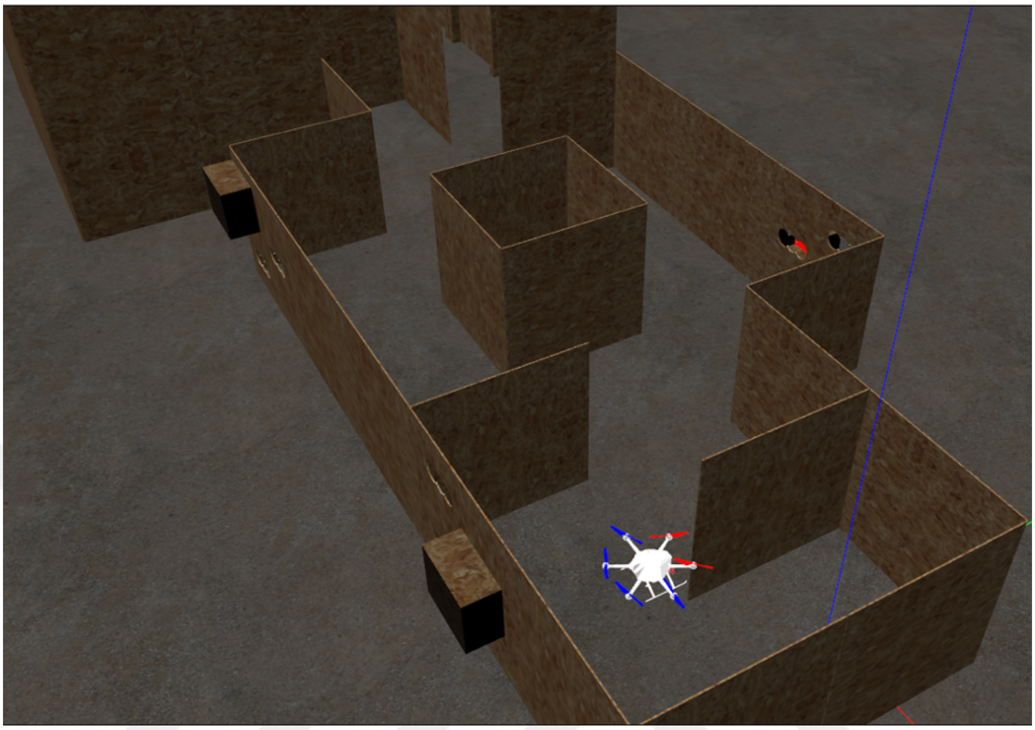

Figure 5.1 Part of the Gazebo environment used for this study

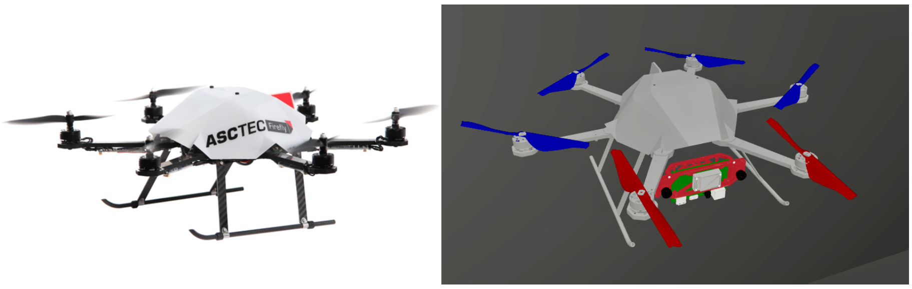

Figure 5.2 Real AscTec Firefly on the left, simulation model of it with a stereo camera attached in front of it on the right

5.2 Multi Mapping

In this study, the mapping was made using the operable version of the ROS octomap package for more than one robot developed by Yildiz Technical University Probabilistic Robotics Group. The improved mapping package basically takes range measurement data coming from all UAVs and merge them in a 3D map. Range data obtained from stereo cameras attached to each MAV. A map created using this package on environment given in Figure 5.1 is given in Figure 5.3.

