Tiago de Brito Novo

EXAMINER
3D Multi-Robot Exploration in Irregular Terrains

March 2021

• U C •

· U C ·

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

EXAMINER - 3D Multi-Robot Exploration in Irregular Terrains

Tiago de Brito Novo

Coimbra, March 2021

EXAMINER - 3D Multi-Robot Exploration in Irregular Terrains

Supervisor:

Prof. Doutor Rui Paulo Pinto da Rocha

Co-Supervisor:

Doutor David Bina Siassipour Portugal

Jury:

Prof. Doutor Rui Alexandre de Matos Araújo

Prof. Doutor Cristiano Premebida

Prof. Doutor Rui Paulo Pinto da Rocha

Coimbra, March 2021

The struggles along the way are meant
to shape you for your purpose

Chadwick Boseman

Acknowledgments

This work would not be possible without the support and companionship of some important people. In one way or another, everyone has been there and made their contribution to the success of this project.

I would like to thank my supervisor, Professor Rui Rocha who was always there to guide me in the right direction, for all the meetings we had, and the way he was always available to clarify all of my doubts, but most importantly, how he always pushed me forward when things were not going well, keeping my mind positive.

To my co-supervisor, Dr David Portugal, who since the beginning of this project showed me that I could count on him for any problem I had. Every time an issue came up, and there were a few that did, David was there to check the problem and to point the best way to solve it.

A special thanks to Dr. Luigi Freda, the mentor of the project, which without him, would not be possible. So many times we spent hours discussing approaches, and I have learned so much from him. From Luigi, I will take plenty of friendly advice for life.

To the Mobile Robotics Laboratory of the Institute of Systems and Robotics (MRL-ISR) of the University of Coimbra for the dedicated space and material provided for this work.

To Carolina Maciel who dedicate her time to review the English writing of this document.

For those who have welcomed me when I arrived in Coimbra. With them, I lived for almost three long years full of friendship and companionship.

At the University, I made a lot of friends without whom I can say that this road would not have had the same joy, and such a good road the one we had. What a great representation we left at the parade of “Queima das fitas” in 2018. “Alone we can do so little, together we can do so much”, never a sentence could correspond better to these many years.

To my scout friends with whom I grew up since my young age and with whom I spent most part of my weekends, I hope we stay together for many years to come.

To all my friends from Viana do Castelo who I have known for many years, I have to

apologize for the little time available in the past few months and all the weekends I have stayed in Coimbra. But yet, thank you all for all your companionship and friendship, after all, friends are the family we choose.

To a very special person, Alice Conceição, for the endless support and patience in the course of this dissertation. Her comprehension and motivation was definitely very important for the success of it.

Finally, to my family, that has always supported me and believed in the pursuit of my dreams. Thank you. All of this would not be possible without your support.

Resumo

Hoje em dia é possível habilitar um robô com sensores para este operar de forma autónoma numa dada tarefa para a qual é projetado, como por exemplo inspeção industrial [1], [2], monitorização de colheitas [3], [4], busca e salvamento [5], [6], etc. Um dos desafios da robótica móvel envolve a sua operação em ambientes desconhecidos, a capacidade de tomar decisões quando nenhum conhecimento prévio é assumido sobre o ambiente, sendo capaz de realizar de forma autónoma tarefas como exploração, reconhecimento e patrulhamento. O desafio é ainda maior ao realizar a exploração de ambientes desconhecidos com uma equipa de robôs, com os seus elementos cooperando entre si com o propósito de explorar todo o ambiente [7]. Uma vez superado esse desafio, estes sistemas multi-robô deverão ser capazes de realizar tarefas com mais facilidade e em segurança, em ambientes mais complexos e em tempos mais reduzidos quando comparados com um único robô.

Esta dissertação pretende dar um passo em frente neste desafio, apresentando uma nova técnica para estabelecer coordenação numa equipa de robôs, maximizando a eficiência na execução da tarefa e, concomitantemente, minimizando a sua duração e os possíveis conflitos ao estender a exploração para um sistema com múltiplos robôs.

Além disso, uma das vantagens deste projeto é apresentar uma abordagem que permite aos robôs móveis percecionar e agir em ambientes mais complexos. Não apenas reconhecer e mapear um ambiente em 3D, mas enquanto o fazem, usar esse mapa para a consequente navegação, mesmo que o terreno seja irregular.

O método apresentado neste documento foi testado, aprimorado e validado com testes experimentais em simulação e em ambientes reais. Com os resultados obtidos, esta técnica mostra ser eficaz e eficiente para assegurar a cooperação numa equipa de robôs em missões de exploração independentemente do terreno em que estes operam.

Palavras-Chave: *Exploração 3D, Cooperação, Sistemas Multi-Robô, Terrenos irregulares, ROS.*

Abstract

Nowadays, we can equip a robot with on-board sensors and it has the potential to operate autonomously for any task it is designed to, such as industrial inspection [1], [2], crop monitoring [3], [4], search and rescue [5], [6], etc. One of the challenges of mobile robotics involves operating in unknown environments. This assumes the ability of robots to make decisions when no prior knowledge is assumed about the environment, being able to autonomously perform tasks like exploration, coverage and patrolling. The challenge is even bigger when performing exploration of unknown environments with a team of robots, with its elements cooperating with each other for the same purpose [7]. By overcoming this challenge, the Multi-Robot System (MRS) is capable of performing tasks with better performance, *e.g.* in less time, in comparison with a single-robot solution.

This dissertation intends to take a step forward towards this challenge, presenting a new MRS exploration method to be used in a team of robots, whereby completeness and time of exploration are optimized and possible conflicts between robots are either avoided or readily overcome.

Also, an important feature of the proposed MRS exploration method is dealing with the problem in complex environments, including performing 3D mapping and navigating in irregular terrains.

The method presented in this dissertation was tested, improved and validated with experimental tests in simulation and real environments. With the presented results, this technique has proven to be effective and efficient to ensure cooperation of a team of robots in exploration missions.

Keywords: *3D Exploration, Cooperation, Multi-Robot Systems, Irregular terrains, ROS.*

List of Acronyms

CPU Central Processing Unit

FoV Field of View

GPU Graphics Processing Unit

GUI Graphical User Interface

IMU Inertial Measurement Unit

IP Internet Protocol

IR Infrared

ISR Institute of Systems and Robotics

ISRsea Institute of Systems and Robotics shared experimental area

LiDAR Light Detection And Ranging

MRL-ISR Mobile Robotics Laboratory of the Institute of Systems and Robotics

MRS Multi-Robot System

NBV Next Best View

PGO Pose Graph Optimization

RGB Red Green Blue

RGB-D Red Green Blue and Depth

RH-NBV Receding Horizon - Next Best View

ROS Robot Operating System

RRT Rapidly-Exploring Random Trees

RTAB-Map Real-Time Appearance-Based Mapping

Rviz ROS visualization

SL Structured Light

SLAM Simultaneous Localization and Mapping

TRADR Long-Term Human-Robot Teaming for Disaster Response

UGV Unmanned Ground Vehicle

UAV Unmanned Aerial Vehicle

Contents

Acknowledgments **iii**

Resumo **v**

Abstract **vii**

List of Acronyms **ix**

Contents **x**

List of Figures **xiii**

List of Tables **xvii**

1 Introduction **1**

1.1 Motivation and Challenges 2

1.2 Objectives 3

1.3 Contributions 4

1.4 Document Overview 5

2 Background and Related Work **7**

2.1 Exploration Strategies 7

2.2 Multi-Robot Cooperation Strategies 11

2.3 Multi-Robot Exploration Strategies Overview 16

3 Proposed method for 3D mapping and exploration **19**

3.1 The Two-Level Coordination Strategy 19

3.2 Spatio-Temporal Filter 25

3.2.1 The algorithm 26

3.3 Sensing system 27

3.4 ROS - Robot Operating System . . . 29
3.4.1 ROS Packages . . . 30
3.4.2 V-REP Simulator . . . 31
3.5 System Outline . . . 31

4 Ablation Study 33
4.1 Experimental Environment . . . 33
4.1.1 Hardware . . . 33
4.1.2 Software . . . 34
4.2 Simulation tests preparation . . . 35
4.3 Simulated scenes . . . 36
4.4 Multi-Robot Exploration Simulations . . . 38
4.4.1 Simulation Results . . . 39
4.5 Summary . . . 44

5 Experimental Validation and Results Analysis 47
5.1 Setup of Experiments with Real Robots . . . 47
5.1.1 Odometry calibration . . . 50
5.1.2 Visual Odometry vs Wheel odometry . . . 52
5.1.3 Mynt Eye S1030 vs Intel Realsense D435i . . . 54
5.2 Experimental Tests . . . 55
5.2.1 Single-Robot Experiments . . . 56
5.2.2 Multi-Robot Experiments . . . 59
5.2.3 Experiments in a More Complex 3D Environment . . . 64
5.3 Summary . . . 70

6 Conclusion and Future Work 73
6.1 Overview . . . 73
6.2 Future Work . . . 75

7 Bibliography 77

8 Appendix 83
8.1 Detailed Experimental Results . . . 83
8.1.1 ISRsea scenario . . . 83
8.1.2 Auditorium scenario . . . 84

List of Figures

1.1 Distinct 3D map representations commonly used in the literature. 2

1.2 Robotic systems. Both snapshots extracted from the V-REP simulator. . . . 5

2.1 Representation of frontiers in a 2D environment. Black markers correspond to occupied areas, in grey is the free area and the frontiers are illustrated in blue. 8

2.2 Voronoi diagram of 10 regions with their points. Space is partitioned into 10 voronoi cells (A_i), each of which is constituted by all the points which are closer to one site than to any other site. The edge between two areas consists of a subset of equidistant points to two sites. Some cells, such as A10 and A4, extend to infinite [8]. 9

2.3 Expanded branches from the root node for selection of the next best view [9]. Expanded branches are represented in red, the green node represents the root node, the green line represents an alternative exploration path computed by the frontier-based approach, blue and black lines represent the whole path of the mission. 10

2.4 Representation of the cached nodes in the map using the potential information gain [10]. 11

2.5 Broadcast of the local grid to all robots, which integrate it in their global grids [11]. 12

2.6 Possible target assignment for two different robot positions. The black point indicates the target with minimum cost for each robot [12]. 13

2.7 Assignment of target (black point) following a coordination approach. The black point indicates the target with minimum cost for each robot [12]. . . . 13

2.8 Local minima (right frame) and path followed by the robot to return the exploration mission (left frame) [13]. 15

2.9 Sequence of images representing the coordination strategy proposed by Colares *et al.* [14]. 15

3.1 Representation of the exploration model on each robot. 20

3.2 Segmentation of the map representation, M_j. 21

3.3 Volumetric map, V_j, represented in the form of an octomap. 21

3.4 Search tree S_j locally expanded around the current view configuration. The node marked in yellow represents the candidate view node and the node marked in green represents the next best node in the exploration tree. 22

