

Renan Sarcinelli

Quantized Multi-Height Occupancy Grid Map Applied to Airplanes

Vitória, ES

2021

Renan Sarcinelli

Quantized Multi-Height Occupancy Grid Map Applied to Airplanes

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Informática
da Universidade Federal do Espírito Santo,
como requisito parcial para obtenção do Grau
de Mestre em Informática.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Informática

Supervisor: Prof. Dr. Thiago Oliveira dos Santos

Co-supervisor: Prof. Dr. Alberto Ferreira De Souza

Vitória, ES

2021

Ficha catalográfica disponibilizada pelo Sistema Integrado de Bibliotecas - SIBI/UFES e elaborada pelo autor

S243q Sarcinelli, Renan, 1994-
Quantized multi-height occupancy grid map applied to airplanes / Renan Sarcinelli. - 2021.
f. : il.

Orientador: Thiago Oliveira dos Santos.
Coorientador: Alberto Ferreira De Souza.
Dissertação (Mestrado em Informática) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Robótica. 2. Inteligência artificial. 3. Veículos autônomos. I. dos Santos, Thiago Oliveira. II. De Souza, Alberto Ferreira. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

To my parents, Robinson and Leila, for all their support and encouragement.

Acknowledgements

I want to thank my supervisor, Prof. Dr. Thiago Oliveira dos Santos, for his guidance and advice. Also, the professors Dr. Claudine Badue and Dr. Alberto Ferreira as well as all the other members of LCAD for the constructive discussions that provided acknowledgement sharing and intellectual growth, their support and the jokes that keep pushing me forward.

I also want to thank Embraer for supporting technological cooperation with UFES. Acknowledge the scholarships supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, Brazil). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES). The authors would also like to thank the NVIDIA Corporation for their donation of GPUs to the lab.

“Something.”
(Author)

Abstract

The automation of cars has been widely researched and developed in the past years. Some of the technologies embedded in self-driving cars are specific for their environments, such as traffic sign recognition and lane marking detection. However, many of the technologies developed for self-driving cars are extensible to general robotic applications, directly or with some adjustments. Aiming for that opportunity, this work proposes a novel application of traditional two-dimension occupancy grid maps, widely used in self-driving cars, to the aircraft context of the taxiing operation. The main contributions are a new collision representation model and a mapping generation system. This approach handles the complex geometry of the airplane, where the wings and the body have different collision heights, the new system also introduces a more flexible collision representation model, that reduces the movement restrictions, compared to the actual system present at IARA. The proposed method also takes advantage of the two-dimensional localization, since the airplane moves basically in a 2D plane during this maneuver, which is simpler when compared to conventional three-dimensional localization and mapping systems. This work is part of an university-industry collaboration effort to unveil the technological challenges involved in automating such manoeuvres on an aircraft. The performance of the proposed method was accessed in partnership with Embraer S.A., as part of an integrated proof-of-concept solution embedded on a Praetor 600 jet, which was able to successfully perform an autonomous taxiing over a four miles closed circuit at a private aerodrome.

Keywords: Mapping. Grid Maps. Multi-Height. IARA.

Resumo

A automatização de carros tem sido alvo de muitas pesquisas nos últimos anos. Muitas das tecnologias desenvolvidas durante estas pesquisas são específicas para o contexto automobilístico, como reconhecimento de placas, sinais de trânsito, entre outros. Porém muitas outras destas tecnologias podem ser estendidas a outras aplicações com algumas modificações ou, às vezes, diretamente. Tendo em vista esta oportunidade, este trabalho apresenta uma nova aplicação dos mapas de grid de duas dimensões, tradicionalmente utilizados por veículos autônomos, no contexto aeronáutico para manobras de taxiamento. As maiores contribuições são um novo modelo de representação de colisão e mapeamento. O sistema proposto é capaz de tratar a geometria complexa de um avião, onde as asas e a fuselagem do avião possuem alturas distintas, a serem consideradas durante o tratamento de colisão, o novo sistema também introduz um modelo de colisões mais flexível, que reduz restrições de movimento, presentes no modelo atual da IARA. O método proposto também se aproveita da simplicidade da localização bidimensional, uma vez que durante esta manobra o avião se move basicamente em um plano 2D, sendo assim, bem mais simples em comparação com sistemas de mapeamento e localização tridimensionais. Este trabalho fez parte do projeto de cooperação tecnológica entre a Embraer e UFES para estudar e compreender os desafios envolvidos no taxiamento autônomo, sendo integrado em uma solução embarcada a uma aeronave Praetor 600 para realização de um taxiamento autônomo em um percurso fechado de aproximadamente quatro milhas.

Palavras-chaves: Mapeamento. Mapas de Grid. Multi-nível. IARA

List of Figures

Figure 1 – Sensor Box Position and sensors setup on Praetor 600.	13
Figure 2 – OGM generation explanation - the red dashed line represents the laser rays, when it touches an obstacle above the green dashed line only the Level 0 occupancy probability map is updated, otherwise both maps are updated.	15
Figure 3 – Working principle of a simple LiDAR. The distance between the emitter and receiver in the image was increased for a better visualization, in the real sensor this distance usually is as close as it can be disconsidered	18
Figure 4 – Occupancy Grid Map obstacle detection	20
Figure 5 – Obstacle Distance Map generation	21
Figure 6 – Overview of the autonomous system architecture of our self-driving car. The perception and decision making systems are shown as a collection of modules of different colors, where TSD denotes Traffic Signalization Detector, and MOT, Moving Obstacles Tracker. The modules that need to consider the multi-height maps are in darker colors.	25
Figure 7 – Multi-height occupancy grid map generation. First, the lasers reach the obstacles, then the points positions are calculated, and finally each map interprets it depending on its height of interest. Although the figure is not exactly as the method was implemented, it gives a better visual comprehension	28
Figure 8 – Aircraft collision model. The picture represents the aircraft collision model where the red circles represent collisions with the level0 map and the yellow ones, collisions with the level1 map.	30
Figure 9 – Simulated wing in the IARA. (a) Shows a front-left view of the car with the wing, notice in the shadow the increase in the car width. (b) shows a back-left view, where the car is heading towards obstacles of the circuit	33
Figure 10 – Simulated wing testing circuit. The map is the multi-height map built using the proposed method. The blue pixels represent unknown cells, the white one's free cells, the black pixels represent obstacles in both levels, and the light gray obstacles are just the level0 map. The green rectangle represents the starting point, while the circles the obstacles. The yellow ones are short obstacles in the wing's path, the orange are also short but in the car's path, and the red are tall obstacles	34

Figure 11 – Designed path for the airplane tests. In the airplane map, the red lines indicate the selected path ,the arrows indicate the directions and the green circle the start position.	35
Figure 12 – Simulated wing results. On each column, the upper image is a representation of the car in the map, the map components were described in the experiments section, the vehicle is represented by a white polygon, in this case a rectangle (the car) with an extended side (simulated wing). The next yellow polygons are the planned future poses and the blue and red lines are plans. The lower image is the picture taken by the car’s front camera. Both images, upper and lower, represent the same moment in time, in two different representations.	37
Figure 13 – Airport multi-height occupancy grid map (on the left), and remission map (on the right)	38

List of abbreviations and acronyms

LiDAR	Light Detection And Ranging
IARA	Intelligent Autonomous Robotic Automobile
LCAD	Laboratório de Computação de Alto Desempenho (High Performance Computing Laboratory)
OGM	Occupancy Grid Map
ODM	Obstacle Distance Map

Contents

1	INTRODUCTION	12
1.1	Motivation	13
1.2	Objectives	14
1.3	Proposal	14
1.4	Contributions	15
1.5	Structure	16
2	THEORETICAL BACKGROUND	17
2.1	LiDAR	17
2.2	Grid Mapping	19
2.3	Occupancy Grid Maps	20
2.4	Obstacle Distance Map	20
2.5	Remission Map	21
2.6	Related Works	22
3	QUANTIZED MULTI-HEIGHT MAPPING SYSTEM	24
3.1	Intelligent Autonomous Robotic Automobile (IARA)	24
3.2	Multi-Height maps	27
3.2.1	Multi-Height Occupancy Grid Map	27
3.2.2	Multi-Height Obstacle Distance Map	29
3.3	Collision Representation	30
4	MATERIALS AND METHODS	32
4.1	Simulated Wing	32
4.2	Airplane	34
5	EXPERIMENTS AND RESULTS	36
5.1	Simulated Wing	36
5.2	Airplane	37
5.3	Memory Consumption and Performance Analysis	38
6	CONCLUSIONS	39
	BIBLIOGRAPHY	40

1 Introduction

The past years have witnessed the emergence of several research studies towards the full automation of the main flight capabilities as Micro Aerial Vehicles (MAVs) became an ubiquitous experimental tool [1]. Nevertheless, sophisticated and robust autopilots are already present in almost all commercial airplanes playing a major role at climb, cruise and descent flight phases. Furthermore, automatic take-off and landing systems are also widely researched and adopted by the aeronautical industry. On the other hand, despite the increased development of autonomous vehicles, the taxi operation at airfields is still a manual and pilot-demanding process. This work therefore proposes a novel approach that exploits state-of-the-art mapping technology present in self-driving cars to deal with taxiing challenges, in particular, to take the aircraft geometry into account. Hence, making mapping-based navigation suitable for aircraft taxiing maneuvers.

Most of the navigation systems for autonomous cars rely on precise localization using prior maps of the environment. These maps are updated with online sensor data. Moreover, in order to reduce complexity, the three-dimensional (3D) world data are usually represented as two-dimensional maps. This is possible because cars are always in the ground plane, also the geometry of most vehicles can be simplified and represented as polygons in this plane, thus reducing the collision detection complexity, when compared to a 3D system.

This approach is suitable for controlled environments such as airports too. In a typical operation, an airplane needs to taxi-in to the runway or taxi-out to an airport finger at a *known* airfield, following predefined tracks and marks along the way, where the plane also moves only on the ground plane. However, since an airplane has a more complex structure, with parts such as wings and fuselage with varying ground clearances, some of them may collide with objects of different heights.