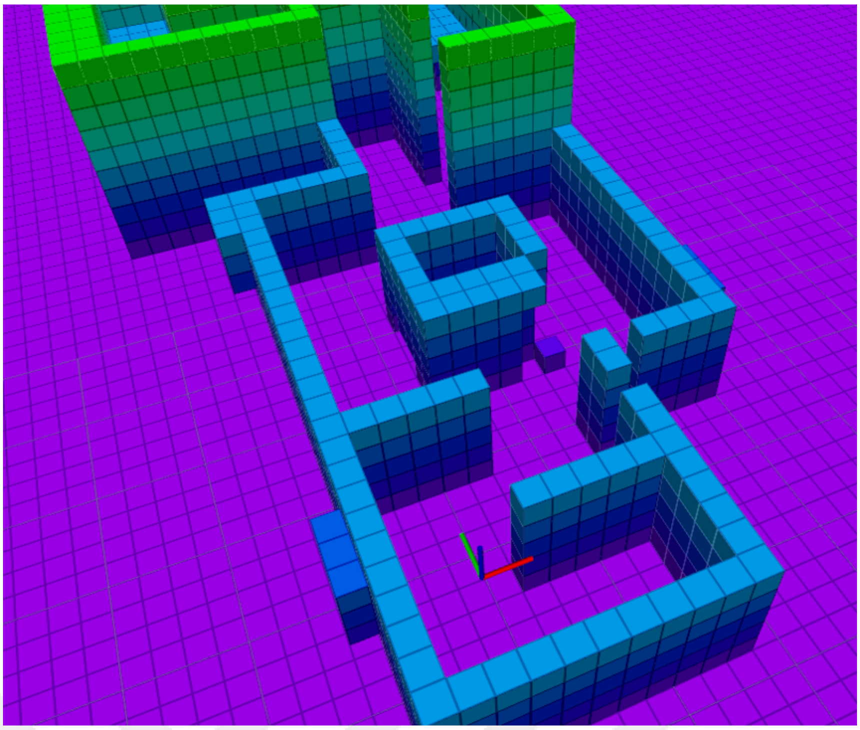

Figure 5.3 A 3D map created using OctoMap

5.3 Costmaps

How a robot is represented in space is an important issue. The robot position for a two-dimensional coordinate system is expressed in (x, y), while in the three-dimensional coordinate system it can be expressed as (x, y, z). This approach, which accepts the position of the robot as a single point, makes it easier to calculate paths and do obstacle avoidance in navigation. The robot's position is usually represented as the volumetric center point of the robot for making obstacle avoidance easier. When planning the road, the road is drawn around the obstacles. In the path tracking step, the robot position is tried to be kept on the calculated road. When the path drawn around the obstacles is calculated considering that a point will follow the path, the robot that occupies a volume in space will hit the obstacles when it gets close to the obstacles.

As a solution to this problem, costmaps can be used. Costmap is a technique that used to inflate obstacles on the map. Inflated obstacles get larger before the path planning step. Costmaps help to prevent the robot from hitting obstacles in the real world when it follows the found path.

In order to create a 3D cost map for UAVs, obstacle cells on the 3D map are inflated. In Figure 5.4, inflated cells by the amount of 1 cell are given. The cost map obtained for the map given in Figure 5.3 is given in Figure 5.5.

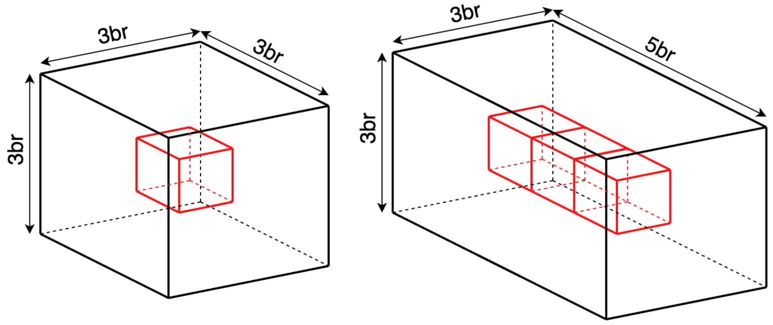

Figure 5.4 Red cells are the original map cells, black volumes represent inflated results