3.5 Illustrations of the two possible conflicts when exploring during mission. . . . 24

3.6 Two-level coordination strategy workflow for each robot. 24

3.7 Depth cameras used. 28

3.8 Simple network setup for the single computer Robot Operating System (ROS) network case to exemplify the network configuration. 30

4.1 Pioneer 3-DX with onboard Red Green Blue and Depth (RGB-D) camera simulated in V-REP. 34

4.2 Scenes containing two robots. 37

4.3 *rooms3 3D* scene - scene with 3 robots. 37

4.4 Simulated environments with irregular terrain and ramps containing four robots. 37

4.5 Amount of occupied cells integrated into the local map of a robot before and after applying the spatio-temporal filter in the *rooms2* scene. 41

4.6 Scanned and received maps of each of the two robots in the *extreme* scene. Results with [2; 10; 0.7] set. 42

4.7 Scanned and received maps of each of the three robots in the *rooms3 3D* scene. Results with [3; 10; 0.35] set. 42

4.8 Scanned and received maps of each of the four robots in the *ramps* scene. Results with [4; 10; 0.35] set. 42

4.9 Trajectory path performed by each robot in the *ramps_2* scene (2D and 3D perspective). Result with [4; 10; 0.35] set. 43

4.10 Trajectory path performed by each robot in the *extreme* scene. Result with [2; 10; 0.7] set. 44

4.11 Trajectory path performed by each robot and their travelled distance in *rooms3 3D* scene. Result with [3; 10; 0.35] set. 44

5.1 *Institute of Systems and Robotics shared experimental area (ISRsea)* scenario. 48

5.2 *A3 Auditorium* scenario. 48

5.3 The Pioneer 3-DX mobile robot platform. 49

5.4 3D Maps of the ISRsea environment. 50

5.5 Correcting odometry using the UMBMark method [15]. 51

5.6 Synchronization example of a stereo camera and odometry. In this case, odometry is computed through wheel encoders. Camera messages are synchronized together using rgbd_sync ROS node before synchronizing the resulting RGB-D image message with the other sensors (which can have different publishing rates). On the right is an example of the resulting TF tree for this sensor configuration, with transforms linked by a dotted line to corresponding publishing ROS nodes. Reproduced from [16]. 53

5.7 Examples of path performed by robots using combined visual and wheel odometry. 53

5.8 Illustrative point clouds of Realsense and Mynt Eye. 54

5.9 Workflow of exploration mission on each robot. 55

5.10 Bar graph with each test on each machine. 57

5.11 Path followed and distance travelled by the robot during the exploration mission. 57

5.12 Segmented point cloud during the exploration mission. Obstacles in red and traversable area in green. 58

5.13 Occupied cells of the volumetric map during the exploration mission in the form of octomaps. 58

5.14 Speedup graph of t_{expl}, t_{95} and t_{end} with 1, 2 and 3 robots in the ISRsea scenario. 60

5.15 Multi-robot exploration with 3 robots. 61

5.16 Sequence of the local map of robot 1. the blue marker represents robot 1. . . 61

5.17 Sequence of the local map of robot 2. The blue marker represents robot 2. . 61

5.18 Sequence of the local map of robot 3. The blue marker represents robot 3. . 62

5.19 Sequence of the local map of robot 1. The blue marker represents robot 1. . 62

5.20 Sequence of the local map of robot 2. The blue marker represents robot 2. . 62

5.21 Number of occ_{expl} of each robot during exploration missions and $t_{expl_{95}}$ of the volumetric map. 63

5.22 Illustration of management of conflicts. Monochromatic point cloud represents the map previously mapped; points in green represent the traversable region; red points represent the obstacles; nodes are expanded from the roots of each robot, in which the best node of the tree and the next planned node are presented in yellow and green, respectively; The blue line illustrates the path that robot will follow to reach the selected node. 64

5.23 Illustration of original (a) and decimated point cloud (b). 65

5.24 Illustrations of the *A3 Auditorium* scenario. 66

5.25 Multi-robot exploration with 2 robots in the *A3 Auditorium* scenario. 66

5.26 Number of occ_{expl} of each robot during exploration missions and $t_{expl_{95}}$ of the local map. 67

5.27 Multi-robot exploration with 3 robots in the *A3 Auditorium* scenario. 67

5.28 Number of occ_{expl} of each robot during exploration missions and $t_{expl_{95}}$ of the volumetric map in the *A3 Auditorium*. 68

5.29 Sequence of the local map of robot 1. The blue marker represents robot 1. . 68

5.30 Sequence of the local map of robot 2. The blue marker represents robot 2. . 69

5.31 Sequence of the local map of robot 3. The blue marker represents robot 3. . 69

5.32 Side view of a ramp of the *A3 Auditorium* with traversable region (in green), in which exploration tree nodes are expanded. Point cloud in shades of gray represents the provided map to be used for the localization. 70

6.1 Robot using an extra sensing source in order to perceive closer objects that are not detected in the Field of View (FoV). 76

List of Tables

2.1 Comparative table between presented multi-robot cooperative strategies. . . 16

3.1 Relevant features of the depth cameras analysed. 29

4.1 Combination of parameters tested in the initial simulation stage. 35

4.2 Detailed specifications of the simulation scenarios proposed. 38

4.3 Overview of the first tests with the application of the spatio-temporal filter. . 39

4.4 Overview of the second tests with the application of the spatio-temporal filter. 40

5.1 Comparative results when using visual and visual + wheel odometry. 52

5.2 Specifications of each machine used in exploration tests. 57

5.3 Table with overall time results performed on each machine. 57

5.4 Summary of the performed tests with the results with superior performance highlighted. Speedup values shown at the right-side of the table. 59

5.5 Summary of the performed tests - results with superior performance highlighted. 69

8.1 Results of exploration tests using one robot with different machines. 83

8.2 Results of multi-robot exploration tests using 2 robots. 83

8.3 Results of multi-robot exploration tests using 3 robots. 84

8.4 Results of multi-robot exploration tests using 2 robots. 84

8.5 Results of multi-robot exploration tests using 3 robots. 84

1 Introduction

This dissertation aims to study, develop and validate a new decentralized 3D multi-robot exploration technique using a team of robots to perform exploration in irregular terrains. This new exploration technique relies on two planners, assigning the position and trajectory of the robot taking into account the assignments of the teammates. This intends to improve the capacity of the robots to perform effectively an exploration mission, aligned with proper coordination among team members.

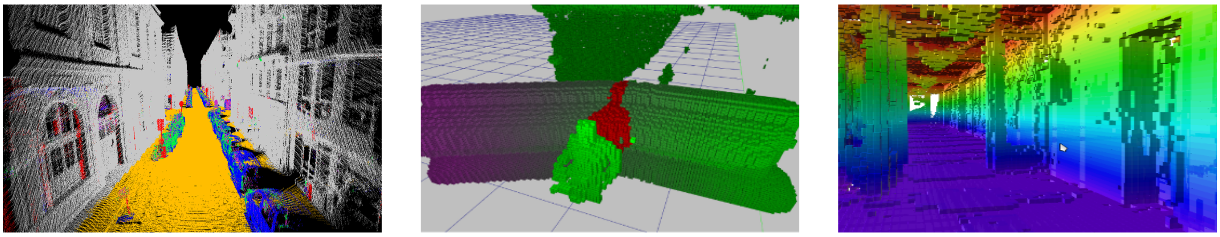
This project has taken place in the Mobile Robotic Laboratory of Institute of Systems and Robotics MRL-ISR from the University of Coimbra and has benefited from the contribution of Dr. L. Freda, a former researcher of La Sapienza University of Rome who was involved in the Long-Term Human-Robot Teaming for Disaster Response (TRADR) project, a successful European research project finished in 2018, in the scope of the 7th Framework Programme (FP7).

Robotic exploration can bring many benefits to society in tasks carried out in places that are hard to reach by man, or where it is more practical to send a robot to perform the task, *e.g.* exploring other planets, the deep sea or underground mines.

Yamauchi [17], defined *exploration* as the “act of moving through an unknown environment while building a map that can be used for subsequent navigation”. This is particularly important to perform a first analysis about that environment, perceiving the surroundings for possible danger, obstacles, passage conditions, etc.

The purpose of the 3D exploration is to acquire a fully detailed 3D map of a zone previously unknown in an efficient way. 3D environments can be represented through different types of data structures, such as point clouds, voxels, octomaps, etc. (see Figure 1.1).

Exploration missions benefit if, instead of using a single robot approach, they make use of a Multi-Robot System (MRS). Having multiple robots exploring an unknown environment will increase the potential to accomplish the task faster than with a single robot [21], increasing their efficiency and reducing total mission time.



(a) Point Cloud [18]. (b) Voxels [19]. (c) Octomap [20].

Figure 1.1: Distinct 3D map representations commonly used in the literature.

Furthermore, using more robots introduces redundancy, so the system is expected to have greater fault-tolerance over a single robot [12]. Another advantage of using a team of robots is the merging of overlapping information, helping to compensate for some sensor uncertainty. This is especially important when using robots with different sensor capabilities [22].

On the other hand, a reasonable number of robots should be employed in the environment, in order to minimize interference among members of the team and maximize the contribution of each one to the mission.

Whenever robots are operating as a team with the mission to explore an unknown environment, they should be capable of sharing information with their teammates. The information about their position and the local map is very important so that other robots realize which locations remain unexplored. For this, regular communication between them is important, otherwise, if robots lose communication, they will be exploring blindly, not knowing whether that area has already been explored by another teammate.

Robots should be capable of building maps online while moving in the field, performing Simultaneous Localization and Mapping (SLAM) [23]. While mapping the environment, a robot has to deal with noise in range measurements and possible errors in odometry, which may imposes some problems when determining the location of a robot relatively to its map. Therefore, sharing information may help the robot to re-localize.

1.1 Motivation and Challenges

Robotic systems have been constantly improving our daily life. One sensitive case of relevance is to use of robots in post-catastrophe scenarios in which the infrastructures may be very fragile and no human life should be risked. Robots are simply instruments that can be easily replaced and considered expandable in such extreme scenarios. In a practical scenario, they can perform the first recognition of the scene, detecting victims and even perceive the most appropriate way to act, thus facilitating the operators' work.