An intuitive solution to this problem is to use a 3D mapping approach commonly used for humanoid robots and quadrotors [2, 3, 4, 5]. However, these solutions have some drawbacks, e.g., they do not perform in real-time (at least 20 Hz), demand a high amount of memory to store all the 3D information, or are not updatable, thus limiting the representation of dynamic obstacles in the environment. Finally, none of them are as simple as the two-dimensional (2D) representation commonly employed by self-driving cars.

In this work, we propose an accurate and real-time updatable Occupancy Grid Maps (OGM) representation for collision detection at different discrete heights. For this method, we take advantage of the compression and optimizations of the 2D maps widely

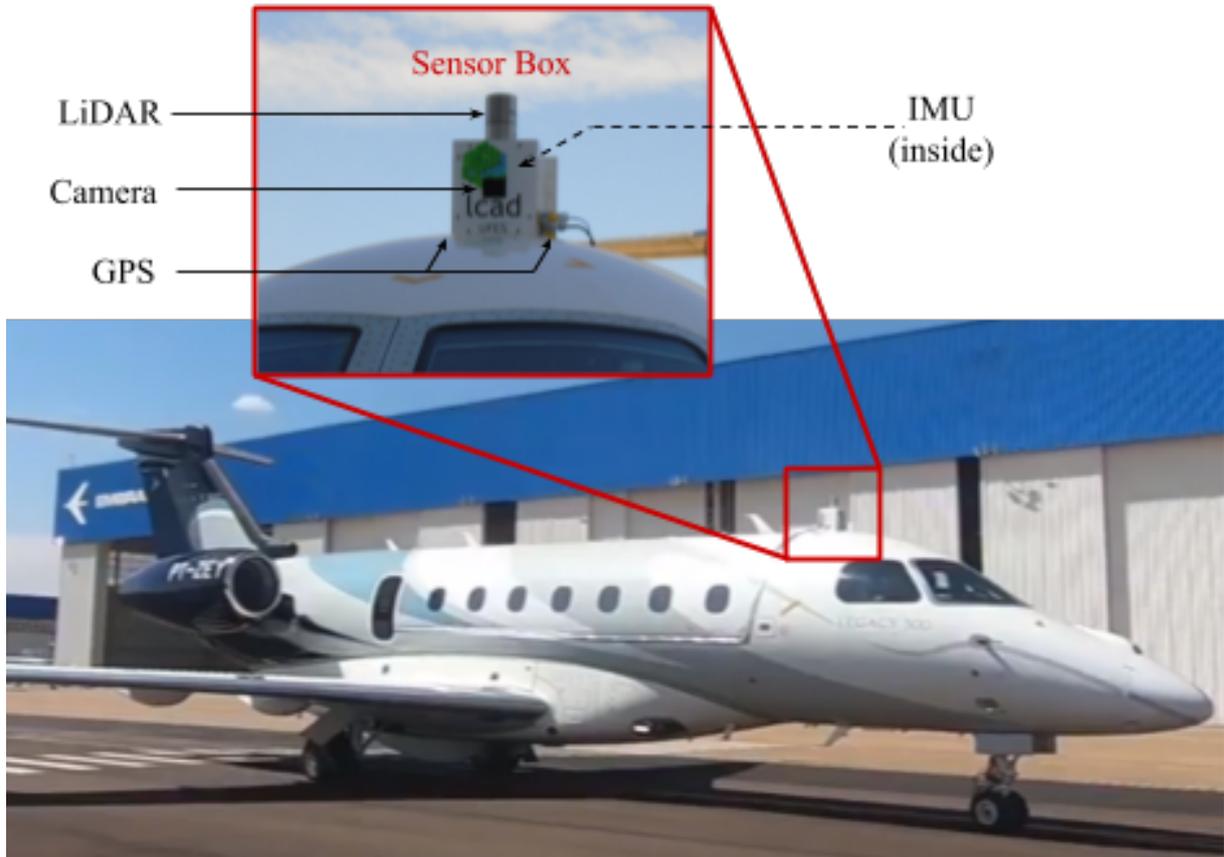


Figure 1 – Sensor Box Position and sensors setup on Praetor 600.

used in self-driving cars, by combining multiple 2D maps for 3D collision detection.

The proposed solution was then embedded on a Praetor 600 executive jet, Figure 1 – in a partnership with Embraer S.A. – alongside relevant modules of the Intelligent Autonomous Robotics Automobile (IARA) [6]. Some other modules were also adapted to have a proper behavior on the aircraft and, to the best of our knowledge, it was the first time that a Part 25¹ aircraft performed a fully autonomous taxiing maneuver following a predefined closed circuit.

1.1 Motivation

Three dimensional mapping and representation is not a novel topic in the literature, from extensions of the OGM [7, 8] to more sophisticated techniques, capable of lossless compression of complex three dimensional structures [9]. However, most of the work present in the literature requires substantially more computational efforts when compared to a traditional OGM, not only to build the map itself, but also considering other tasks that should be running on a self driving car, and demands the maps, *such as localization and planning*

¹ As stated by Federal Aviation Regulations (FAR), Part 25 defines specific airplane categories.

Although 3D mapping techniques have been present in the literature for quite some time, they are mainly addressed at applications in humanoids and quadrotors [2, 3, 4, 5], for this kind of applications there are not much changes in the environment and most of them consider the obstacles represented by this maps to be static. Self driving cars, on the other hand, require a map that must include dynamic objects, such as pedestrians and other vehicles. Due to the geometry of the self-driving car problem, most of the current research in the area addresses two dimensional maps[6]. Using two dimensions instead of three, reduces the complexity of localization and mapping algorithms.

Most of the 3D mapping techniques present in the literature are focused on a good representation of the world, however methods to handle vehicles with complex geometry while still taking advantage of the simplicity of the 2D maps are lacking in the literature.

1.2 Objectives

This work proposes to extend the current map and collision representation present at the Intelligent Autonomous Robotics Automobile (IARA), that is limited to two dimensional representation, to be capable of handling some parts of the vehicles with different collision heights, more specifically the wings and fuselage of an aircraft.

To accomplish this task, the specific objectives are:

- Proposes a new collision representation model of the vehicle, capable of handling complex vehicle structures and indicates different collision heights
- Proposes a new map creation system, to fulfil the collision representation model heights
- Integrates the complete proposed systems with the IARA architecture to be applied on a aircraft

1.3 Proposal

This work extends the collision representation and mapping system that present at the IARA self-driving car, to handle different vehicles, with more complex geometries, including a multi-height collision object such as the airplane.

Its important to notice the main focus of this work is to improve the IARA mapping and collision system to handle more complex vehicle's geometry, specially multi-height collisions, at the same time keeping advantage of the built-in IARA modules. Completely changing the mapping system from two to three dimensions would imply several changes in

the navigation, localization and path planning module, among others, that will drastically impact the overall system's performance.

Instead, the proposed method consists in a replication of the current occupancy grid map, present in IARA, but discarding points below a certain height, according to the desired geometry. This process can be repeated for any number of heights of interest to fit the desired vehicle geometry, but as long as this number increases the map structure becomes more similar to a 3D representation, but without optimizations for 3D map representation.

In summary, less height maps are always better for performance, with that in mind the optimal number of height maps is the lowest possible, as long as they can handle the necessary collisions safely. In this specific case where the vehicle is a plane, running on a taxiway, the fuselage could not go over any obstacle, on the other hand the wings must pass over some markings on the border of the taxiway, like traffic cones for instance. For these reasons the number of height-maps chosen for this scenario was one, not taking in account the ground level.

For this work specifically, the height of interest will be the height of the wings, generating two different maps, the map that is already present in IARA representing all the obstacles above ground level, this will be addressed as level 0 map, and the map above the wings level, that will be addressed as level 1 map, as shown in Figure 2.

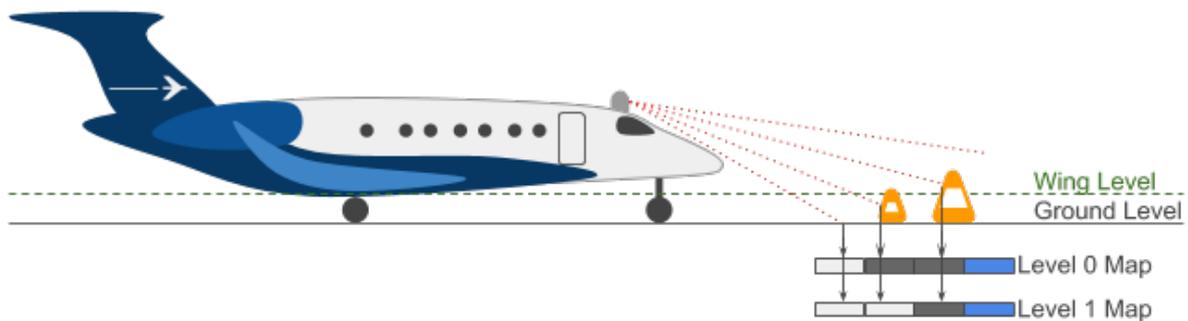


Figure 2 – OGM generation explanation - the red dashed line represents the laser rays, when it touches an obstacle above the green dashed line only the Level 0 occupancy probability map is updated, otherwise both maps are updated.

1.4 Contributions

The main contributions of this work are:

- A new collision system for IARA, capable of handling more complex geometries
- A new mapping system that handles different heights.

- A new method for annotating the 3D poses of traffic lights.
- A paper submitted to the International Conference of Robotics and Automation (ICRA), in partnership with EMBRAER.

Regarding to the main focus of this work, some other contributions were developed related to it:

- A paper published at the Computers & Graphics, named Handling pedestrians in self-driving cars using image tracking and alternative path generation with Frenét frames. That also improves the IARA's collision module besides many others.
- Integration of the Intel Realsense d435 depth image in the IARA system as a point cloud. This data was intended to be used to detect holes and the car will be represented with the wheels and chassi in different height levels.