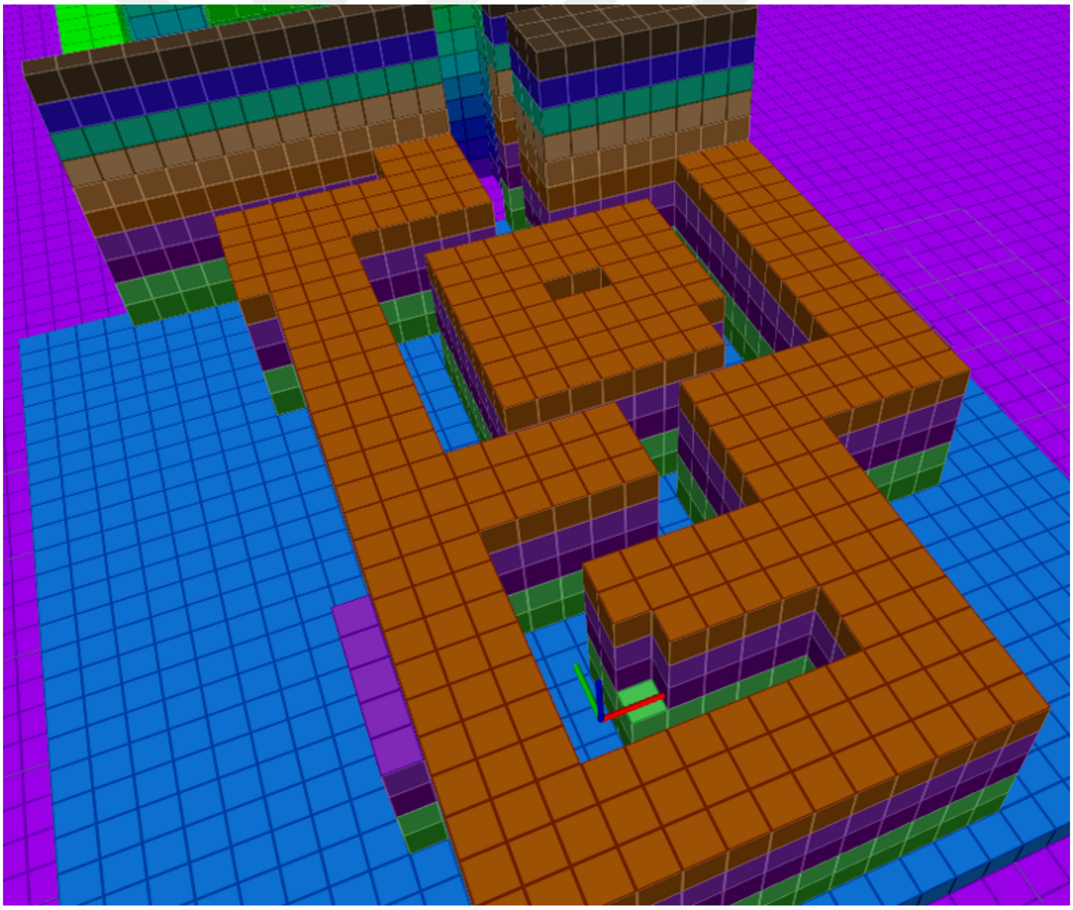

Figure 5.5 Inflated costmap of a given 3D map

5.4 Path Planning

A three-dimensional version of the A* algorithm was used to plan the route from the UAVs to the target points on the obtained map. The cost from the starting point to the n node is expressed as $g(n)$, the estimated cost to the destination point is $h(n)$, and the total cost is $f(n)$. Situations that require changing direction between two adjacent cells are also added as extra costs to the $g(n)$. In this way, it is aimed to calculate routes with minimum amount of direction change requirement. $h(n)$ is calculated based on the Euclidean distance to the target. The basic A* equation expressed in the Figure 5.1.

$$f(n) = g(n) + h(n) \tag{5.1}$$

5.5 Path Tracking

Path tracking is done using the Lee Position Controller. Given a target space coordinate and heading angle parallel to the ground, Lee Position Controller produces the required propeller thrusts. RotorS has a path tracking package that works for quad and hexa rotors and is used in this project. Successive coordinates on the found path is given to local planner and whole path is tracked by this way.

5.6 Result

In the simulation environment, it was observed that multiple UAVs can move and map the environment without hitting each other. A demonstration video is given in [72].