Performing multi-robot exploration in complex environments brings several challenges, such as robots maintaining communication with others in order to not lose information about the mission and the environment, or robots interfering with other teammates when exploring close regions. Also, there is the need to endow the robots to plan and navigate in irregular terrains, especially when using robots that were not designed for this purpose. Consequentially, this brings the constraint that they should be capable of permanently localizing themselves in that environment while mapping, thus performing SLAM in those complex environments.

1.2 Objectives

The main goal of this dissertation is to study and validate a new technique of 3D multi-robot exploration that enables the exploration of unknown and possibly irregular terrains with a team of robots. This technique aims to improve exploration efficiency, by having each robot exploring different regions while avoiding conflicts between teammates.

In order to validate the system, an ablation should be firstly carried out, evaluating the approach and its capabilities, and improving it as needed. This should been done in a simulated environment, where a controlled environment can be set and tests can be easily reproduced with different simulation configurations.

After the ablation study, real-world environment tests should be performed, in order to demonstrate the feasibility of deploying the approach with physical teams of robots. Performing these tests brings additional challenges that a simulator framework does not present, such as the measurement error of sensors, conditions and network connection that may be affected by the constraints of the environment, possible hardware failures, wheel slippage, etc.

Since this dissertation's work focuses on the exploration of unknown regions, multi-robot coordination and conflict avoidance, and not the SLAM approach itself, the localization of the robots is previously induced so that robots know exactly where they are while exploring the scenario. This should be performed with the restriction that robots have no prior knowledge of the scene, only their poses in a common frame of reference.

To the success of the proposed approach, robots should be able to explore, without prior information, in the unknown environment, share mission information within the team and finish the exploration mission when no new information is possible to acquire. The final result should be a 3D map with the information collected by all the robots, which individually contribute to this map in the course of the exploration mission. Also, the exploration mission time is expected to decrease as many robots are added to the team.

The developed approach should be agnostic, *i.e.* able to work regardless of the sensor type, *e.g.* a Red Green Blue and Depth (RGB-D) camera or a 3D laser range finder. As for the robot type, this should be chosen by taking into account the scenario where they should operate. Some robots are not designed to maneuver in irregular terrains. In fact, the robots used in this work are not designed for that purpose, so well reasoned types of scenarios should to be considered.

1.3 Contributions

The main contributions of this dissertation are described in this section.

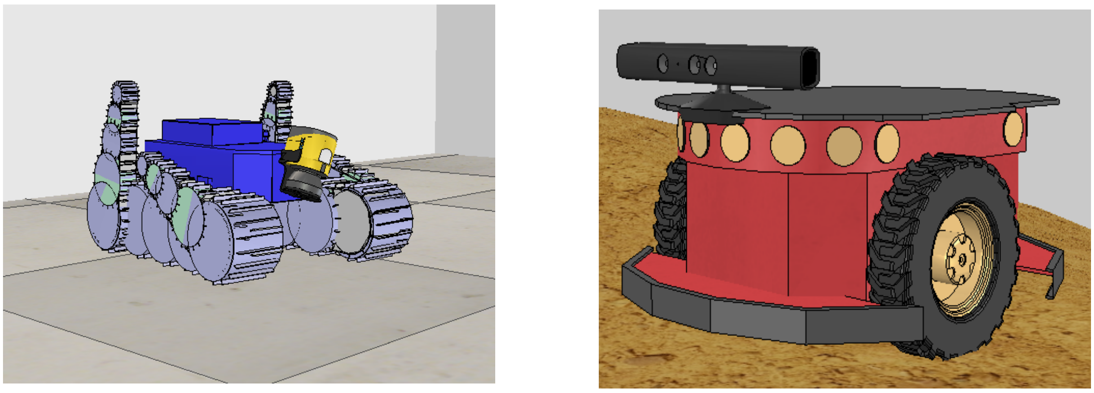
At the beginning of this work, the initially designed system by Dr. L. Freda was already capable of performing 3D multi-robot exploration. Yet, this was only possible in simulation, with a specific robot and a specific set of sensors (Figure 1.2a). After a deep study of the existing system, we have extended the system to make use of a simulated version of the hardware available in the MRL-ISR (Figure 1.2b). This specific hardware used in both simulated and real environments is described in Chapter 3 and 5.

With the need to reduce the computation load on each robot, and to efficiently share the essential information to other robots, we have developed a spatio-temporal filter described in section 3.2 which also reduces network bandwidth needed for each robot.

To perform real-world tests, a commercial depth camera was integrated as the main sensory system of the robot, and thus underpins as a contribution since the original perception system was based instead on the acquisition of point clouds extracted from a rotating 2D laser scanner (see Figure 1.2a).

Although SLAM is not the main goal of this dissertation, it is crucial that robots keep their correct location along with the mission. For that reason, combined visual and wheeled odometry was used to perceive the motions of the robot, serving as input to the localization system that we deployed in the environment, allowing for a permanently precise localization in the tested environments.

Extensive experimental validation of the proposed multi-robot exploration approach has been conducted in both simulation and real world, allowing for a deep analysis, evaluation and discussion of the work. This is, as far as we are concerned, the most important con-



(a) Initially available robotic simulated system, consisting of the TRADR tracked Unmanned Ground Vehicle (UGV) equipped with a 2D rotating laser-scanner.

(b) Modified robotic simulation system composed of a Pioneer-3DX differential robot equipped with a depth camera.

Figure 1.2: Robotic systems. Both snapshots extracted from the V-REP simulator.

tribution of this dissertation, being reported in Chapters 4 and 5. From these real-world experimental tests, extensive datasets have been recorded allowing for offline analysis.

Finally, the approach will be available as open-source to the robotics community, enabling reproducibility, improvements and comparison against other multi-robot exploration strategies.

1.4 Document Overview

This dissertation is organized as follows: Firstly, Chapter 2 presents a discussion of the most important related works with this project, as well as relevant approaches, a comparison between them and, at the end, a presentation of requirements for the approach developed. Next, in Chapter 3, the proposed system is presented, clarifying what has been developed, and introducing the main tools used. In Chapter 4, an ablation simulation study is presented to test different system parameters and initially analyse and discuss system performance under different configurations. In Chapter 5, real-world tests are presented and the acquired results are discussed in detail. Finally, in Chapter 6, we make an overview of the developed work and present possible future lines of work.

2 Background and Related Work

2.1 Exploration Strategies

The main objective of an exploration mission is to build a precise map of the environment without previous knowledge of it and do it in minimal time. In order to achieve that, the robot should know how to plan and act throughout the mission, considering the data gathered from the environment.

Many approaches are dedicated to perform exploration missions such as frontier-based approaches [17] [11] [24] [25], Voronoi-based partitioning [8], potential fields [26], etc. The frontier-based strategy is certainly the most common approach to be used and from which several more were developed.

To improve the efficiency of exploration missions, coordination strategies with teams of robots were developed. These strategies intend to advertise information about the environment to all robots of the team. With each robot exploring different regions of the same environment, the time to explore will be reduced as more robots are added to the team.

In this chapter, the most relevant exploration strategies are reviewed, considering their strengths and weaknesses. We first focus the concept of exploration, and then cooperation strategies where explorations approaches are extended to be used in MRS.

Frontier-based strategy

In 1997, Yamauchi presented the Frontier-based exploration approach for robotic exploration [17], which can be used to obtain the newest information from unknown areas by moving the robot to the boundary between open space and uncharted territory, denoted as *frontier* (Figure 2.1). This early approach chooses the frontier where it can obtain the newest information, *i.e.* the region with the most unexplored area.

This approach was firstly developed to be used in 2D environments, but soon it was extended to 3D workspaces [24], so the idea of Frontier-based would now combine the frontiers

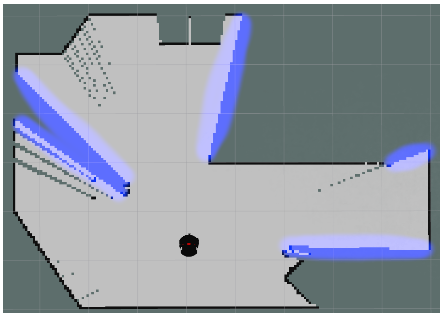


Figure 2.1: Representation of frontiers in a 2D environment. Black markers correspond to occupied areas, in grey is the free area and the frontiers are illustrated in blue.

in 3D with the concept of *voids*. Voids are unexplored volumes automatically generated from 3D sensor observations that are occluded or enclosed by obstacles.

For an adequate sensor viewpoint for observations of the interior of those voids, the extracted voids are combined with nearby frontiers. By intersecting all the vectors, it is possible to determine a location from where many of the void spaces can be perceived, in which for a larger scenario could be more than one location. Once the goal locations are extracted, it is time to evaluate their cost and utility based on the distance that robot has to those goal locations and how much new information can be acquired. The robot then moves to the locations sequentially until there are no more frontiers to explore.

Voronoi-based Partitioning Strategy

The Voronoi-based partitioning[1] strategy [8] is presented as an appropriate solution to perform exploration in large environments, reducing significantly the computational time. This technique relies on dividing the environment into polygons (Figure 2.2). In the beginning, the map is considered as a single unknown polygon. There are as many polygons as there are robots and so, each polygon is assigned to a single robot.

The system starts to assign a point (x,y location) to each robot, they start scanning and moving to those locations. This point will define the region of the polygon to explore for that robot. As the exploration mission goes, each polygon is marked as free, occupied or unknown. After each robot reaches the goal position, a new frontier is assigned to them. A frontier, in this case, is defined where an edge of an unknown polygon is adjacent to a polygon marked as free. Each robot should then move there considering that the frontier assigned is located in their region.

[1] Voronoi diagram is a partition of a plane into regions close to each of a given set of objects.

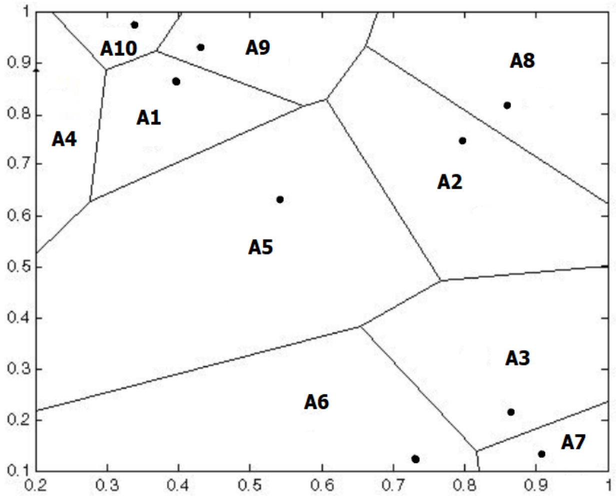


Figure 2.2: Voronoi diagram of 10 regions with their points. Space is partitioned into 10 voronoi cells (A_i), each of which is constituted by all the points which are closer to one site than to any other site. The edge between two areas consists of a subset of equidistant points to two sites. Some cells, such as A10 and A4, extend to infinite [8].