1.5 Structure

The remaining of this document is organized as follows: the theoretical background is discussed in Section 2; the proposed method is described in Section 3; the materials and methods are presented in Section 4; the experiments and their results are shown and discussed in Section 5; and, finally, Section 6 concludes and discusses future work.

2 Theoretical Background

This chapter will give a brief introduction about occupancy grid maps, and other kinds of maps present at IARA that will be used in this work, as well as a better comprehension about the LiDAR, that is the main sensor used to build these maps. Then contrast some other works present in the literature that may be used for the presented situation.

The maps that will be presented in the next chapters are the occupancy grid map, the obstacle distance map and the remission map. The maps used for the autonomous system provide prior information about the environment, from a source that collected the information from that location. The autonomous vehicle IARA has a series of specific maps, so to guarantee the integrity and precision of the maps it has to previously go to the environment in non autonomous mode and build its own maps.

As human maps are not perfect, normally it contains static information that will not change very often, like buildings, and trees for instance, otherwise when someone uses the map it may have divergent information. However, when you go to a place previously mapped it may have some dynamic information, such as people and cars, that are not in the map, but you can mentally place them in the map according to the static information on the map. IARA does almost the same thing, when it runs through the map it will have prior information about the place, and updates its surroundings with new information, for this reason it's really important the maps used are easily and fast adaptable.

Next, will be discussed how the world can be represented in a map structure that can be used by a computer, usually the 3d information is condensed in a plane that is then divided in cells to be similar to a bi dimensional matrix representation, called grid map. However, many different pieces of information can be represented in a grid map. In this work, three different grid maps will be presented: The Occupancy Grid Map (OGM), that represents occupied, free and unknown cells, in the collision point of view. The Obstacle Distance Map (ODM), that indicates the distance to the closest known obstacle, based on the OGM. And the Remission map, that uses the reflected intensity from the LiDAR laser beam to determine if an object is dark or bright.

2.1 LiDAR

The light detection and ranging sensor or LiDAR is a very versatile sensor used basically to measure distances, this kind of sensor is widely used in autonomous systems because it can measure distances directionally, with low interference of environment light

conditions.

This sensor principle is quite simple, it emits a directional laser beam pulse and waits until it is detected. Considering then the light speed and the time between the sent pulse and the detection, the travel distance from the laser to the obstacle that it reaches, and return, can be calculated as depicted in Fig. 3. The limitations of this kind of sensor are, the kind of materials that can be detected, transparent or translucent objects that lets the laser beam passes through it will not be detected, dark materials also tends to absorb light, making them harder to detect, specially at long distances, also due to the high frequency of the laser when compared to sonars or radars the signal attenuates much faster, which gives to the this kind of sensor a shorter range of detection.

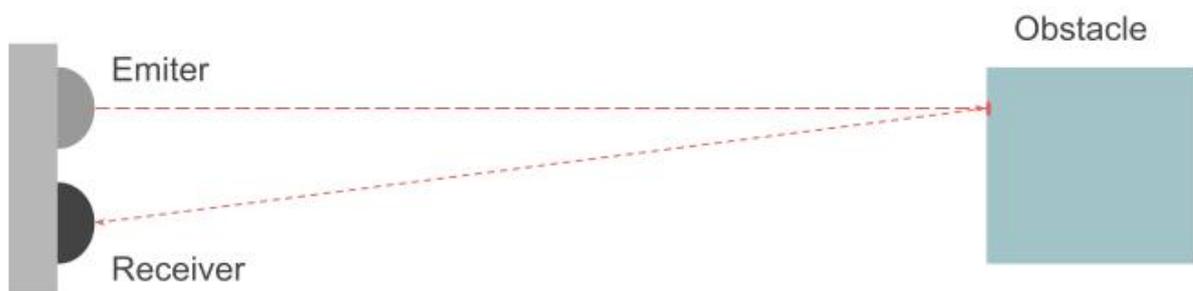


Figure 3 – Working principle of a simple LiDAR. The distance between the emitter and receiver in the image was increased for a better visualization, in the real sensor this distance usually is as close as it can be disconsidered

This kind of sensor as it was just explained, measuring distances in a single direction has many applications, one of the most common one is to measure aircrafts' altitude, but it does not give a good perception of the environment, as desired for self-driving cars. To give a wide perception of the environment many of these sensors could be placed in the surroundings of the vehicle, measuring the distance to objects around all the time. However, lots of these sensors may be expensive and require quite some power to run. Also, compared to the light speed that is used for the sensor to measure distances, the car moves much slower, generating a lot of unnecessary data.

Exploiting these characteristics the industry developed a sensor that uses the LiDAR technology to scan an area more efficiently. Instead of positioning sensors in different directions there is a motor that turns the laser while it shoots the pulses, in this way instead of gathering the distance in a single direction, a single laser can gather information from multiple directions, but it stills limited to a plane, or cone, ... Depending on your motor spinning direction and laser position. To spread this perception into a wider area, an array of Lasers are used, instead of just one, in which each laser is pointed with a different angle.

Instead of generating a single distance as a directional LiDAR does, this sensor generates a bunch of distances in different directions depending on the position that the

laser is pointed at and the current angle that the motor is. To represent this kind of data these distances are converted from spherical coordinates, to Cartesian coordinates and placed in a list containing the coordinates of each point, this data structure is named point cloud. Additionally to this point cloud information some LiDARs also provides the intensity of which the laser beams returned, this intensity can also be addressed as remission.

Due to the wide variety of manufactures of this kind of sensor and the popularity that it gained in the last years, it is addressed also as LiDAR. In this work we will be using a LiDAR that contains 32 laser beams, and turns 360 degrees. It is placed in the top of the vehicle, in this way a single sensor can give a better perception of all the surroundings.

2.2 Grid Mapping

Most of the autonomous vehicles systems nowadays assume a previously mapped environment [6]. Since these vehicles usually consider any obstacle as a potential collision to any part of the vehicle, a 2D representation of the environment is usually suitable for these situations.

To generate this representation the system projects the 3D information in a plane (each mapping method can condense different information in a distinct way), that is usually the ground. This plane is then divided by 20cm x 20cm squares, forming a grid representation, very similar to a matrix.

To build these maps the vehicle must run in the desired environment, without any previous information, to collect the data. This process is usually made manually(not in autonomous mode). The odometer and GPS data is fused to obtain a more precise localization. All the other sensor data, such as the LiDAR and IMU, are then synchronized according to the slowest sensor, generating packages of information from all sensors for all the courses. All this information, from the complete run, is combined to determine the car's pose during all the course for each sample, according to the synchronization [10]. The vehicle's pose is essential to merge information from a sample with the overall map.

With all this information gathered in the manual run of the car in the course a grid map can be built, and is usually named as prior map. This first map is important to provide extra information to the vehicle in its consecutive runs through this course in autonomous mode. Since the car is now self-driven it can no longer run through all the course to determine its poses, the prior map is then included to improve the localization system [10].

Although each map has a different way and can use different sensors to be built, all of them use the same principle to obtain the vehicle's pose and analyze the sensor's

information synchronized with that pose to build it.

2.3 Occupancy Grid Maps

The most common information to be in any autonomous system is the occupancy in the environment. Determining which cell is occupied or not is crucial for any autonomous system to choose the most suitable path without any collisions. Therefore, a widespread representation technique is to build an Occupancy Grid Map (OGM). Proposed by Moravec & Elfes [11], an OGM consists of a fixed spaced discretization of the environment; these little squares of the world are named cells and each of them indicates the probability that the space is occupied.

The sensor responsible for detecting the obstacles at the IARA autonomous system such as in many other self-driving cars [6] is the LiDAR. Assuming that all the lasers are vertically aligned each sample of the LiDAR on the 360 degrees turn is on a same plane, since the angle between two consecutive laser beams in one of this samples are known, using the cosine law and an arbitrary laser measurement, the distance of the next beam to be at the same plane of this one can be calculated, and then compared to the actual measurement to determine the presence of an obstacle in this position [10], as shown in figure 4

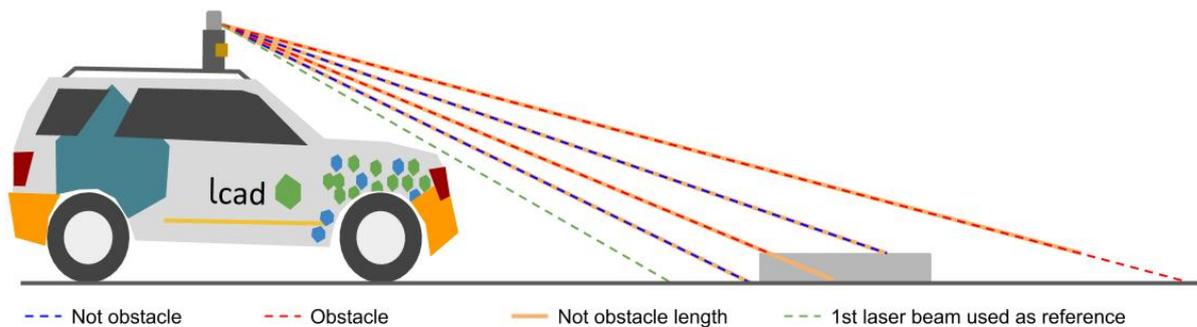


Figure 4 – Occupancy Grid Map obstacle detection

2.4 Obstacle Distance Map

The Obstacle distance map (ODM) present in the IARA system is essential to speed up the collision calculations. Instead of checking the distance for each cell of the OGM on demand, the system calculates all distances to the closest obstacle for each cell previously, based on the current and updated OGM, and stores this information on a map structure.

This process would be computationally demanding if done for each cell individually.

However, the system uses a dynamic programming technique to make this process much faster.