6
RESULTS AND DISCUSSION

In this study, a system where multiple multirotor UAVs can navigate in static indoor environments without crashing each other has been created. To achieve this, a 3D map was created using an OctoMap version that works with depth data of stereo cameras from multiple robots. A single costmap can be obtained from the extracted map and road planning can be made for each UAV on this costmap by using A* algorithm. The local targets of the calculated path were given to the LPC method one by one and the path was completed this way. By increasing the cost of the map cells around the planned and active path of a UAV, it was ensured that other UAVs did not pass through this path when they planned a path. The goal of this approach has been to try to prevent collisions between UAVs by ensuring that their global paths do not intersect. ROS was used as a framework and the RotorS drone simulator was used on Gazebo as a simulation medium. The created system has been tested in Gazebo environments. With this method, it has been observed that two or more UAVs can operate in the same indoor environment without hitting each other. UAV and related robotics studies in the literature have been examined and information about current studies has been obtained.

The study does have many improvements to be made in the future. First things first, the study could have been tried with real multirotor UAVs in the real world, although sufficient results were obtained in the simulation. Although simulation environments are tried to be designed realistically, it is obvious that Gazebo is not the best simulator on the market, the real world has its own problems.

There are many path planning and path tracking methods in the literature for UAV navigation. Faster road planners can be used to find paths faster for multiple robots. Algorithms that plan paths can be tried by taking into account the dynamic constraints of robots instead of costmaps. The calculated paths can be smoothed. Instead of giving all the local targets of the calculated path to the path tracker, paths can be represented with fewer local points and can be followed faster by doing some optimizations.

The whole system can be built as a more effective and modular program and it can be easier to try different methods. Contribution to the open source world with a system whose modularity is trusted can be made. As the studies on UAVs are increasing day by day, quality tools offered in this field for researchers will attract great attention.

REFERENCES

[1] K. Dalamagkidis, “Aviation history and unmanned flight,” in *Handbook of Unmanned Aerial Vehicles*, Springer Netherlands, Jun. 2015, pp. 57–81, ISBN: 9789048197071. DOI: `10.1007/978-90-481-9707-1_93`.

[2] S. Gupte, P. I. T. Mohandas, J. M. Conrad, “A survey of quadrotor unmanned aerial vehicles,” in *Conference Proceedings - IEEE SOUTHEASTCON*, 2012, ISBN: 9781467313742. DOI: `10.1109/SECon.2012.6196930`.

[3] M. A. Boon, A. P. Drijfhout, S. Tesfamichael, “Comparison of a fixed-wing and multi-rotor UAV for environmental mapping applications: A case study,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 42, International Society for Photogrammetry and Remote Sensing, Aug. 2017, pp. 47–54. DOI: `10.5194/isprs-archives-XLII-2-W6-47-2017`.

[4] N. Dijkshoorn A. Visser, “Integrating sensor and motion models to localize an autonomous AR. Drone,” *International Journal of Micro Air Vehicles*, 2011, ISSN: 17568293. DOI: `10.1260/1756-8293.3.4.183`.

[5] T. Krajník, V. Vonásek, D. Fišer, J. Faigl, “AR-drone as a platform for robotic research and education,” in *Communications in Computer and Information Science*, 2011, ISBN: 9783642219740. DOI: `10.1007/978-3-642-21975-7_16`.

[6] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, J. Buchli, “Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2016, ISBN: 9781467380263. DOI: `10.1109/ICRA.2016.7487274`.

[7] B. Remes, D. Hensen, F. van Tienen, C. De Wagter, E. van der Horst, G. de Croon, “Paparazzi: how to make a swarm of Parrot AR Drones fly autonomously based on GPS,” in *IMAV 2013*, 2013.

[8] A. Hussein, A. Al-Kaff, A. De La Escalera, J. M. Armingol, “Autonomous indoor navigation of low-cost quadcopters,” in *10th IEEE Int. Conf. on Service Operations and Logistics, and Informatics, SOLI 2015 - In conjunction with ICT4ALL 2015*, 2015, ISBN: 9781467384803. DOI: `10.1109/SOLI.2015.7367607`.

[9] Y. M. Mustafah, A. W. Azman, F. Akbar, “Indoor UAV positioning using stereo vision sensor,” in *Procedia Engineering*, vol. 41, Elsevier Ltd, Jan. 2012, pp. 575–579. DOI: `10.1016/j.proeng.2012.07.214`.

[10] F. Nex F. Remondino, *UAV for 3D mapping applications: A review*, Nov. 2014. DOI: `10.1007/s12518-013-0120-x`.

[11] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, “Octomap: An efficient probabilistic 3d mapping framework based on octrees,” *Autonomous Robots*, vol. 34, Apr. 2013. DOI: `10.1007/s10514-012-9321-0`.