Next-Best-View strategy

The Next Best View (NBV) strategy implemented in exploration systems focus on the problem of a viewpoint selection and it was developed with the mission of 3D objects reconstruction taking different viewpoints [27].

In the context of mobile robotics, the NBV planner consists in determining the most favourable viewpoint of sensing action to be performed by the robot, moving the robotic platform to the desired location and acquire data with the purpose of obtaining as much of new information about the environment as possible [28].

Receding Horizon - Next-Best-View strategy

An alternative strategy employs a sampling-based receding horizon path planning [9], which generates a path that covers a significant volume of information yet unexplored.

A sensing system capable of acquiring information from the surroundings of the robot is used to combine that information, along with the localization of the robot, and then create an octomap[2] [20] of the already explored area. This map takes into account collision-free navigation and determination of the exploration progress in order to compute the most suitable path to go through. To compute it, a finite iteration of a random tree is expanded

[2] **Octomaps:** Representation of 3D environments, based on octrees using a probabilistic occupancy estimation.

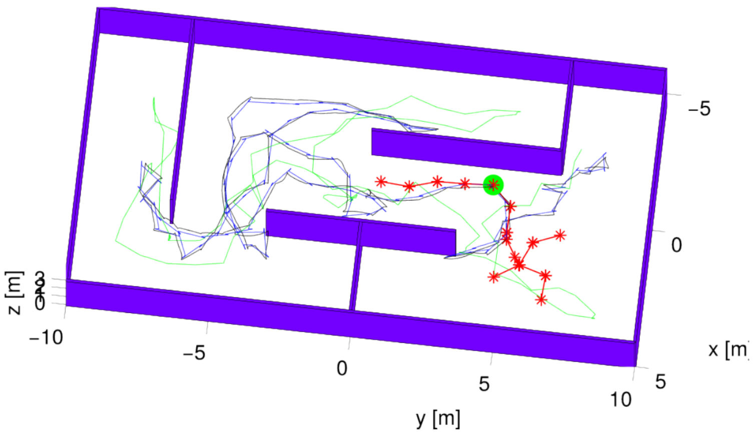


Figure 2.3: Expanded branches from the root node for selection of the next best view [9]. Expanded branches are represented in red, the green node represents the root node, the green line represents an alternative exploration path computed by the frontier-based approach, blue and black lines represent the whole path of the mission.

using the Rapidly-Exploring Random Trees (RRT) algorithm [29], [30], where each branch is evaluated for the amount of unmapped space that can be mapped from it. The first node of the best branch (next best node) is executed, while the whole process is then repeated.

With this approach, executing only the first node of the RRT finite branch is enough to explore the area until the robot reaches the frontier (Figure 2.3). While the robot is moving there, it is simultaneously mapping the area around it. Therefore, when it arrives at the frontier, the area ahead of it may already be mapped and there might not be necessary to explore further ahead. Another frontier is calculated, redirecting the robot to another unexplored area. Summing up, no long paths are needed to explore the entire environment.

Frontier-based with Receding Horizon

Recently, Magnus Selin *et al.* presented a new approach where two exploration strategies are combined [10], aiming at capitalizing the benefits of each one to improve the exploration mission. This approach relies on using the receding horizon strategy to perform local exploration planning and the frontier strategy to perform global exploration planning.

When the robot explores the area in its proximity, the exploration mission may terminate prematurely if the scores of the nodes of the RRT used are low, even if the most appropriate frontier is too far away from it to perceive it. In order to solve this problem, the authors decided to cache all the nodes with high potential information gain from the previous RRT. Therefore, when no node is locally found as a target to explore, these nodes are taken into account and the robot should move towards it.

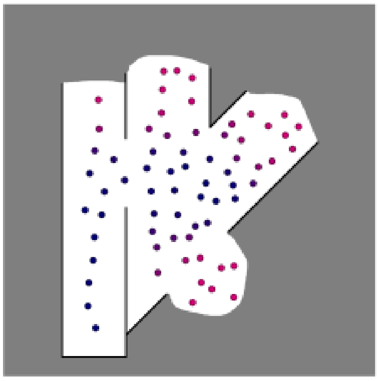


Figure 2.4: Representation of the cached nodes in the map using the potential information gain [10].

In Figure 2.4, the nodes marked in pink are the ones with highest potential information gain and in dark blue are the ones with the lowest. The black, white and grey colors represent occupied, free and unknown space, respectively.

Figure 2.4 illustrates how cached nodes spread along with the explored environment. The nodes at the frontiers are brighter because they represent high potential information gain, while blue nodes represent only previous nodes that the robot has already visited. As such, when the robot does not find further nodes to check in a local area, it may choose nodes from other frontiers to visit.

2.2 Multi-Robot Cooperation Strategies

Extending exploration approaches to support MRS may increase the efficiency of the exploration mission. Having a team of coordinated robots, exploring different regions of the environment allows to perform the mission in less time than if it was performed with a single robot. Using more robots to explore an environment may even create redundancy in the mapping process, thus correcting the explored area if errors occur.

Frontier-based Exploration using MRS

After Yamauchi presented the frontier-based exploration [17], he applied that strategy to a team of robots, thus extending the concept of frontier-based exploration to MRS [11]. In order to coordinate robots using frontiers, Yamauchi uses the frontier concept as a region which contains multiple connected frontier cells, denoted as a frontier edge. This edge is, as the frontier concept, where the known free space meets the unknown space.

The applied approach assumes that each robot has its own global evidence grid, which

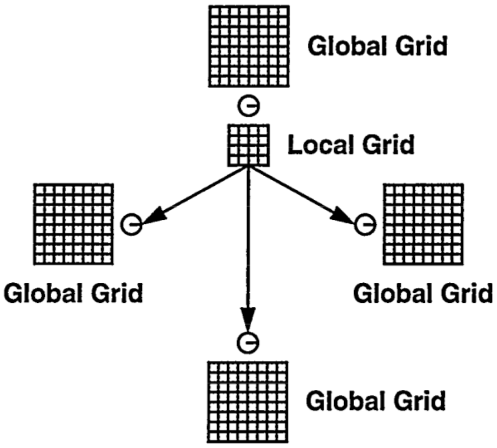


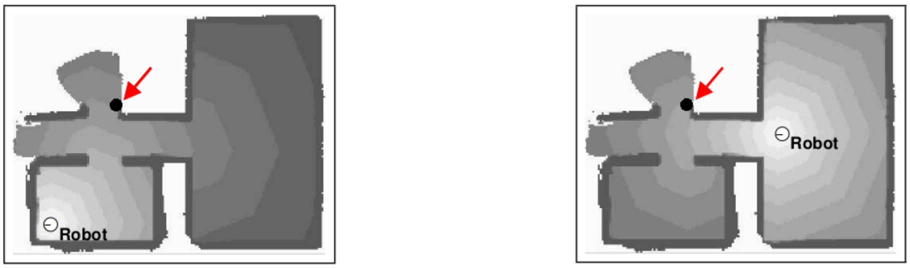
Figure 2.5: Broadcast of the local grid to all robots, which integrate it in their global grids [11].

represents its knowledge from the entire environment. When the robot acquires information from a frontier edge, the map will be built in its local occupancy grid, which represents its knowledge from that specific frontier. The local grid will then be integrated into the global grid and broadcasted to all other robots (see Figure 2.5), therefore, each one receives the new information about the environment.

This approach benefits from the fact of being decentralized, whereby each robot may have access to all the information from others; they can use that to determine where to navigate autonomously. This advantage may even be used when a robot discovers that a frontier detected from another robot is near to it and can decide to explore it. Since the navigation is independent, if one of the robots gets lost, out of communication or blocked, the mission can proceed without it.

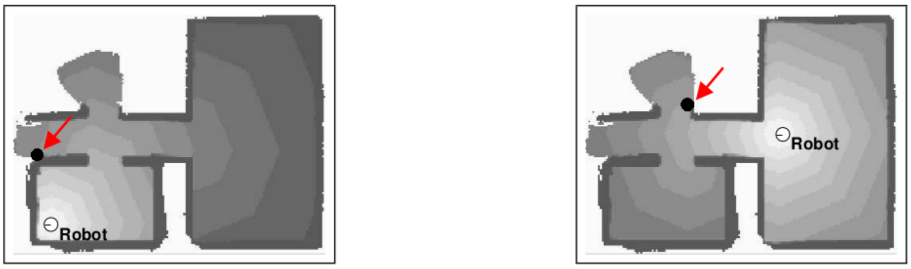
Decision-Theoretic Coordination Strategy

Burgard *et al.* presented a centralized system capable of coordinating a team of robots following a decision-theoretic approach [12] primarily presented by M. Moors in his PhD thesis [31]. This technique presents a solution to the problem of how to assign exploration tasks to individual robots of the same team, without having them moving to the same location (Figure 2.6), which determines target locations for individual robots. This determination simultaneously considers the cost of reaching a frontier cell and the utility of that cell. The cost of reaching a frontier cell is proportional to the distance from the robot to that cell and the utility of a frontier cell depends on the number of robots that are moving to that cell or close to it.



(a) (b)

Figure 2.6: Possible target assignment for two different robot positions. The black point indicates the target with minimum cost for each robot [12].



(a) (b)

Figure 2.7: Assignment of target (black point) following a coordination approach. The black point indicates the target with minimum cost for each robot [12].

To perform the multi-robot exploration and coordinate the robots' actions, each robot starts with a blank occupancy grid map. The system relies on the assumption that robots begin their operation in nearby locations, thus their scans show substantial overlap. Another assumption is that the system knows the approximate relative initial pose of the robots, allowing minor errors in orientation.

This technique was developed to be centralized, so the assignment of target points to all robots is done by the same controller. To prevent the assignment of the same areas to different robots, the algorithm assigns the most appropriate target to the robot with the most suitable evaluation and, the utility of that target is reduced so that the next assignment will not evaluate the same point. An example can be seen in Figure 2.7.

Finally, to achieve coordination, the team should be able to communicate and transmit individual maps during the mission. This is done by sending messages from a robot to all teammates within a correspondent cluster. To prevent the overlap of information, each robot stores a log of sensor measurements for each of the teammates perceived by this robot. The robot only transfers those measurements that have not been sent to a corresponding teammate. If a robot already has those measures, they are discarded, thus avoiding overlapping.

This approach is denoted as the most relevant pioneering work on multi-robot exploration, providing special attention to multi-robot coordination.

Multi-Robot exploration with topological recovery

Later, in 2008, R. Rocha *et al.* developed an approach based on grid maps and entropy in order to perform local search and to extend it to multi-robot exploration [13].