This map is generated based on the OGM, because of that it inherits the OGM cell structure and size, but instead of representing the presence of obstacles each cell is composed by a number representing the distance to the closest object in meters. To initialize this map a copy of the OGM is made filling all the cells with infinity, as depicted in Figure 5 Initialization, this means that there is no obstacles in the map, after that the algorithm passes to all the cells on the map two times. In the first pass from the top left corner to the bottom right corner verifying the presence of obstacles in the neighborhoods of the OGM, if any was found it takes the value from the neighbor closest to any obstacle and adds the distance from it to this neighbor, Figure 5 First Pass. The second pass applies the same process, but this time, from the bottom right to the top left, and taking in account the values obtained in the previous pass, as shown in Figure 5 Second Pass. With this technique the obstacle's distance for any cell in the map will be computed with a linear processing effort according to the map size.

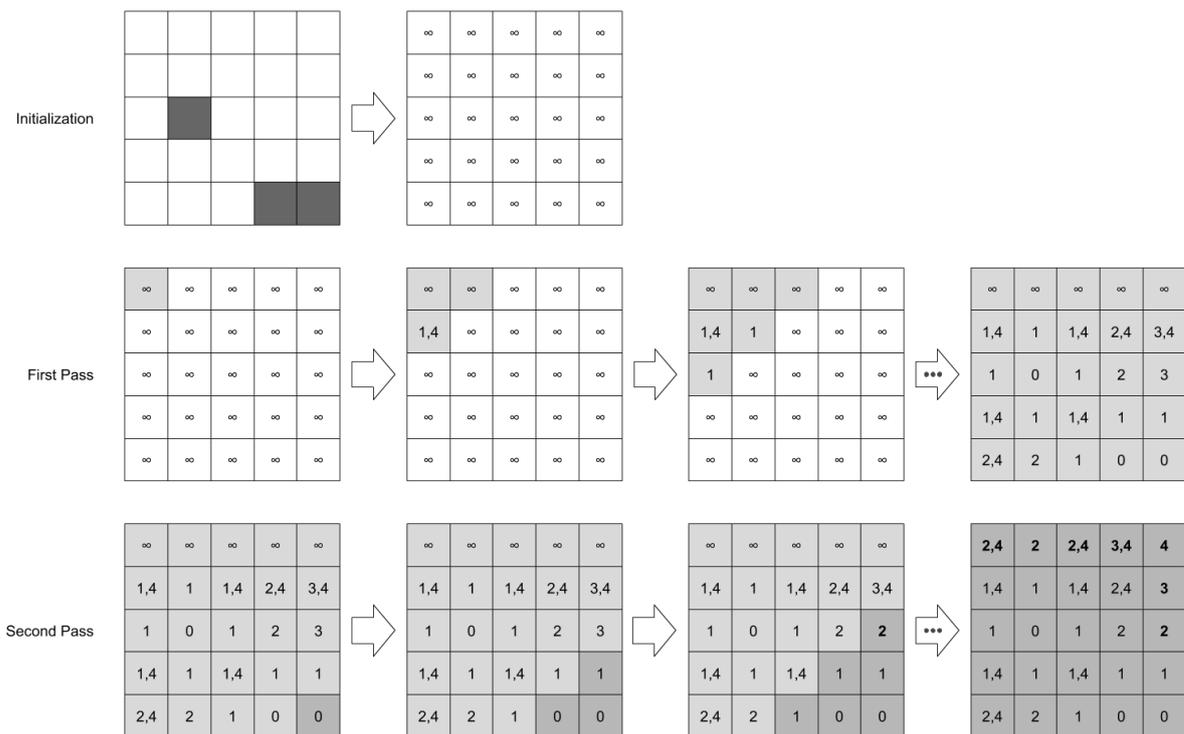


Figure 5 – Obstacle Distance Map generation

2.5 Remission Map

The LiDAR sensor is known for providing a precise point cloud of the environment by emitting laser beams and calculating the travel time for this beam to hit an object and

return, however this is not the only information provided by this sensor, most LiDARs are also capable of measuring the intensity of the beam that returned, providing information about the color that this laser reached, this is called the laser remission information.

With this information a different map can be created, since dark surfaces reflect less and white ones more, the remission map is very similar to a grayscale version of the environment. This kind of map is especially useful for localization, since most of the surfaces are not plain in texture, even in an open field, where there is no obstacle present on the OGM.

2.6 Related Works

In the scenario depicted by this work, some obstacles may not collide with some parts of the aircraft. A traffic cone, for instance, may be an obstacle for the aircraft land gears, but not for its wings. Taking this issue into account, we propose a way of representing the relevant obstacles according to different parts of the vehicle. Other applications, such as humanoids and quadrotors, require a complete volumetric view of the environment to operate autonomously [3, 4]. Therefore, these applications usually have a 2.5D or a complete 3D representation of the environment.

Y. Roth-Tabak and Ramesh Jain [7], applied the OGM in three dimensions, dividing the environment into cubes which contain the occupancy probability of each cell in the 3D environment. Next, updated probabilities are used to classify the corresponding cells as free, occupied, or unknown. The presented approach is effective and easily updatable; however, it needs a large amount of memory that grows cubically with the size of the mapped space volume.

A less memory-demanding OGM technique, named elevation map, is proposed by Miller & Campbell [8] in which each cell stores the maximum occupied height as if it is a void or unknown cell. This approach fits well for terrain mapping and contour lines, but environments with bridges or other surfaces with horizontal holes can not be well represented by this system. To overcome these issues Rivadeneyra et al. [12] proposed an extension for the elevation maps, improving its 2.5D representation to a full 3D representation called Probabilistic Multi-level Surface Maps. Similarly to OGM-based elevation maps, their approach consists of probabilistically estimating both the occupancy state and the height of a cell in the map. However, the proposed method estimates the occupancy at multiple heights, i.e. each cell can contain one or more patches that indicates the occupied height(s).

Dryanovski et al. present a remarkably similar approach named Multi-Volume Occupancy Grids [2] which is in turn fast updatable. Unlike other methods, this map can

be updated in real time, with less memory consumption, making it very suitable for MAVs. The method also makes a probabilistic analysis of the sensor data to determine a list of occupied heights for each cell in the grid.

To overcome these problems, Hornung et al. [9] proposed a probabilistic flexible and compact map named OctoMap. This map exploits octrees to create a lossless, compact map that is also capable of fully representing any 3D scenario with no previous assumptions about the surrounding environment. However, even with all the described characteristics, updating this map is not a trivial task making this process highly computationally demanding to run in real-time. Another interesting approach that also exploits non-fixed discretization of the environment is presented by Ryde and Hu. They proposed a novel multi-resolution algorithm which aligns 3D sensor data and creates a multi-resolution voxel list representing the environment.

In contrast to previous works on multi-purpose 3D representations, our task can be further simplified by considering rigid constraints of the taxiing operation scenario. Specifically, we can assume that the aircraft will always move in the same plane as the 3D space. This paper thus proposes a novel approach that takes advantage of the less complex and lightweight 2D OGM map characteristics, and at the same time, handles collision avoidance considering different heights for each part of the vehicle.

3 Quantized Multi-Height Mapping System

The next sections describe: the IARA platform (Section 3.1), the autonomous car on which the proposed approach was tested, and the software base for the tests on the aircraft; the multi-height mapping proposed (Section 3.2), which includes all the maps mandatory for the system works properly; and, finally, the collision system (Section 3.3).

3.1 Intelligent Autonomous Robotic Automobile (IARA)

Our self-driving car (Figure 6) is named Intelligent Autonomous Robotic Automobile (IARA) and it was developed at Laboratório de Computação de Alto Desempenho (LCAD) from the Federal University of Espírito Santo (Universidade Federal do Espírito Santo – UFES). The car is a Ford Escape Hybrid, which was adapted to enable electronic actuation of the steering, throttle and brakes, and to provide power supply for computers and sensors. These computers and sensors include a workstation (Dell Precision R5500, with 2 Xeon X5690 six-core 3.4GHz processors and one NVIDIA GeForce GTX-1030), networking gear, two LiDARs (a Velodyne HDL-32E and a SICK LD-MRS), three cameras (two Bumblebee XB3 and one ZED), an IMU (Xsens MTi), and a dual RTK GPS (based on the Trimble BD982 receiver).

The architecture of the autonomous system of our self-driving car can be roughly divided into two main parts: the Perception System and the Decision-Making system. The Perception System is currently divided into four main modules responsible for the tasks of localizing the self-driving car, mapping of obstacles, detection and tracking of moving obstacles, and detection and recognition of traffic signalization. The Decision-Making system is currently divided into six main modules responsible for the tasks of route planning, path planning, behavior selection, motion planning, obstacle avoidance, and control.

Figure 6 shows a block diagram of the autonomous system architecture of our self-driving car. The Mapper module [10] constructs occupancy grid maps [13] that represent obstacles of the environment, as long as other maps, such as the remission map, among others. The Mapper operates in offline and online modes. In the offline mode, the Mapper receives input data from multiple sensors (odometer, LiDAR, IMU, and GNSS), which are stored while our self-driving car is driven by a driver along a path of interest - this only needs to be done once. It then estimates the self-driving-car's states along the path and the values of the offline map cells (Offline Map) using the GraphSLAM algorithm [13, 10]. In the online mode, the Mapper receives as input the offline map, the sensors' data

(odometer, LiDARs, and IMU) and the car's state, and computes the online map. The online map is a mix of information present on the offline map and a grid map computed online using the sensors' data and the car's state. In the online mode, the Mapper also computes the Obstacle Distance Map (ODM), in which each cell contains the distance to the nearest obstacle [6].