[12] S. Thrun, “Robotic Mapping: A Survey,” Tech. Rep., 2002.

[13] G. Grisetti, C. Stachniss, W. Burgard, “Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, 2005, pp. 2432–2437, ISBN: 078038914X. DOI: `10.1109/ROBOT.2005.1570477`.

[14] C. Stachniss, J. J. Leonard, S. Thrun, “Simultaneous localization and mapping,” in *Springer Handbook of Robotics*, 2016, ISBN: 9783319325521. DOI: `10.1007/978-3-319-32552-1_46`.

[15] G. Grisetti, C. Stachniss, W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007, ISSN: 15523098. DOI: `10.1109/TRO.2006.889486`.

[16] W. Hess, D. Kohler, H. Rapp, D. Andor, “Real-time loop closure in 2D LIDAR SLAM,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, Institute of Electrical and Electronics Engineers Inc., Jun. 2016, pp. 1271–1278, ISBN: 9781467380263. DOI: `10.1109/ICRA.2016.7487258`.

[17] J. M. Santos, D. Portugal, R. P. Rocha, “An evaluation of 2D SLAM techniques available in Robot Operating System,” in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2013*, 2013, ISBN: 9781479908806. DOI: `10.1109/SSRR.2013.6719348`.

[18] S. Grzonka, G. Grisetti, W. Burgard, “A fully autonomous indoor quadrotor,” *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 90–100, Feb. 2012, ISSN: 1941-0468. DOI: `10.1109/TRO.2011.2162999`.

[19] J. Li, H. Zhan, B. M. Chen, I. Reid, G. H. Lee, “Deep learning for 2D scan matching and loop closure,” in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-September, Institute of Electrical and Electronics Engineers Inc., Dec. 2017, pp. 763–768, ISBN: 9781538626825. DOI: `10.1109/IROS.2017.8202236`.

[20] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, M. E. Munich, “The vSLAM algorithm for robust localization and mapping,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, 2005, pp. 24–29, ISBN: 078038914X. DOI: `10.1109/ROBOT.2005.1570091`.

[21] Y. Lu, Z. Xue, G.-S. Xia, L. Zhang, “A survey on vision-based UAV navigation,” *Geo-spatial Information Science*, vol. 21, no. 1, pp. 21–32, Jan. 2018, ISSN: 1009-5020. DOI: `10.1080/10095020.2017.1420509`.

[22] L. Von Stumberg, V. Usenko, J. Engel, J. Stuckler, D. Cremers, “From monocular SLAM to autonomous drone exploration,” in *2017 European Conference on Mobile Robots, ECMR 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, ISBN: 9781538610961. DOI: `10.1109/ECMR.2017.8098709`. arXiv: `1609.07835`.

[23] S. García, M. E. López, R. Barea, L. M. Bergasa, A. Gómez, E. J. Molinos, “Indoor SLAM for Micro Aerial Vehicles Control Using Monocular Camera and Sensor Fusion,” in *Proceedings - 2016 International Conference on Autonomous Robot Systems and Competitions, ICARSC 2016*, Institute of Electrical and Electronics Engineers Inc., Dec. 2016, pp. 205–210, ISBN: 9781509022557. DOI: 10.1109/ICARSC.2016.46.

[24] J. Park, S. Im, K. H. Lee, J. O. Lee, “Vision-based SLAM system for small UAVs in GPS-denied environments,” *Journal of Aerospace Engineering*, vol. 25, no. 4, pp. 519–529, Oct. 2012, ISSN: 08931321. DOI: 10.1061/(ASCE)AS.1943-5525.0000160.

[25] J. Ha R. Sattigeri, “Vision-based obstacle avoidance based on monocular SLAM and image segmentation for UAVs,” in *AIAA Infotech at Aerospace Conference and Exhibit 2012*, 2012, ISBN: 9781600869396. DOI: 10.2514/6.2012-2464.

[26] C. Wang, T. Wang, J. Liang, Y. Chen, Y. Zhang, C. Wang, “Monocular visual SLAM for small UAVs in GPS-denied environments,” in *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 - Conference Digest*, 2012, pp. 896–901, ISBN: 9781467321273. DOI: 10.1109/ROBIO.2012.6491082.