Conceptually, the approach may not be able to cover the entire environment, because it is based on frontiers search and the robots can get stuck in local minima. This happens when a robot has completed the exploration in its region and there are no other regions to explore in its vicinity. To recover from local minima, the strategy makes use of a topological map to store a road map in the free space of the environment. The topological map stores every position (in form of nodes) where a region of frontiers was already explored. Whenever a robot acquires new sensor data, it localizes itself in the topological map and sets a node in its position if no previous node is close.

To pursue with the exploration mission, when finding local minima, the algorithm will not return directly to a frontier cell. Instead, the robot will sequentially take the shortest path through the nodes already visited while searching for frontiers. As soon as a frontier is detected, the robot will move towards it.

In the left frame of the Figure 2.8, the blue line represents the path performed by the robot, while the red line represents the recovery path taken by the robot to return to the exploration. In the right frame, the pink circle represents the local minima where the robot got stuck.

Extending this approach to MRS, each robot is aware of the other robot's exploration state, knowing its selected exploration view and an estimate of its visibility. Yet, this information is only shared with other robots when information is updated.

Combination of information gain and distance cost

Colares *et al.* have recently presented an approach based on the classical frontier-based strategy [14]. They have proposed an utility function that takes in consideration the information gain and the distance cost of the frontiers to perform exploration. Considering the information gain of a frontier, an evaluation is carried out on how much new information can a robot get when scanning there. Using this information gain together with the distance cost to that frontier region, allows for a reasonable and efficient assignment of frontiers to

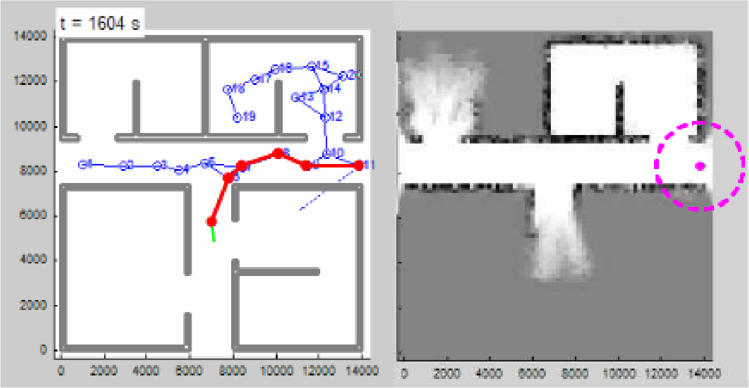
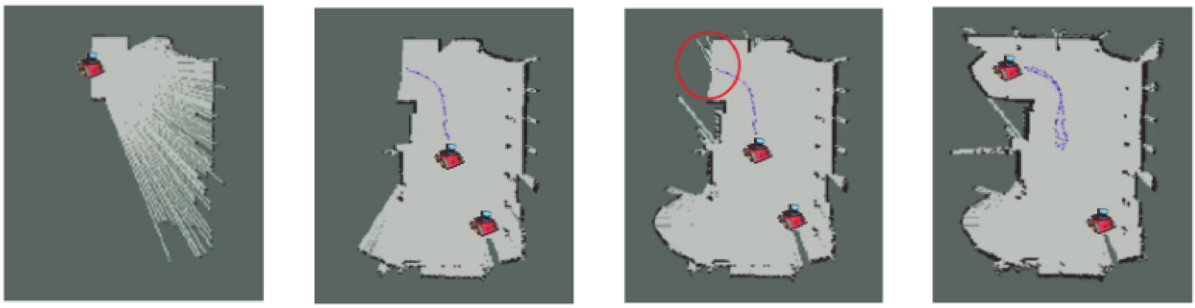


Figure 2.8: Local minima (right frame) and path followed by the robot to return the exploration mission (left frame) [13].



(a) (b) (c) (d)

Figure 2.9: Sequence of images representing the coordination strategy proposed by Colares *et al.* [14].

each robot, considering their position in the map.

This technique relies on a decentralized approach whereby each robot is able to plan and explore autonomously. Considering that robots sometimes meet their teammates, when this happens, they both merge their estimated maps and the utility function is calculated for the distance and information factor on the new merged map in order to get the highest utility. Therefore, the robots should choose to explore the regions far from other robots.

Figure 2.9 represents how the coordination strategy assigns targets to robots. In (a), the robot “wakes up” and explores towards frontier areas, when in (b) it meets with another robot. After both merge their maps, in (c), the first robot assigns to itself the distant frontier considering the other robot, marked with a red circle. In (d), the robot moves towards to the assigned frontier.

Strategy	Local approach	Global aproach	Information sharing	Conflict avoidance	Descentralized approach	Cost utility
Yamauchi [11]	no	yes	constant	no	no	yes
Burgard et al. [12]	no	yes	constant	yes	no	yes
Rocha et al. [13]	no	yes	when updated	yes	yes	yes
Colares et al. [14]	no	yes	when robots meet	yes	yes	yes

Table 2.1: Comparative table between presented multi-robot cooperative strategies.

2.3 Multi-Robot Exploration Strategies Overview

Table (2.1) summarizes the weaknesses and strengths of the different methods surveyed in the previous section. As described therein Section 2.2, these strategies perform better than others under specific circumstances.

Gathering all the information from the presented strategies and also inspired in additional work, the approach presented in this dissertation should be able to cover the above mentioned strengths and suppress possible weaknesses.

Analysing Table 2.1, the approach should be as balanced between local and global approach as possible. Whereas the local approach can explore in detail the nearest surroundings of the robot and identify when no more additional new information can be obtained in the area, the global approach can explore beyond the area around the robot, reaching new unknown locations to explore.

Having a central machine processing all the sensed data (*i.e.* a centralized system) becomes unpractical or unfeasible as more robots are integrated into the mission, which requires additional processing time, and hugely limits the scalability of the approach. Instead, a decentralized system allows the division of the computation load of the system and each robot has its own processing unit. As such, the system should be decentralized and robots should be independent from each other and fully autonomous. In this scenario, if one robot fails, the others can proceed with the exploration or, if the communication fails, it can operate blindly until it reconnects and shares the information acquired alone.

In order to not overload the network that connects the robots, the information has to be reasonably managed. Robots' poses and goals can be shared continuously since it involves small data chunks, being very relevant to avoid conflicts. Sharing local maps with teammates involves large data chunks, therefore the sharing frequency needs to be carefully selected to share the essential and to not overload the network.

Also, 3D exploration strategies are an important subject of study, since we intend to present a 3D exploration strategy in this dissertation. K. Ebadi *et al.* [32] presented a centralized SLAM system for subterranean operations in uneven terrains, which includes a Light Detection And Ranging (LiDAR)-based front-end, and a flexible and robust back-end that automatically rejects outlying loop closures. While the front-end estimates odometry, detects loop closures (using a 3D LiDAR) and artifacts in the environment (using vision), the local back-end uses Pose Graph Optimization (PGO) [33] to estimate the trajectory of the robot. Each robot sends its data (pose graph, pose estimation and point clouds) to a centralized base station where this information is fused, searching for potential loop closures between the maps. The paper also presents key SLAM challenges and how they can be mitigated, e.g., fusing information from other sensors. R. Rocha *et al.* address two main problems in [34]: *i*) the development of a probabilistic model for vision-based 3D mapping and frontier-based exploration using information theory and; *ii*) share information efficiently through communication in a team of cooperative mobile robots, driven by information utility maximization. Each robot starts with an initial map that is updated when new and useful measurements of the environment are acquired. These batch of measurements are shared with other robots so all the robots can update their local map. When the map is updated, a new viewpoint for the sensor is chosen and the robot starts to move there. New acquisitions are acquired during the navigation, proceeding in the same way, sharing and receiving new updates to/from other robots. The system proposes a grid-based probabilistic representation of 3D maps, enabling to explicitly model uncertainty within those maps.

To summarize, we have gathered all the requirements of our study, and we are now in a position to present a well-defined approach to perform 3D exploration missions with a team of mobile robots in irregular terrains.

3 Proposed method for 3D mapping and exploration

The method proposed in this dissertation was designed by casting the two-level coordination strategy for multi-robot patrolling missions [35] and adapting it to 3D exploration in this work. This chapter presents this new method which was originally designed by Dr. L. Freda, who has been a tight collaborator in the course of this dissertation. However, at the time of the beginning of this dissertation project, the method needed some improvements and consolidations in order to be ported to a team of Pioneer 3-DX robots and to be experimentally validated. The developed work and techniques used to improve it are also presented in this chapter.

3.1 The Two-Level Coordination Strategy

The multi-robot coordination method adopted assumes a decentralized system, where each robot works independently and autonomously, *i.e.* each robot has the capability of sensing, planning and moving autonomously in the environment. Using the Receding Horizon - Next Best View (RH-NBV) approach [9], the robot selects where to scan next based on the information that it already has about the environment. After reaching the target pose, a sampling-based node tree (RRT) is expanded in the traversable terrain of the 3D map. For each node, a gain value is assigned considering the probability of acquiring new information at that pose, and the robot proceeds to move to the next best node of the best branch of the tree. Further in the exploration mission, when two robots are in a nearby area, each one has the responsibility to select the most efficient target and plan the most suitable path to it, by taking into account the node and path selected by the other. This way, robots avoid conflicts among teammates.