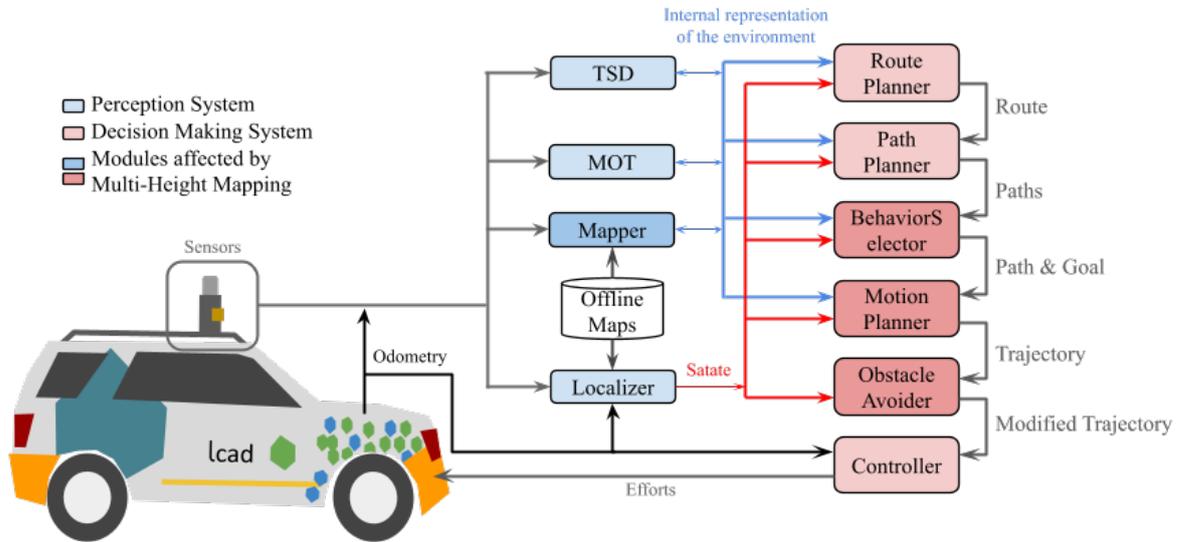


Figure 6 – Overview of the autonomous system architecture of our self-driving car. The perception and decision making systems are shown as a collection of modules of different colors, where TSD denotes Traffic Signalization Detector, and MOT, Moving Obstacles Tracker. The modules that need to consider the multi-height maps are in darker colors.

The Localizer module [14] estimates the self-driving car's state in relation to the offline map (that can be an OGM or a remission map). When initialized (global location [13]), the Localizer receives as input the initial state of the car from the GNSS RTK an approximation of the initial state of the car can also be provided manually. At each subsequent stage, it receives as input the last state of the car, the offline map, and the sensors' data (odometer, LiDAR, and IMU), and calculates a local grid map and estimates the car's state, by combining the local map with the offline map, using for this a particle filter.

The Moving Obstacles Tracker (MOT) module receives the ODGM, the self-driving car's state and a path segment, and calculates the pose and velocity of the nearest moving obstacles being tracked. The MOT also handles obstacles in motion by altering the longitudinal velocity of the self-driving car in order to (a) maintain adequate distance from obstacles in motion and (b) allow a smooth steering experience. In addition, it changes the ODGM seen by other modules of the self-driving car, in order to hide obstacles in motion from then on, which significantly simplifies its implementation.

The Traffic Signalization Detector (TSD) [15, 16, 17] receives the sensors' data

(camera and LiDAR) and the self-driving-car's state, and detects the position of traffic signalizations and recognizes their class or status. Traffic signalization includes horizontal traffic signalizations (lane markings) and vertical signalizations (speed limits, traffic lights, etc.) that must be recognized and obeyed by self-driving cars.

The Path Planner creates a path from the current state of the self-driving car to a local goal state. The Path Planner receives the car's state and a Road Definition Data File (RDDF) file as input. The RDDF is composed of a sequence of car's states, which are stored while our self-driving car is driven by a human along a path of interest. The Path Planner extracts a path from the RDDF, which consists of a sequence of equally spaced states that goes from the current state of the car to a goal state a few meters ahead.

The Behavior Selector module establishes a local goal state in the path according to the real-world instantaneous scenario. The Behavior Selector receives the ODM, the self-driving car's state, the path, and a set of annotations (made previously) from the offline map, such as speed bumps', safety barriers', crosswalks' and speed limits' positions, as well as traffic lights' positions and states. It then sets a local goal state along the path a few seconds ahead of the current state of the car, and adjusts the pose and velocity of the goal, so that the car behaves according to the instantaneous scenario, such as stopping at traffic lights, and reduces velocity or stop on busy crosswalks, etc.

The Motion Planner module [18] calculates a trajectory from the current state of the self-driving car to a local goal state in the path. The Motion Planner receives as input the ODM, the car's state, the path and the goal state in the path. It then calculates a path from the current state of the car to the goal state in the path that follows the path and avoids the obstacles represented in the ODM using a predictive approach. The trajectory consists of a sequence of control commands, each composed of linear velocity, steering angle and execution time.

The Obstacle Avoider module [19] checks and eventually changes the trajectory if a collision is to be avoided. The Obstacle Avoider receives as input the ODM, the self-driving-car's state and the trajectory. It then simulates the execution of the trajectory along the ODM. If the trajectory collides with an obstacle, the Obstacle Avoider decreases the velocity commands of the trajectory to avoid the accident.

Finally, the Controller module [20] calculates the commands that will be sent directly to the actuators of the self-driving car in the steering wheel, accelerator and brakes, employing a Proportional Integral Derivative (PID) controller. The Controller receives as input the path and the odometer data.

The Modules affected by the multi-height mapping proposed by this work are marked in darker colors. The Mapper module, that needs to create multiple OGMs and ODMs. The Behaviour Selector, Motion Planner and Obstacle Avoider, now have to

consider multiple ODMs for different parts of the vehicle, according to the current collision representation to calculate if a collision will occur, and take appropriate actions.

3.2 Multi-Height maps

As described in the previous section, IARA has an exclusive mapping system and builds their own maps, but not all of them are relevant to consider the collisions. This section will describe the main maps in the IARA system that are required for the novel collision system.

3.2.1 Multi-Height Occupancy Grid Map

The main motivation for changing the actual mapping system of the IARA is to include the capability of dealing with different collision heights. The main map used for collision treatment is the ODM, described in chapter 2.4, but this map relies on information provided by the OGM.

To build an ODM capable of dealing with different heights the OGM must be adapted to provide this information, or a different mapping system, like the ones presented at chapter 2.6, that provides height information, could be adopted. The second solution will imply several changes in the current IARA system, such as the localization system for instance, including complexity to the overall system. To keep the simplicity of the 2D localization system, among other modules, this work proposes a different approach to represent obstacles in multiple height levels.

The multi-height occupancy grid map consists of a combination of multiple OGMs, in which each map represents a different height level, representing only obstacles relevant to this height. Depending on the way that the LiDAR data is filtered to build the map there are two possible representations. In the first representation, a strict range could be used for all different levels, so that the maps are able to represent caves and bridges, more similarly to a 3D representation. However, parts of the vehicle that collides with more than one range must check multiple maps for collisions. In the second approach each map represents all obstacles above a certain height of interest, similarly to a 2.5D map, structures with hollows cannot be represented correctly, but, since the lower maps have all the information from the ones above, and additional information of lower obstacles, the collision can be checked in only one map.

For the specific application of this work, the plane in a taxiway, the second solution will be more suitable, since there will be no structures with holes in the environment, this solution will reduce the collision checks. This choice also benefits other modules. The lowest height level map will contain all the collisions above ground, like the actual OGM of

the IARA system, this is also the only mandatory map, because of that it will be addressed as level zero map. This characteristic of this map also ensures that other modules, not related to collision verification, that depend on the OGM will keep functional if they use only the level zero map.

The main concerns, related to the collision, in the plane scenario will be the fuselage and the wings. This makes necessary at least two height levels, above ground, that will be used for the fuselage, and just above the bottom of the wings that will be addressed as wing level and the respective map will be called level1 map. Using this method more maps could be created to treat other plane parts with different heights separately, however, it will increase memory consumption and processing load. Since two maps are the minimum necessary to solve the scenario depicted by this work, this was the number of maps chosen.

These maps are built following the same principle as the actual OGM map is built [10]. It also builds offline maps for each height of interest, although only the level zero map is used for localization, all maps are used as base occupancy maps and are updated in real time. The main difference in the multi-height approach is when a collision is detected. After comparing two consecutive rays, using the cosine law, to determine if they are in the same plane or not, the position of the second point can be calculated in spherical coordinates, in respect to the sensor. Given the distance, angle between lasers and rotation angle, these coordinates can be directly converted to cartesian coordinates, and then transposed from the sensor's system of coordinates to world coordinates. With this information the height of the collision is now known. Therefore, all the height maps will now process this same information according to its height of interest. If the collision occurs above its height level, the point will be considered as a normal collision and added to the map, if it's below, the point will be considered in the same plane as the point before it, ignoring the cosine law decision and considering it as a flat space. Fig. 7 and Fig. 2 gives a visual comprehension of the method adopted in this work

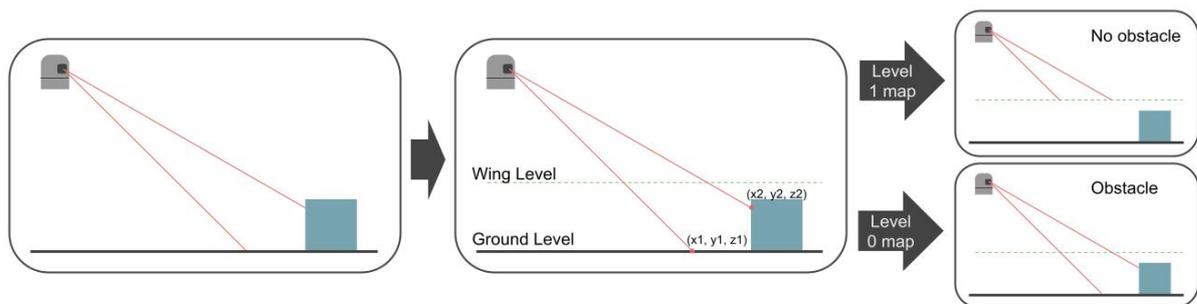


Figure 7 – Multi-height occupancy grid map generation. First, the lasers reach the obstacles, then the points positions are calculated, and finally each map interprets it depending on its height of interest. Although the figure is not exactly as the method was implemented, it gives a better visual comprehension

3.2.2 Multi-Height Obstacle Distance Map

The obstacle distance mapper was created in order to reduce the number of comparisons during the collision checks. Instead of calculating the distance to the nearest obstacle each time that is needed, the nearest distances from each cell are pre-calculated and stored in a map structure. Using this map, the necessary distances for collision checks are already calculated and can be obtained directly from the map cell index.