[27] T. Yang, P. Li, H. Zhang, J. Li, Z. Li, “Monocular Vision SLAM-Based UAV Autonomous Landing in Emergencies and Unknown Environments,” *Electronics*, vol. 7, no. 5, p. 73, May 2018, ISSN: 2079-9292. DOI: 10.3390/electronics7050073.

[28] J.-C. Trujillo, R. Munguia, E. Guerra, A. Grau, “Cooperative Monocular-Based SLAM for Multi-UAV Systems in GPS-Denied Environments,” *Sensors*, vol. 18, no. 5, p. 1351, Apr. 2018, ISSN: 1424-8220. DOI: 10.3390/s18051351.

[29] C. Fu, M. A. Olivares-Mendez, R. Suarez-Fernandez, P. Campoy, “Monocular Visual-Inertial SLAM-based collision avoidance strategy for Fail-Safe UAV using Fuzzy Logic Controllers: Comparison of two Cross-Entropy Optimization approaches,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 73, no. 1-4, pp. 513–533, Oct. 2014, ISSN: 15730409. DOI: 10.1007/s10846-013-9918-3.

[30] L. R. García Carrillo, A. E. Dzul López, R. Lozano, C. Pégard, “Combining stereo vision and inertial navigation system for a quad-rotor UAV,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 65, no. 1-4, pp. 373–387, Jan. 2012, ISSN: 09210296. DOI: 10.1007/s10846-011-9571-7.

[31] K. Schauwecker, N. R. Ke, S. A. Scherer, A. Zell, “Markerless Visual Control of a Quad-Rotor Micro Aerial Vehicle by Means of On-Board Stereo Processing,” in, Springer, Berlin, Heidelberg, 2012, pp. 11–20. DOI: 10.1007/978-3-642-32217-4_2.

[32] F. J. Perez-Grau, F. R. Fabresse, F. Caballero, A. Viguria, A. Ollero, “Long-term aerial robot localization based on visual odometry and radio-based ranging,” in *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, Institute of Electrical and Electronics Engineers Inc., Jun. 2016, pp. 608–614, ISBN: 9781467393331. DOI: 10.1109/ICUAS.2016.7502653.

[33] W. G. Aguilar, G. A. Rodríguez, L. Álvarez, S. Sandoval, F. Quisaguano, A. Limaico, “Visual slam with a rgb-d camera on a quadrotor uav using on-board processing,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10306 LNCS, Springer Verlag, 2017, pp. 596–606, ISBN: 9783319591469. DOI: 10.1007/978-3-319-59147-6_51.

[34] S. A. Scherer A. Zell, “Efficient onbard RGBD-SLAM for autonomous MAVs,” in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1062–1068, ISBN: 9781467363587. DOI: 10.1109/IROS.2013.6696482.

[35] R. G. Valenti, I. Dryanovski, C. Jaramillo, D. P. Strom, J. Xiao, “Autonomous quadrotor flight using onboard RGB-D visual odometry,” in *Proceedings - IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers Inc., Sep. 2014, pp. 5233–5238. DOI: 10.1109/ICRA.2014.6907628.

[36] F. J. Perez-Grau, R. Ragel, F. Caballero, A. Viguria, A. Ollero, “Semi-autonomous teleoperation of UAVs in search and rescue scenarios,” in *2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 1066–1074, ISBN: 9781509044948. DOI: 10.1109/ICUAS.2017.7991349.

[37] R. Ait-Jellal A. Zell, “Outdoor obstacle avoidance based on hybrid visual stereo SLAM for an autonomous quadrotor MAV,” in *2017 European Conference on Mobile Robots, ECMR 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, ISBN: 9781538610961. DOI: 10.1109/ECMR.2017.8098686.

[38] K. Schmid, T. Tomic, F. Ruess, H. Hirschmuller, M. Suppa, “Stereo vision based indoor/outdoor navigation for flying robots,” in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 3955–3962, ISBN: 9781467363587. DOI: 10.1109/IROS.2013.6696922.

[39] R. DePaola, C. W. Chimento, K. M. Brink, A. R. Willis, M. L. Anderson, “UAV navigation with computer vision - Flight testing a novel visual odometry technique,” in *AIAA Guidance, Navigation, and Control Conference, 2018*, 2018, ISBN: 9781624105265. DOI: 10.2514/6.2018-2102.