In order to achieve cooperation, the robotic team is connected to a common network

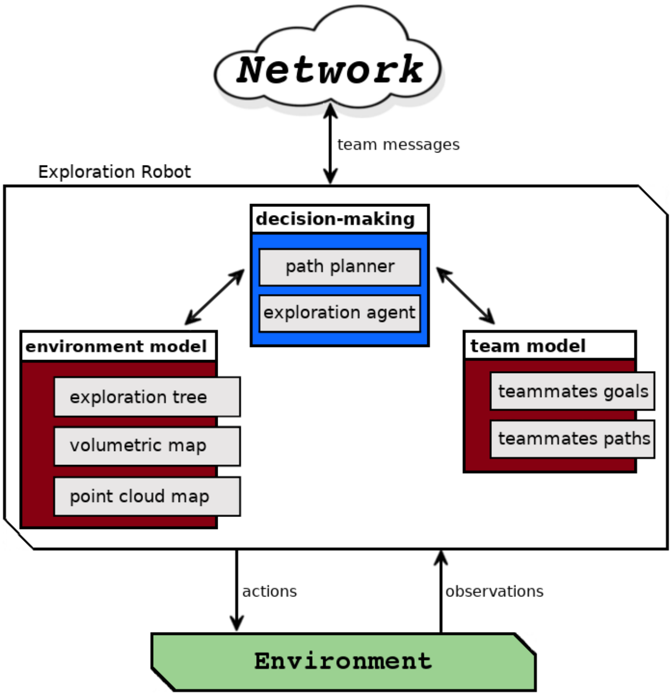
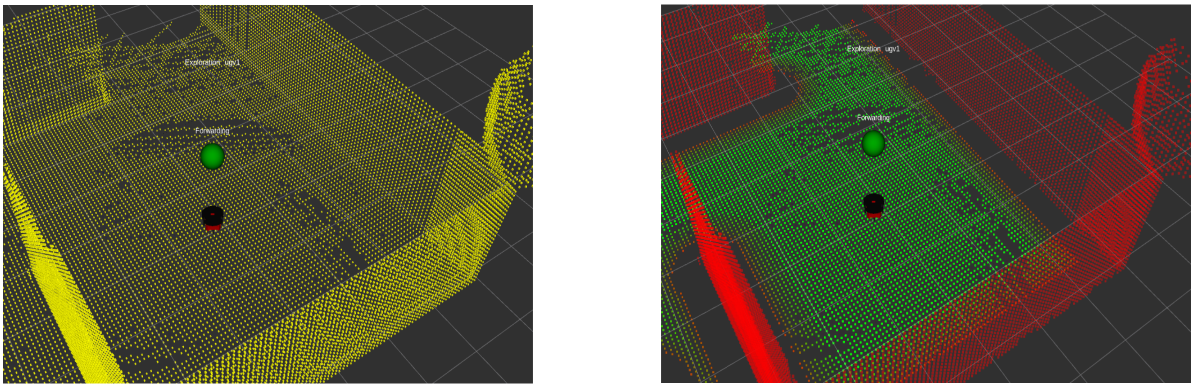


Figure 3.1: Representation of the exploration model on each robot.

in which they exchange messages concerning their observations and actions. The decision-making of each robot is achieved by a path planner and an exploration agent collaboration, taking the exploration process in consideration. These two - the path planner and the exploration agent - have the necessary information based on two models: an environment model that corresponds to the environment information concerning its conditions; and a team model, providing the knowledge of the belief and current plans of teammates. Figure 3.1 represents the exploration model of each robot.

Environment model

At the beginning of the exploration mission, robot j has its first metric map representation M_j (Figure 3.2), of the environment (E) in form of a 3D point cloud. Metric maps are grid-based and can offer a detailed description of the environment [36]. These are constituted by a matrix of values, each corresponding to a location in space. Higher values indicate a higher probability of that space being occupied and lower values, higher probability of the corresponding space being empty. The points of the 3D point cloud are segmented and partitioned into *wall* and *no wall* points, which will determine where the robot can or can not plan its movements, in the traversable and obstacle maps, respectively. A region is defined as traversable if the scanned surface is comprehended between a fixed pitch angle interval, negative and positive, for descending and ascending terrain. If does not comprehends, it is



(a) 3D point cloud of the explored area.

(b) Segmentation of the point cloud in *wall* (red) and *no wall* (green).

Figure 3.2: Segmentation of the map representation, M_j.

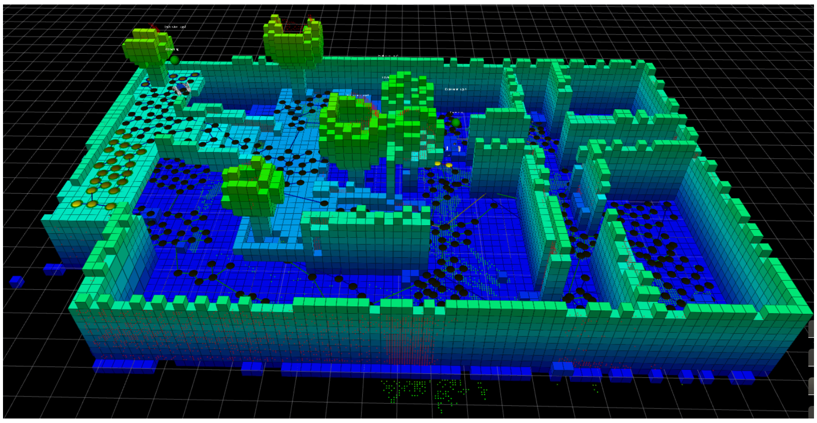


Figure 3.3: Volumetric map, V_j, represented in the form of an octomap.

partitioned into obstacle region. The surface where the robot is placed is used as reference to the segmentation of the scanned surface.

A volumetric map, V_j (Figure 3.3), is built to represent the explored region and to associate an information gain to each safe configuration. This map is a probabilistic occupancy grid map stored in the form of an octomap [20] and it is represented by *free*, *occupied* and *unmapped* cells with a maximum predefined resolution. In order to compute the information gain, two regions should be considered: the obstacle and the free boundary. The obstacle boundary is where the obstacles are detected. The free boundary is where there are potentially explorable regions. The information gain is computed based on these boundaries. When no obstacle is detected within the free boundary, that region will have higher probabilities of having an explorable region. The inverse happens when an obstacle meets the free boundary, less likely to be a high-gain region for new information.

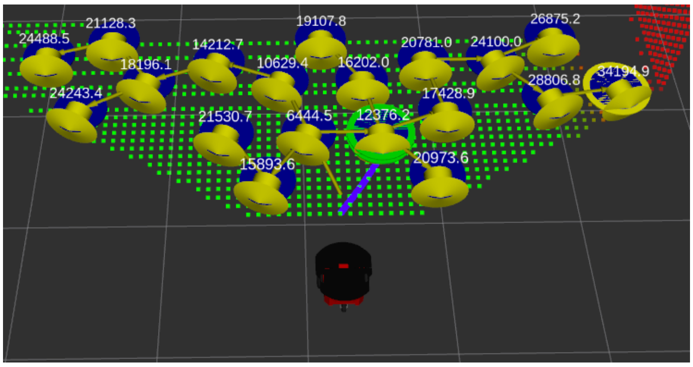


Figure 3.4: Search tree S_j locally expanded around the current view configuration. The node marked in yellow represents the candidate view node and the node marked in green represents the next best node in the exploration tree.

A search tree S_j is used to plan the next pose to where the robot should move and acquire new information (Figure 3.4). This tree is expanded from the current view configuration through the traversable map within a defined radius centred in the current view configuration. Here, each node is set with a value of utility based on the V_j configuration. The exploration agent chooses the best view node to visit. However, the robot does not move to it; instead, the robot moves to the next best node, according to the RH-NBV strategy mentioned in Section 2.1. The next best node is selected based on how much more of new information it can be acquired in that view node.

Related to the previous tree, the exploration tree, K_j, composed by nodes, represents the already explored regions in the environment. This is considered as a topological map. Topological maps produce graph-like maps that can be used much more efficiently [36]. They are simpler, permit efficient planning and do not require accurate determination of the robot's position. In these maps, nodes correspond to locations or landmarks, which are connected by edges that represent the distance between them. Each node is denoted as *view node* and represents an explored local region contained within a ball of radius centred at the associated view configuration, where the robot stops to acquire a new scan. An edge between the nodes represent safe paths connecting view nodes.

Every time a robot registers a new observation, all these maps are updated. The metric map will acquire additional points to update the traversable and obstacle regions. The volumetric map will update the octomap representation of the explored area. The exploration planner will make use of these to update the frontier tree and redefine the utility values of each frontier node based on new octomap cells, when necessary.

The points from the segmented point cloud of the metric map are associated with a multi-robot traversability cost function (Equation 3.1) [35] which is meant to be used in the navigation cost to compute a safe path to the robot. Given the current and the goal poses of the robot, the path planner will compute a safe path which minimizes the navigation cost between them.

Team model

The main aim of the team model is to avoid conflicts between teammates. Each robot maintains an internal representation of the planning state of the team, storing current positions, selected goals, last computed safe paths, associated cost-to-go (length of safe path) and the timestamp of the last message used to update this information. In this sense, a robot makes use of the team model to get the teammates' plans and actions in order to plan the most suitable path to the target pose.

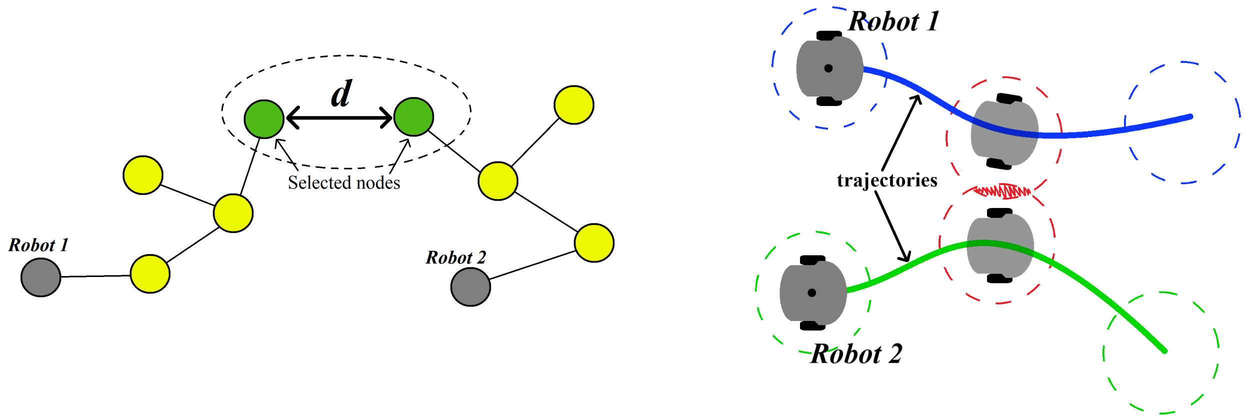
Team communication protocol

In order to share information, robots broadcast their messages through a common network. The messages are sent with an ID of the robot in order to inform others of various pieces of information such as if the robot is reaching, planning, selecting or aborting a goal node and when acquiring a new observation. This message broadcast also allows robots to share their volumetric map, exploration tree and observations amongst teammates so that they are aware of which areas have and have not been explored.

Assuming that messages sometimes may be lost during communication, the volumetric map shared may not correspond to the explored region. Therefore, similarly to the keep-alive persistent connection concept, messages are continuously broadcasted from each robot in form of a tree message [37] at a fixed frequency to the network. Every time a robot h receives a message from j of its volumetric map V_j, the robot h checks if V_h sufficiently overlaps with V_j or if it missed some integrated observations, requesting the missing observations if needed.

Two-level Coordination Strategy

The two-level coordination strategy for MRS [35] is supported by a topological and metric level and it boils down to solving conflicts. Topological conflicts happen when two or more robots try to concurrently select a new view node in a close spatial region (Figure 3.5a). The metric conflicts happen when two robots intend to follow a path that will cause interference if they are too close (Figure 3.5b). For both situations, the solution is to enforce a safety



(a) Topological conflict.