In the current map structure, OGM is the most important map, being used for many modules, including to build the ODM. However, when considering only the collision treatment the ODM is the key for the efficiency, even with complex vehicle structures.

One of the challenges when adapting the IARA system to the plane, is an efficient collision calculation even when the vehicle has a complex geometry. Fortunately the current IARA mapping system already has this solution, however, this is not applicable for collisions in different heights. Now the challenge is how to add the multi-height capability in the current system, in a way that it takes advantage of the fast and efficient collision calculations even when the vehicle have a complex geometry

One of the possible solutions to solve this problem, maybe the more intuitive one, is using a 3d grid map, as mentioned in chapter 2.6, as a base map. Then extending the idea of the 2D ODM, a 3D ODM will be created, using the same algorithm described at chapter 2.4, that now has to fill a 3D matrix. Naturally, in this 3D representation, the obstacles will be treated with spherical collisions, instead of circles. This solution will create a much more precise map, in the height aspect, with a drawback of significantly increase in the memory consumption.

Another solution would be building the ODM from a 2.5D map. However, creating a 2.5D ODM, that stores the closest obstacle distance and its height, will not work. When a vehicle part has a higher collision height than the closest collision obstacle, the map will not provide any information about the other heights. However, this kind of map can be used to generate multiple OGMs, in the 2D grid structure, as proposed by this work.

When looking at the proposed Multi-Height Occupancy Grid Map, and the actual IARA system, this solution just follows the pipeline. For each OGM an ODM is created, generating as many maps as heights of interest. Each part of the vehicle will address a different map, according to its collision representation. However this solution is not limited to occupancy grid maps, the same ODMs could also be generated from 2.5D maps, for instance. In this case, a variant of the algorithm described in chapter 2.4, could be used to generate the maps, the main difference is that when comparing the distances from the base map, that in this case will be a 2.5D map, the height should be considered to eliminate obstacles lower than the height of interest.

3.3 Collision Representation

To take direct distances using the Obstacle Distance Map, the vehicle collision area is represented by circles. Each circle is represented by the position of the center, related to the vehicle yaw axis, a radius and the height level. Fig. 8 shows the airplane collision model.

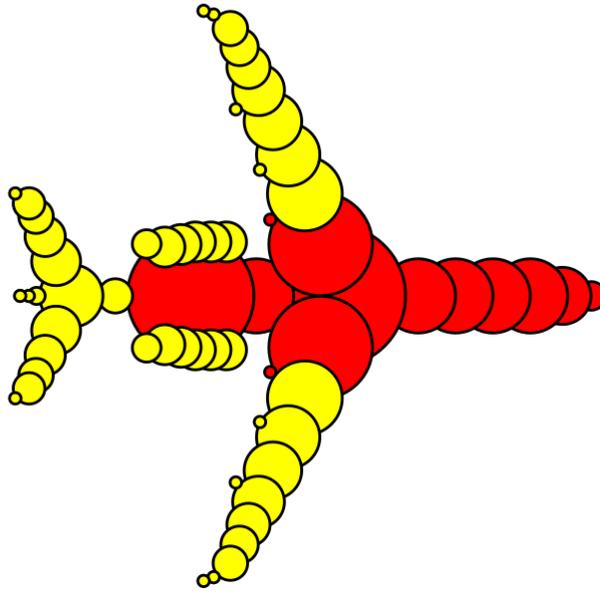


Figure 8 – Aircraft collision model. The picture represents the aircraft collision model where the red circles represent collisions with the level0 map and the yellow ones, collisions with the level1 map.

The proposed architecture allows two different interpretations of heights levels leading to distinct world representations. The first one considers each level as the obstacles between the last height of interest and the next one, the second one considers all the obstacles above the current height of interest. Each of these interpretations has its own advantages and this decision will also reflect on the map construction.

The biggest advantage of the first approach is a more accurate representation of the environment, since this map can represent tunnels and suspended obstacles. The disadvantages, however, are that: if a part of the robot will collide with multiple height levels this part should be represented by multiple circles, one in each height level; and, to obtain a map with the collision information from all levels the maps must be merged.

The second interpretation will not represent tunnels nor floating objects. On the other hand, the lower maps will have all the information from the above. In this way, the collision circle that is on a lower height level will check collisions with all the obstacles above. Since the aerodrome is an open field, this representation is more suitable for the application proposed by this paper, in order to reduce the collision checks. Another positive point of this interpretation is that the lowest map will have all the collision information

from the other maps, very similar to the map that IARA already uses for localization, making this representation compatible with the current localization module [6].

The main drawback of this representation is the approximation of straight lines. When compared to a polygonal representation that can define a straight boundary by two points, the circle needs to combine some elements to achieve a suitable approximation. The car itself, which can be represented by a rectangle, was represented by 5 circles in this model. And the plane, which has a much more complex geometry, was represented by 62 circles, with a wide variation of radius. Although it increases the number of elements for the collision check, the ODM has previously calculated all the closest obstacles distances.

4 Materials and Methods

To evaluate the proposed method our solution was tested in two main scenarios: a parking lot and a private aerodrome. The first scenario was set for safety reasons, since we must guarantee the exclusive use of the aerodrome during experimental tests with the Praetor 600 jet. However, in the second scenario, the presence of obstacles in the taxiway was continuously monitored by the system as e.g., other vehicles along the taxiway or even foreign object debris on the runway are not that rare in a typical airfield. Then, in order to test the obstacle avoiding system, a simulated wing was fixed at IARA and some tests were performed on an empty parking lot using traffic cones and wheeled containers as obstacles to ensure that the obstacle avoider was working properly. This test setup will be addressed as a simulated wing. In the second scenario, at an aerodrome, the system was adapted to a Praetor 600 jet in partnership with the Embraer S.A., this experiment will be addressed as an airplane.

To perform the experiments, the mapping and localization system of the IARA platform were adapted according to this paper. As all other modules and previous works based on this platform, the full code is open source and available at https://github.com/LCAD-UFES/carmen_lead.

The hardware used in both experiments were partially identical. The sensors were assembled on a single unit that will be addressed as Sensor Box, containing: differential GPS system; an Inertial Measurement Unit (IMU); A 8 Megapixels camera and a high-resolution 3D LiDAR; Additionally a single board computer, and a four port simple switch were needed to provide an Ethernet interface for all the sensors in the box, as shown in Fig. 1. The IARA has its own processing unit embedded, composed of a Dell Precision R5500, with 2 Xeon X5690 six-core 3.4GHz processors and one NVIDIA Titan X. For the experiment on the aircraft, a Notebook Dell G5 (Intel Core i7 8th Gen 8750H 2.2GHz and Nvidia GeForce GTX 1050 Ti) was used. Fig. 1 details the sensor box structure and sensors' positions, as well as its position on the aircraft. It is known that for a final solution a higher number of sensors has to be used to cover the whole plane, but as a proof of concept experiment, a single Sensor Box was enough.

4.1 Simulated Wing

Adapting an airplane is an expensive and sophisticated task that must be carried out by experienced specialists to ensure that all critical systems of the plane remain functional to safely operate during and after the tests. In this sense, the Embraer S.A.

team, as designers and manufactures of the Praetor 600 jet, was crucial for this operation. Also, the taxiing tests required the exclusive use of an entire aerodrome for some hours, and fortunately Embraer S.A. also disposed of a huge, well equipped testing facility. With these issues in mind, before starting the experiments on a real aerodrome, a more viable setup was mounted on the IARA in order to guarantee a proper functionality of the system.

In this experiment, a wooden board was fixed at IARA to simulate the plane wing (i.e., a fake wing). The board was placed in the front of the car, 0.7 m above the ground, and extended the vehicle width by approximately 1 m. Since it is not a usual item of the car, the roads are not prepared for this vehicle width, and the safety driver also must be prepared for unexpected situations. To reduce risks, only one wing is enough to prove the concept of the system. Fig. 9 (a) and (b) shows the vehicle, with the simulated wing, from different angles.

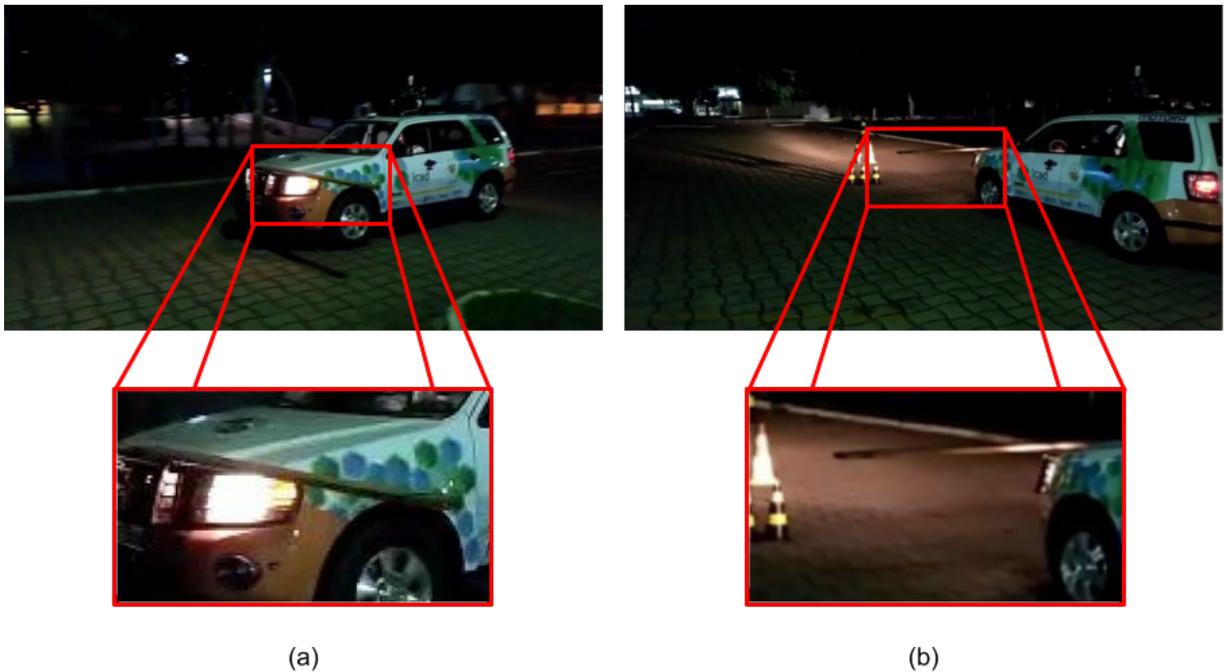


Figure 9 – Simulated wing in the IARA. (a) Shows a front-left view of the car with the wing, notice in the shadow the increase in the car width. (b) shows a back-left view, where the car is heading towards obstacles of the circuit