[40] K. Schauwecker A. Zell, “On-board dual-stereo-vision for autonomous quadrotor navigation,” in *2013 International Conference on Unmanned Aircraft Systems, ICUAS 2013 - Conference Proceedings*, 2013, pp. 333–342, ISBN: 9781479908172. DOI: 10.1109/ICUAS.2013.6564706.

[41] F. Fraundorfer, L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, M. Pollefeys, “Vision-based autonomous mapping and exploration using a quadrotor MAV,” in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 4557–4564, ISBN: 9781467317375. DOI: 10.1109/IROS.2012.6385934.

[42] L. E. Kavraki S. M. LaValle, “Motion Planning,” in *Springer Handbook of Robotics*, Springer Berlin Heidelberg, 2008, pp. 109–131. DOI: 10.1007/978-3-540-30301-5_6.

[43] G. Balamurugan, J. Valarmathi, V. P. S. Naidu, "Survey on uav navigation in gps denied environments," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, Oct. 2016, pp. 198–204. DOI: `10.1109/SCOPES.2016.7955787`.

[44] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger, P. Feyzeau, "Path planning: A 2013 survey," in *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, Oct. 2013, pp. 1–8.

[45] F. Gao S. Shen, "Online quadrotor trajectory generation and autonomous navigation on point clouds," in *SSRR 2016 - International Symposium on Safety, Security and Rescue Robotics*, Institute of Electrical and Electronics Engineers Inc., Dec. 2016, pp. 139–146, ISBN: 9781509043491. DOI: `10.1109/SSRR.2016.7784290`.

[46] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, Jun. 2017, ISSN: 2377-3774. DOI: `10.1109/LRA.2017.2663526`.

[47] X. Z. Peng, H. Y. Lin, J. M. Dai, "Path planning and obstacle avoidance for vision guided quadrotor UAV navigation," in *IEEE International Conference on Control and Automation, ICCA*, vol. 2016-July, IEEE Computer Society, Jul. 2016, pp. 984–989, ISBN: 9781509017386. DOI: `10.1109/ICCA.2016.7505408`.

[48] C. L. Lin, C. S. Lee, Y. J. Tsai, C. H. Huang, "Flight path planning for mini rotor UAVs," in *IEEE International Conference on Control and Automation, ICCA*, IEEE Computer Society, 2014, pp. 1339–1344, ISBN: 9781479928378. DOI: `10.1109/ICCA.2014.6871118`.

[49] F. Yan, Y. S. Liu, J. Z. Xiao, "Path planning in complex 3D environments using a probabilistic roadmap method," *International Journal of Automation and Computing*, 2013, ISSN: 14768186. DOI: `10.1007/s11633-013-0750-9`.

[50] P. Yao, H. Wang, Z. Su, "Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment," *Aerospace Science and Technology*, vol. 47, pp. 269–279, Dec. 2015, ISSN: 12709638. DOI: `10.1016/j.ast.2015.09.037`.

[51] T. Lee, M. Leok, N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se(3)," in *49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 5420–5425. DOI: `10.1109/CDC.2010.5717652`.

[52] Y. Lin S. Saripalli, "Sampling-Based Path Planning for UAV Collision Avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3179–3192, Nov. 2017, ISSN: 15249050. DOI: `10.1109/TITS.2017.2673778`.

[53] H. Oleynikova, D. Honegger, M. Pollefeys, "Reactive avoidance using embedded stereo vision for MAV flight," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, Institute of Electrical and Electronics Engineers Inc., Jun. 2015, pp. 50–56. DOI: `10.1109/ICRA.2015.7138979`.

[54] K. Cesare, R. Skeele, Soo-Hyun Yoo, Yawei Zhang, G. Hollinger, "Multi-uav exploration with limited communication and battery," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2230–2235. DOI: `10.1109/ICRA.2015.7139494`.

[55] S. Marangoz, M. F. Amasyalı, E. Uslu, F. Çakmak, N. Altuntaş, S. Yavuz, "More scalable solution for multi-robot–multi-target assignment problem," *Robotics and Autonomous Systems*, vol. 113, pp. 174–185, Mar. 2019, ISSN: 09218890. DOI: `10.1016/j.robot.2019.01.005`.

[56] N. Mathew, S. L. Smith, S. L. Waslander, "Planning Paths for Package Delivery in Heterogeneous Multirobot Teams," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, Oct. 2015, ISSN: 15455955. DOI: `10.1109/TASE.2015.2461213`.