(b) Metric Conflict.

Figure 3.5: Illustrations of the two possible conflicts when exploring during mission.

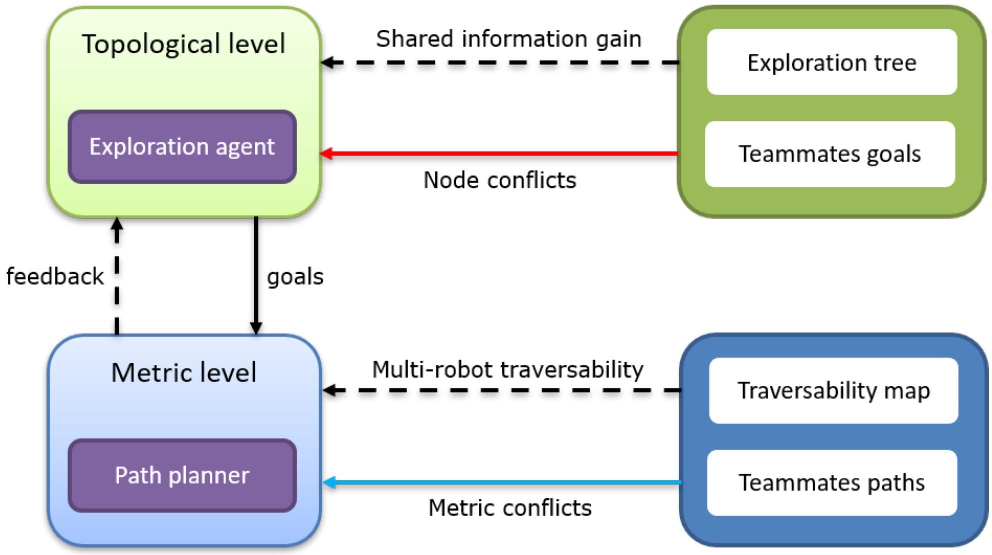


Figure 3.6: Two-level coordination strategy workflow for each robot.

pre-defined distance (d).

The **exploration agent** fits into the topological level by planning and adding a new view node on its exploration tree. Here the cooperation is achieved once the planning of the new node is driven by shared information gain. Whenever a conflict occurs, the robot with the smaller path length should proceed while the others stop and re-plan a new view node. Continuous monitoring and negotiation of conflicting nodes helps to maintain cooperation along the exploration mission.

The **path planner** fits into the metric level by computing the safest path from the current position of the robot to the goal position (p) using its traversable map. Applying a multi-robot traversability function, presented in equation 3.1, guarantees metric coordination by inducing prioritized path planning where robots negotiate metric conflicts, preventing the intersection of planned paths.

The traversability cost function is computed as:

$$trav(p) = w_L(p)(1 + w_{Cl}(p))(1 + w_{Dn}(p))(1 + w_{Rg}(p)) \tag{3.1}$$

Where the weight w_L depends on the point classification; w_{Cl} the multi-robot clearance[1]; w_{Dn} depends on the local point cloud density; and w_{Rg} measures the local terrain roughness (average distance of outlier neighbour points from a local fitting plane) [35].

Both agents collaborate as the explorer assigns the desired goals to the path planner, which continuously replies with feedback, informing the other of its status and computations. The behaviour of the two-level strategy is depicted in Figure 3.6.

This approach allows to reduce interference and manage possible deadlocks. While the exploration agent focuses on the most important exploration aspects, the path planner deals with possible incoming metric conflicts. Furthermore, where the path planner agent may fail in arbitrating challenging conflicts, the exploration agent intervenes and reassigns tasks to distribute the robots over the environment.

3.2 Spatio-Temporal Filter

In the process of collecting data through intensive simulation testing, we have noticed in the course of this dissertation project that several scans were being integrated into the robot's map without significant added value to the local map. In the TRADR project, the sensory system used was a 3D LiDAR, where each scan was acquired at every 3 seconds. In this work, a camera sensor can provide raw data at 30 frames per second, which raised the need to optimize computing resources.

When a scan[2] is acquired and added to the robot's map, if the robot has not moved significantly, there is no advantage in integrating that scan in the map as the additional information obtained is likely not relevant. The probability of that new scan overlapping the previous is very high, so an evaluation should be performed in order to decide if it should be integrated or discarded. For that reason, a spatio-temporal filter was developed to discard

[1] Traversable region which discards a prefixed distance to obstacles and radius of a robot in order avoid collision with these.

[2] Throughout this work, a *scan* denotes the most recent measurement acquired from the robot's onboard range sensors. In our setup, scans corresponds to point clouds acquired by the depth cameras used.

all those scans which do not have a valuable contribution to the robot's map. This was a key improvement introduced into the initial design of the method, which was implemented in the course of this dissertation work.

Another practical case of when the filter may be an advantage is when performing exploration with a numerous team of robots, where each robot is constantly sharing their scans under the same network. This excess of data may overload the network, as information may get lost, for that reason, the filter allows to reduce the data transmission rate. Besides of saving bandwidth of the network, it may also reduce computation load in each robot which will allow to use more of that computation in the exploration model.

The spatio-temporal filter is designed to propagate relevant information taking into account the where and when the information was acquired. Accordingly, the filter handles two main fronts, one relying on space (where) and the other on time (when).

The spatial front part is primarily concerned with the robot pose where the scan was acquired. The filter evaluates the current pose of the scan and compares it to the poses where the previous scans were integrated. Recalling that pose is, in fact, composed by position and orientation, these two are evaluated separately, considering adjustable thresholds, one for the position and another for orientation. If the new pose surpasses any of those thresholds, the scan is then integrated and its pose is added to a list of poses, so that it can be used as a reference for subsequent analysis.

The temporal front is essential to avoid registering permanent objects in the map that refer to dynamic objects that only temporarily obstruct the passage of a robot, such as other robot or a person passing by. When a dynamic entity is mapped, it is then considered as an obstacle in the robot's map. If a new scan is acquired after a certain amount of time, that dynamic entity may not be there anymore and the area that once was occupied is now free and available to navigate through.

On the other hand, if the robot cannot plan a new goal position, possibly because there is something on its map obstructing it, after a certain timeout, a new scan is integrated, which may then allow the robot to unlock itself and proceed with the mission.

3.2.1 The algorithm

Algorithm 1 provides the pseudocode of the spatio-temporal filter developed. We can observe that the spatio-temporal filter receives as inputs, the information of the depth sensor in the form of a point cloud (S_i) and the pose (q_i) of where the scan was acquired, and it returns

Algorithm 1: Spatio-temporal filter for scans.

```
FilterPointCloud($S_i$, $q_i$)
$t_i \leftarrow$ extract timestamp of $S_i$
if $Lf$ is empty then
    $Lf \leftarrow q_i$ // $q_i$ is added to $Lf$ set, where $Lf$ represents the list of poses of validated scans
    $t_i^* \leftarrow t_i$ // $t_i$ is the new timestamp of $Lf$
    return true;
end
$q_i^* \leftarrow$ extract the closest pose of $q_i$ in $Lf$ set
$t_i^* \leftarrow$ extract the timestamp of $Lf$ set
$D_q \leftarrow q_i^*$ - $q_i$
$\Delta s \leftarrow t_i$ - $t_i^*$
if $D_q > D_{thresh}$ or $\Delta s > time_{thresh}$ then
    $Lf \leftarrow q_i$ // $q_i$ is added to $Lf$ set
    $t_i^* \leftarrow t_i$ // $t_i$ is the new timestamp of $Lf$
    return true;
end
return false;
```

true or false whether the scan is integrated or not.

In the beginning, the filter checks whether the list of poses which maintain validated scans, Lf, is empty. If it is empty, it means that the scan (S_i) is the first one to be validated so the pose, q_i, and timestamp, t_i, of the S_i are inserted into Lf. If Lf is not empty, then the filter, using the k-d tree algorithm [38], finds the closest point to q_i from the Lf list and, from that, it evaluates if their Euclidean distance and rotation surpass the threshold set as reference (D_{thresh}), either on Euclidean distance or orientation. If so, the filter validates the scan. Moreover, the filter evaluates the temporal using a threshold too, comparing the difference of time from the S_i, t_i, and the last time a scan was validated, t_i^*.

3.3 Sensing system

In order to perceive the environment, robots are equipped with sensors to capture environment features throughout the mission. Some types of sensors can be adopted in 3D exploration missions, such as 3D LiDAR and cameras, etc., all with different strengths and

(a) Mynt Eye S1030.

(b) Intel RealSense D435i.

(c) Orbbec Astra S.

Figure 3.7: Depth cameras used.

weaknesses. Three depth cameras were made available in the course of this dissertation work. One of them is based on the Structured Light (SL) principle, and the two others are Infrared (IR) stereo cameras (Figure 3.7).

IR Stereo cameras make use of two cameras placed at a known distance, called baseline, in which each one takes its own image, separately but approximately synchronized. Those two images are then triangulated, extracting the same features in both images, and the depth of the scene is reconstructed. In order to better perceive features from the environment, these cameras project an IR pattern into the environment. Stereo has great performance on outdoor environments since it is a passive sensor, so if the scene has enough luminance, the stereo will work fine. The same thing goes for indoor environments. Yet, these cameras have the constraint that if the scene is textureless, such as white walls, the triangulation will be harder to perform. Another weakness of these cameras is the uncertainty of depth as distance increases.

SL cameras consists in projecting a known IR speckle pattern (light pulses) onto the scene. The depth estimation is performed by perceiving how that pattern is deformed in the scene. These cameras are very accurate in its depth estimation at near distances. Since these cameras run with IR light, its propagation may be spoiled if there is interference with sunlight. Therefore, these cameras are not meant to operate in outdoor environments.

The implemented system revolves around on the stereo depth camera Mynt Eye S1030 (Figure 3.7a). As the Field of View (FoV) is one of the most significant features to consider, this camera provides a wide FoV and a long range. A second stereo camera was tested: the Intel RealSense D435i (Figure 3.7b). This camera provides a well-supported community and some stability in the market against the recent Mynt Eye S1030. A benchmark between Mynt Eye and RealSense was carried out and is presented later in Chapter 5.

Since stereo cameras face some difficulties in perceiving surfaces with no textures, a SL

Camera model	**Type of camera**	**Depth Resolution**	**Acquisition frame rate**	**FOV**	**Range (m)**	**IMU**	**indoor/ outdoor**	**Dimensions (mm)**
Mynt Eye S1030	IR stereo	752 x 480	60FPS	122°H x 76°V	0.5 - 18	√	both	165x31x30
Realsense D435i	IR stereo	1280 × 720	30FPS	86°H × 57°V	0.1 - 10	√	both	90x25x25
Orbbec Astra S	SL	640 x 480	30FPS	60°H x 49.5°V	0.4 – 2	x	indoor	165x30x40

Table 3.1: Relevant features of the depth cameras analysed.

depth camera was introduced to give the robot the knowledge of the traversable region in front of it with the accuracy needed. The SL camera used is the Orbbec Astra S (Figure 3.7c).