The test was made on a wide parking lot controlled environment with enough space for safe maneuvers. In this experiment, the car should go through a circuit in the parking lot. First, using the approach described at Section 3.2, a Multi-Height Occupancy grid map of the environment was built. After that, six obstacles were placed in the car's path to test specific functionalities.

The first two are short obstacles (only identified by the level0 map), placed outside the car's path but in the fake wing's path, represented by the yellow circles in Fig. 10. The expected behavior is that the car continues its normal path, since the obstacles will not

collide with the fake wing. The next two obstacles are also short, but this time placed in the car's path, represented by the orange circles in Fig. 10. According to the flight safety rules, the airplane case should stop in these situations. However, to allow continuous testing on autonomous mode without changing the environment, this behavior was changed during the experiments with the car to deviate from the obstacle and finish the planned trajectory. Finally, the last two obstacles were tall obstacles (more than 0.7 m and, therefore, visible on level1 map), placed only on the wing's path and represented by the red circles in Fig. 10, like the previous obstacles, they should be identified and avoided. Fig. 10 illustrates the described scenario.

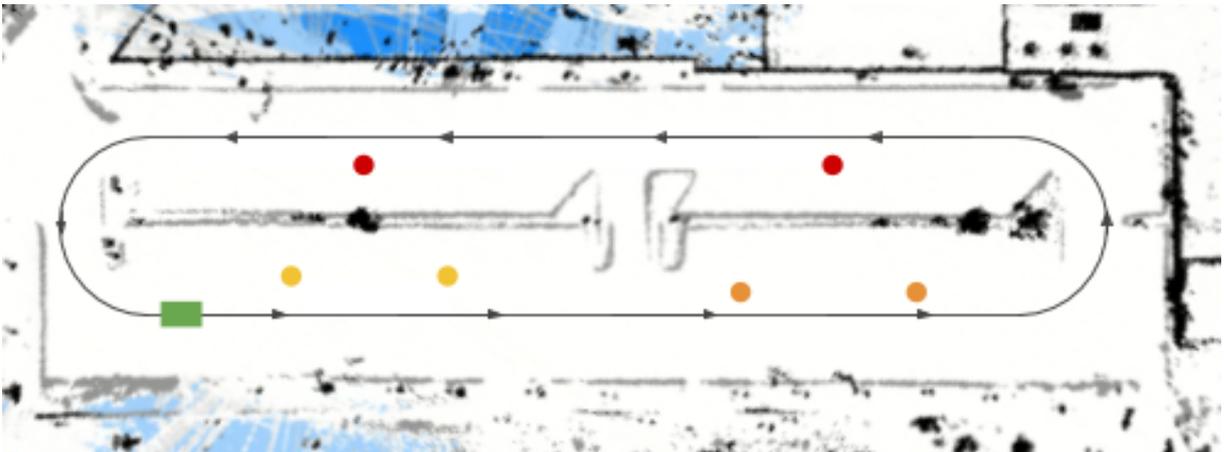


Figure 10 – Simulated wing testing circuit. The map is the multi-height map built using the proposed method. The blue pixels represent unknown cells, the white one's free cells, the black pixels represent obstacles in both levels, and the light gray obstacles are just the level0 map. The green rectangle represents the starting point, while the circles the obstacles. The yellow ones are short obstacles in the wing's path, the orange are also short but in the car's path, and the red are tall obstacles

4.2 Airplane

The final experiment was made after the simulated wing tests succeeded. In this experiment the Embraer S.A. expertise and knowledge about the planes were fundamental for equipping the aircraft with the necessary sensors, mainly a GPS and LiDAR in the Sensor Box. As well as for the integration between the commands generated by the IARA system and the complex airplane actuation systems, ensuring that the safety system as the autopilot remains operating properly.

For this experiment, a four miles closed circuit path was selected in the airport simulating a taxiing maneuver. The plane then should keep in this path, passing the wings above the markers and obstacles in the path when necessary. Fig. 11 illustrates the chosen path; the aircraft ran the selected path three times.

A challenging aspect of this experiment is that the aerodrome is approximately a flat, wide road in a plan and clear field. In this sense, even the level0 map has few features represented, making the localization process particularly hard. To overcome this problem the remission map was used to localize the vehicle instead of the grid map [21].



Figure 11 – Designed path for the airplane tests. In the airplane map, the red lines indicate the selected path, the arrows indicate the directions and the green circle the start position.

5 Experiments and Results

This section will present and discuss the results obtained by the proposed method in both scenarios.

5.1 Simulated Wing

In these experiments, the car ran seven laps on the described circuit. On all laps, the car passed through the first two obstacles with no issues. In the two following obstacles, the car performed as expected but, sometimes, it slowed down a bit. In the last two obstacles (the tall ones), the car also deviated as expected, meaning that the obstacles were seen as they should. Fig. 12 illustrates an example of each of these situations. All the experiments were made at night to avoid other vehicles, albeit the lightning conditions should not have much impact on the results, since the mapping system relies on LiDAR data.

The first two obstacles were the simplest issues. As expected, the purpose of this set of obstacles was just to ensure that the first level obstacles were not interfering on the second level map.

The second pair of obstacles were a little more complex, firstly because the car must take an unusual action, as mentioned, when there is an obstacle the desired behavior is to stop, not to overtake it. Secondly, the obstacles were placed really close to the car's path, in order to force a deviation during the overtake. The car's objective is to drive as close as possible to the obstacle to avoid changing lanes. Depending on the car's current speed, it may consider that there is not enough time to deviate safely, causing a break event, but after the car slows down it finds a suitable path to follow. This is the cause why the car slows down in some moments, as seen in the orange obstacles.

Finally, the last pair of obstacles were placed to guarantee that the level1 map is working properly. Forcing an unnatural behavior, i.e. not previously designed for the system, was again an issue. The car properly overtakes the obstacles but could not maintain the safety distance, although never closer than the danger distance.

The main conclusions about the experiments are: the obstacles are being handled according to the map referent to its height; and, also according to the collision model of each height. There were no crashes in any of the seven laps, in a total of forty two obstacles were successfully overtaken. In addition, it is worth mentioning that overtaking maneuvers are not expected with the airplane. For a better visualization, the experiment video is available at https://youtu.be/g_VrjMadjmQ.

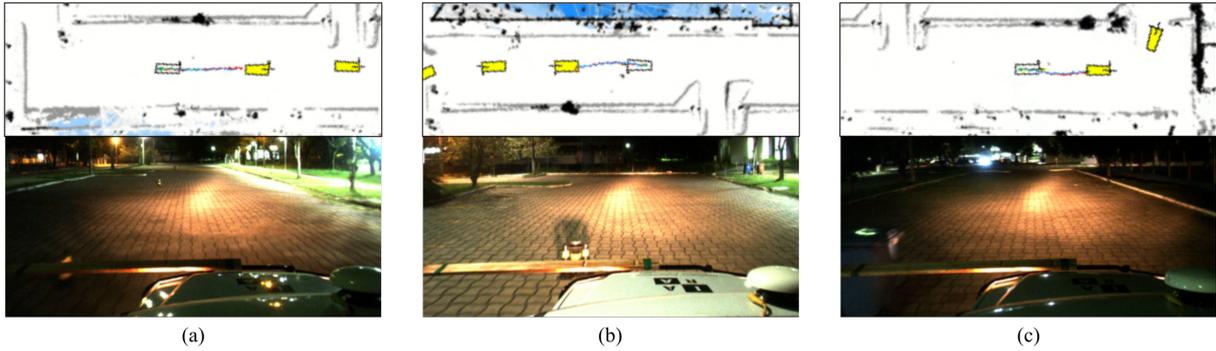


Figure 12 – Simulated wing results. On each column, the upper image is a representation of the car in the map, the map components were described in the experiments section, the vehicle is represented by a white polygon, in this case a rectangle (the car) with an extended side (simulated wing). The next yellow polygons are the planned future poses and the blue and red lines are plans. The lower image is the picture taken by the car’s front camera. Both images, upper and lower, represent the same moment in time, in two different representations.

5.2 Airplane

As mentioned before, tests on a real aircraft scenario involve a lot of effort, costs and risks. In particular, an integration test campaign was performed using the aircraft rig, a.k.a. iron bird, comprising all aircraft software with real hardware-in-the-loop, to check the autonomous system interfaces and validate its overall behaviour. Moreover, during the taxiing tests a experienced pilot should be available to take control under dangerous circumstances, and the company has to assign one of their testing aircrafts for the project. Besides that, international test protocols must be followed. Therefore, the tests were not designed exclusively for the Multi-Height Occupancy grid map but for an overall test of the system. More specifically, the performance of the proposed method was assessed during several systematic tests performed to demonstrate the full integration of an autonomous taxiing system using a real aircraft, the Praetor 600 jet. Fig. 13 depicts the remission and multi-height occupancy grid map made at the airport side by side. A sample of the full experiment is also available in a promotional video produced by Embraer S.A. and available at <https://www.youtube.com/watch?v=jDCvRSJeUN8>.