[57] R. Williams, B. Konev, F. Coenen, "Multi-agent environment exploration with AR.Drones," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8717 LNAI, Springer Verlag, 2014, pp. 60–71, ISBN: 9783319104003. DOI: `10.1007/978-3-319-10401-0_6`.

[58] S. Liu, K. Mohta, S. Shen, V. Kumar, "Towards collaborative mapping and exploration using multiple micro aerial robots," in *Springer Tracts in Advanced Robotics*, vol. 109, Springer Verlag, 2016, pp. 865–878. DOI: `10.1007/978-3-319-23778-7_57`.

[59] T. Nestmeyer, P. Robuffo Giordano, H. H. Bülthoff, A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous Robots*, vol. 41, no. 4, pp. 989–1011, Apr. 2017, ISSN: 15737527. DOI: `10.1007/s10514-016-9578-9`. arXiv: `1505.05441`.

[60] Y. Dong, C. Fu, E. Kayacan, "RRT-based 3D path planning for formation landing of quadrotor UAVs," in *2016 14th International Conference on Control, Automation, Robotics and Vision, ICARCV 2016*, Institute of Electrical and Electronics Engineers Inc., 2016, ISBN: 9781509035496. DOI: `10.1109/ICARCV.2016.7838567`.

[61] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, A. Ng, "ROS: an open-source Robot Operating System," Tech. Rep.

[62] F. M. Noori, D. Portugal, R. P. Rocha, M. S. Couceiro, "On 3D Simulators for Multi-Robot Systems in ROS: MORSE or Gazebo?," ISBN: 9781538639221.

[63] *Publications List - Robotics and Autonomous Systems Group*. [Online]. Available: `https://research.csiro.au/robotics/our-work/publications/` (visited on 05/17/2020).

[64] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. Van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, P. Mahoney, "Stanley: The Robot that Won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006. DOI: `10.1002/rob.20147`.

[65] F. Furrer, M. Burri, M. Achtelik, R. Siegwart, “Robot operating system (ros): The complete reference (volume 1),” in, A. Koubaa, Ed. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625, ISBN: 978-3-319-26054-9. DOI: `10.1007/978-3-319-26054-9_23`.

[66] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, O. von Stryk, “Comprehensive simulation of quadrotor uavs using ros and gazebo,” in *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, 2012, to appear.

[67] *octomap - ROS Wiki*. [Online]. Available: `http://wiki.ros.org/octomap` (visited on 05/18/2020).

[68] L. Yang, J. Qi, J. Xiao, X. Yong, “A literature review of UAV 3D path planning,” in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2015, ISBN: 9781479958252. DOI: `10.1109/WCICA.2014.7053093`.

[69] D. V. Lu, D. B. Allan, W. D. Smart, “Tuning cost functions for social navigation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8239 LNAI, Springer, Cham, Oct. 2013, pp. 442–451, ISBN: 9783319026749. DOI: `10.1007/978-3-319-02675-6_44`.

[70] T. Kruse, A. K. Pandey, R. Alami, A. Kirsch, “Human-aware robot navigation: A survey,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013, ISSN: 09218890. DOI: `10.1016/j.robot.2013.05.007`.

[71] D. V. Lu, D. Hershberger, W. D. Smart, “Layered costmaps for context-sensitive navigation,” in *IEEE International Conference on Intelligent Robots and Systems*, Institute of Electrical and Electronics Engineers Inc., Oct. 2014, pp. 709–715, ISBN: 9781479969340. DOI: `10.1109/IROS.2014.6942636`.

[72] *Çoklu Mikro Hava Araçları için İç Ortamda 3B Haritalama ve Eş Zamanlı Navigasyon - YouTube*. [Online]. Available: `https://youtu.be/MGn1trFuADY` (visited on 06/30/2020).

PUBLICATIONS FROM THE THESIS

Contact Information: mdilaver@yildiz.edu.tr

Conference Papers

1. M. B. Dilaver, F. Çakmak, E. Uslu, M. F. Amasyali and S. Yavuz, "3D Mapping and Simultaneous Navigation for Multi Micro Aerial Vehicles in Indoor Environments", Signal Processing and Communications Applications Conference (SIU), 2020 (in press).