After the proper calibration of the parameters of the experimental control law¹, the aircraft were able to complete the entire four miles closed circuit autonomously twice, without any pilot interventions. The overall experiment was considered a success, since it demonstrated the feasibility of a novel application of these techniques on the global

¹ Since Praetor 600 is a full fly-by-wire aircraft, the control law was safely designed to take partial (surmountable) control over the pilot trust levers and the copilot pedals. Specifically, the steering control loop was directly commanding the nose land gear steering actuators, whereas the speed control loop was commanding the engine thrust levers through the auto throttle and the mainland gears break pedals which was specifically designed for the taxiing proof of concept.

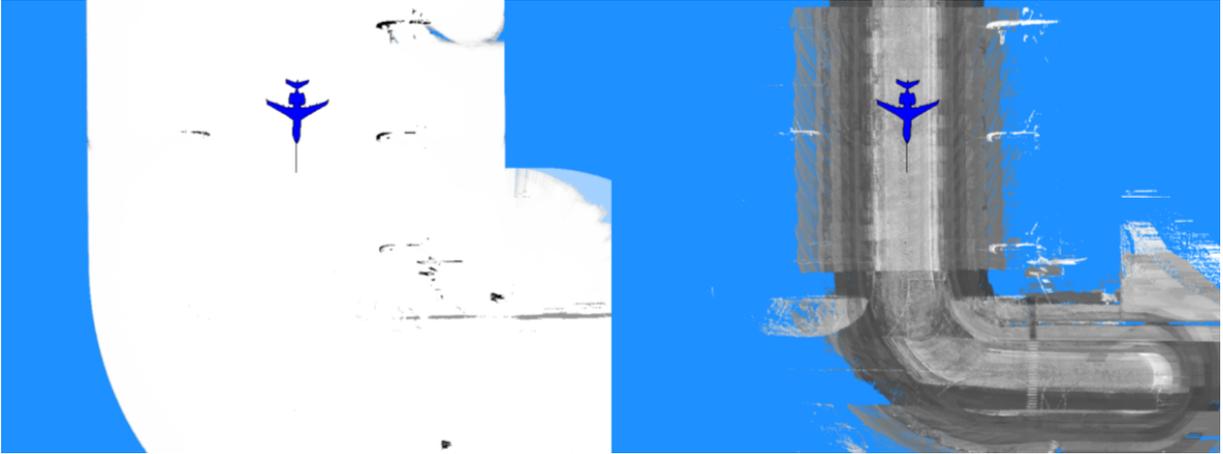


Figure 13 – Airport multi-height occupancy grid map (on the left), and remission map (on the right)

scenario. The functionalities related to this paper (i.e., Multi-Height Occupancy grid map) worked perfectly during the three experiments on the airplane, indicating to be a promising technique to be used in related applications.

5.3 Memory Consumption and Performance Analysis

Comparing the proposed solution with some other methods in the literature, the benefits of this approach is evident.

The memory consumption of a 3D occupancy grid map increases with its length, width and height, and the same goes for the 3D ODMs. While the proposed method increases considering only the width, length and number of height levels, which should be significantly lower than the height discretization. The processing time to build the OGM should be really close in both, 3D or 2D scenarios, since they rely on the LiDAR data, both of them must process each point in the cloud individually. However, to build the ODM the 3D solution will have to be calculated considering length, width and height, while the proposed method only considers width and height, but for each height of interest.

The 2.5D maps are more similar to the proposed solution in performance and memory consumption. Assuming that both maps have the same width and height, the only difference in memory will depend on the data structure that stores the information on each cell, resulting in small differences that can be relieved. As mentioned in chapter 3.2.2 the ODM maps created will be the same, this means that they will have similar performance to build. Therefore, The biggest difference between the two will be in the map creation performance. However, more complex navigation and localization systems will be required to handle 2.5D maps.

6 Conclusions

The system of multi-height grid maps presented by this work was designed to make the IARA system capable of considering complex vehicles geometries that include collisions in different heights, instead of considering the vehicle always as a rectangle.

To accomplish this task, first a new collision representation of the vehicle was proposed. Based on circular collisions and with indication of the height for each circle, the novel collision model is capable of representing almost any vehicle shape with discrete levels of different heights of interest, taking advantage of the current map structure.

Subsequently, the map system generates additional maps according to the heights of interest. The proposed mapping system does not fully represents the surrounding environment in three dimensions, as many other mapping systems already on the literature [9, 2], however, the proposed method takes advantage of the maps and other additional modules already present at the IARA, preserving the simplicity and other improvements developed along the past years.

Finally, the modules that handle the collisions need to be adjusted to consider the different collision heights in the model and the appropriate maps.

In summary, the proposed system accomplished the main goal of dealing with the collisions from different heights of interest, more specifically the fuselage and wings of an aircraft, by creating multiple grid maps according to the heights of interest. A full 3D representation of the environment can be used to solve such a problem, however, many other modules of the IARA system must be completely reworked to adapt to such changes, adding unnecessary complexity and more computational load to the system.

Even though the system fits its purpose, there is always room for improvements and additional studies, for instance:

- The intuitive perception that a 3D model will be more complex than 2D is shown in many other works, but a deeper study of its impact on this specific situation would show a quantitative comparison between the proposed method and the present in the literature.
- The decision making modules that needed to consider multi-height obstacles were not the main focus of this work, they were very basically adapted in order to work properly.

Bibliography

- 1 CHAO, H.; CAO, Y.; CHEN, Y. Autopilots for small unmanned aerial vehicles: a survey. *International Journal of Control, Automation and Systems*, Springer, v. 8, n. 1, p. 36–44, 2010. Cited on page 12.
- 2 DRYANOVSKI, I.; MORRIS, W.; XIAO, J. Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles. In: IEEE. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2010. p. 1553–1559. Cited 4 times on pages 12, 14, 22, and 39.
- 3 FRAUNDORFER, F. et al. Vision-based autonomous mapping and exploration using a quadrotor mav. In: IEEE. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2012. p. 4557–4564. Cited 3 times on pages 12, 14, and 22.
- 4 GUTMANN, J.-S.; FUKUCHI, M.; FUJITA, M. A floor and obstacle height map for 3d navigation of a humanoid robot. In: IEEE. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. [S.l.], 2005. p. 1066–1071. Cited 3 times on pages 12, 14, and 22.
- 5 KWAK, N. et al. 3d grid and particle based slam for a humanoid robot. In: IEEE. *2009 9th IEEE-RAS International Conference on Humanoid Robots*. [S.l.], 2009. p. 62–67. Cited 2 times on pages 12 and 14.
- 6 BADUE, C. et al. Self-driving cars: A survey. *arXiv preprint arXiv:1901.04407*, 2019. Cited 6 times on pages 13, 14, 19, 20, 25, and 31.
- 7 ROTH-TABAK, Y.; JAIN, R. Building an environment model using depth information. *Computer, IEEE*, v. 22, n. 6, p. 85–90, 1989. Cited 2 times on pages 13 and 22.
- 8 MILLER, I.; CAMPBELL, M. A mixture-model based algorithm for real-time terrain estimation. *Journal of Field Robotics*, Wiley Online Library, v. 23, n. 9, p. 755–775, 2006. Cited 2 times on pages 13 and 22.
- 9 HORNUNG, A. et al. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, Springer, v. 34, n. 3, p. 189–206, 2013. Cited 3 times on pages 13, 23, and 39.
- 10 MUTZ, F. et al. Large-scale mapping in complex field scenarios using an autonomous car. *Expert Systems with Applications*, Elsevier, v. 46, p. 439–462, 2016. Cited 4 times on pages 19, 20, 24, and 28.
- 11 MORAVEC, H.; ELFES, A. High resolution maps from wide angle sonar. In: IEEE. *Proceedings. 1985 IEEE international conference on robotics and automation*. [S.l.], 1985. v. 2, p. 116–121. Cited on page 20.
- 12 RIVADENEYRA, C. et al. Probabilistic estimation of multi-level terrain maps. In: IEEE. *2009 IEEE International Conference on Robotics and Automation*. [S.l.], 2009. p. 1643–1648. Cited on page 22.

- 13 THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic robotics*. [S.l.]: MIT press, 2005. Cited 2 times on pages [24](#) and [25](#).
- 14 VERONESE, L. de P. et al. A light-weight yet accurate localization system for autonomous cars in large-scale and complex environments. In: IEEE. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.], 2016. p. 520–525. Cited on page [25](#).
- 15 BERGER, M. et al. Traffic sign recognition with wisard and vg-ram weightless neural networks. *Journal of Network and Innovative Computing*, v. 1, n. 1, p. 87–98, 2013. Cited on page [25](#).
- 16 SOUZA, A. F. D. et al. Traffic sign detection with vg-ram weightless neural networks. In: IEEE. *The 2013 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2013. p. 1–9. Cited on page [25](#).
- 17 BERRIEL, R. F. et al. Ego-lane analysis system (elas): Dataset and algorithms. *Image and Vision Computing*, Elsevier, v. 68, p. 64–75, 2017. Cited on page [25](#).
- 18 CARDOSO, V. et al. A model-predictive motion planner for the iara autonomous car. In: IEEE. *2017 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2017. p. 225–230. Cited on page [26](#).
- 19 GUIDOLINI, R. et al. A simple yet effective obstacle avoider for the iara autonomous car. In: IEEE. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.], 2016. p. 1914–1919. Cited on page [26](#).
- 20 GUIDOLINI, R. et al. Neural-based model predictive control for tackling steering delays of autonomous cars. In: IEEE. *2017 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2017. p. 4324–4331. Cited on page [26](#).
- 21 VERONESE, L. P. de et al. Evaluating the limits of a lidar for an autonomous driving localization. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, 2020. Cited on page [35](#).