Relevant technical specifications of all these cameras are presented in Table 3.1.

3.4 ROS - Robot Operating System

This work was developed using the ROS [39], an open-source middleware specially designed for robotic development. ROS provides a structured communication layer over the host operating systems of a heterogeneous computing cluster. Developed to manage the complexity and facilitate rapid prototyping of robots software, there are already many different distributions of this framework available and it is the most widely used software in robotics.

ROS is organized in packages, which may contain ROS nodes, libraries, datasets, configuration files or anything else that is useful in a modular implementation. The basic ROS concepts[3] are presented as follows:

- **Nodes:** The main common processes capable of performing computation, executing tasks and sharing information.
- **Master:** Provides name registration under the ROS network. Without the master, nodes cannot find each other.
- **Topics**: Provide asynchronous communication between nodes. Nodes publish a message of a certain type into a given topic and whenever a node is interested in that data type, it will subscribe to the appropriate topic to receive data.
- **Messages:** Used as the main data structure that nodes use to communicate with each other. Nodes publish messages into topics so that other nodes subscribe them.
- **Services:** Provide synchronous communication between two nodes. They use request and response methods to receive data from other nodes.

[3] For a more comprehensive understanding of ROS concepts, one should refer to `www.ros.org`.

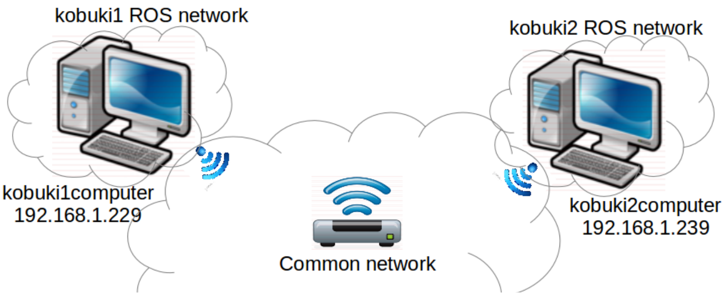


Figure 3.8: Simple network setup for the single computer ROS network case to exemplify the network configuration.

3.4.1 ROS Packages

Since ROS is highly modular, in order to work with different software and hardware, packages are commonly provided in open-source so that the community can reuse them instead of developing functionality from scratch.

The RTAB-Map package

Real-Time Appearance-Based Mapping (RTAB-Map) [16] is a ROS package developed to perform 3D SLAM based on an incremental appearance-based loop closure detector. The loop closure detector uses an incremental bag of visual words [40] approach to determine how likely new observations are to come from a previous or new location. This package will be used in this work so that robots can estimate their pose in the environment.

The Multimaster_fkie package

The multimaster_fkie [41] [42] is a very versatile package. It allows robots or modules that are running independent ROS systems to communicate with other hosts. In order to do that, they should be under the same network and the Internet Protocol (IP) address from each machine should be well-known (see Figure 3.8). This starts a process of discovering the other machines and once the communication has been established, every node and topic running in one machine is then visible to all others in the communication channel. In this project, this package was adopted in order to establish communication between robots running independent ROS masters, so that they are able to share topics during the exploration mission.

The rviz visualisation tool

The ROS visualization (Rviz) tool allows to visualize specific behaviour of the robots as observations and actions during the exploration mission. This tool can display sensor data from cameras, lasers, whether they are 2D or 3D, and even represent any type of ROS information in the form of markers. It also provides 3D models of the robots and can even replay captured data from previous sessions or datasets. For instance, we use the Rviz mainly to visualize the map of the explored area being built in the form of a point cloud and/or 3D octomap.

3.4.2 V-REP Simulator

Testing the approach in a real environment is a time-consuming process that requires significant effort. Before proceeding with real world tests, it is common to test the system behavior in simulations, enabling the evaluation of several different configurations with distinct parameters, allowing for adjustments and improvements to the system. Therefore the V-REP was the simulator used to preliminarily test and simulate the system.

The V-REP simulator is one of the most well-known simulators [43] seamlessly interacting with ROS through ROS topics. Its versatility relies on its user-friendly interaction and its capability of supporting several distinct features. The simulator is also less demanding than others whose computation capacity would be impacted when working with a considerable amount of robots and sensors.

The simulator was mainly used to graphically represent an environment and run multi-robot exploration missions.

3.5 System Outline

To summarize, the presented two-level coordination approach is focused on preventing conflicts between robots. Many tasks require efficient cooperation between robots, which will have a significant impact on the mission progression when compared to a single robot solution. Reducing the overall time of a mission and cover a larger area at the same time [44] are not the only advantage of using a MRS, this can also introduce fault-tolerance, so when a robot fails, the others maintain the mission.

Another important advantage is the possibility of having many robots or an heterogeneous robotic team in numerous places, carrying out diverse tasks at the same time, *i.e.*

space distribution. Most missions are solved much quicker if robots operate in parallel. Increasing robustness and reliability of the solution is also feasible in MRS by introducing redundancies in the capabilities across robot team members and graceful performance degradation, remaining functional if some of the agents fail.

The robots take into account the information perceived by their teammates and use that information to improve their own knowledge about the environment and the team. By combining their shared perception, it allows the team to explore the environment in a coordinated way, guiding robots to explore areas which have not been visited by any of the team members yet.

In contrast to the state-of-the-art techniques presented in Chapter 2, this multi-robot exploration approach has the advantage to deal with 3D environments, not only mapping in 3D but also allowing the navigation in irregular terrains. The extraction of the ground plane and its conditions allows the robot to evaluate a traversable area where it can navigate through.

4 Ablation Study

When a new approach is developed, it has to be carefully tested and validated in order to become consolidated, and recognized by the scientific community.

Working with a team of robots in a real environment can be very toilsome, requiring long time and engineering effort. It is harder to respect the same initial deployment conditions of each test (*e.g.* initial pose of the robots, network connectivity, environment lights). To rearrange some parameter, robot, sensor or even add more robots to the team, is more practical in this type of simulation framework than in the real world.

Performing tests in an earlier stage of the approach will require to adjust configurations, in order to obtain the most suitable results possible. Only then, when the approach is stable, extend it to real-world environments. Also, the simulation framework allows each test to be performed quicker than in the real world by speeding the simulation process [45].

Therefore, the tests performed to evaluate the system are divided into two groups: one within a simulation environment and another with real-world exploration tests with a team of physical mobile robots (described in Chapter 5). The first group, the simulation tests, is described in detail in this chapter and took place throughout several weeks (for about more than one month).

4.1 Experimental Environment

4.1.1 Hardware

Simulations were executed in a computer available at the MRL-ISR laboratory. It uses Ubuntu 18.04 as the main operating system and it is empowered with an Intel Core i7-7700HQ Central Processing Unit (CPU), 2.80GHz with 8 threads, a Graphics Processing Unit (GPU) Nvidia GTX 1060 with 6GB dedicated VRAM dedicated and 48 GB of RAM memory. This computer was used to simulate exploration missions with as many robots as needed.

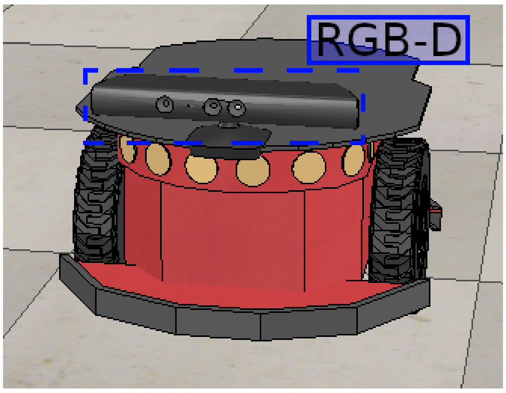


Figure 4.1: Pioneer 3-DX with onboard RGB-D camera simulated in V-REP.

The exploration team designed in simulations consisted of a differential wheeled robot equipped with forward-facing RGB-D camera. The differential wheeled robot used was the Pioneer 3-DX (Figure 4.1), which is later used on real tests. This can approximate the results obtained in simulation to real world tests.

For the simulated camera sensor, V-REP provides a depth camera that was adjusted with respect to the range and resolution. In our experiments, we only acquire depth information from the sensor and the Red Green Blue (RGB) data was disabled in order to save graphical computation resource. Since the octomap also downsamples most of the points when segmenting the point cloud, the resolution of the camera was also downgraded, to 256x128 pixels, also saving computation resource.

4.1.2 Software

For this project, the V-REP simulator was a convinient choice as it had already objects/models that could be used to represent real hardware and therefore, represented a platform that could provide us hints of what would be expected when performing real tests. The models used in simulations were set with parameters that closely respected the real specifications of the approach. Each object or model that was introduced in the simulator would cost processing time due to its renderization together with all the ROS nodes sharing data with the simulator. Therefore, a reasonable configuration was employed, aiming at a less demanding computation burden, yet providing appropriate simulation realism.

Initially, the virtual Pioneer robots were tested in V-REP to evaluate their behaviour in a simulated environment. As the simulator already contains models of commonly used robots and sensors, the effort consisted on how to program them so they could interact using ROS, that is, how data should be published and subscribed in the form of ROS topics. For

Parameters	exploration step	maximum range	extension range	number of tests	max simulation time (minutes)	Total tests
Values	[2; 5; 8; 11]	[7; 10; 15]	[0.3; 0.5; 0.7]	10	15	360

Table 4.1: Combination of parameters tested in the initial simulation stage.

this, V-REP has specific ROS interface using the LUA programming language.

4.2 Simulation tests preparation

In a first stage, the exploration approach needed to be adjusted in order to work adequately with a depth sensor along with the provided robotic platform, the Pioneer 3-DX robot. With this purpose, a set of tests was designed with multiple configurations, such as different scenes, number of robots and input parameters to gather relevant information to analyse the performance of each simulation.

In order to adjust the approach, different configurations with parameters that are expected to have direct influence over the exploration planner were selected (Table 4.1). The *exploration step* parameter corresponds to the maximum distance traveled by the robot from the root of the current search tree to the selected node, along the path towards the current best node. The robot, instead of travelling to the best node of the tree branch, travels to a goal node in the branch of the selected node located at a distance defined by *exploration step*. The *sensor maximum range* parameter is the maximum distance that the exploration planner will use from the sensor data to build the low-resolution octomap. It is in this octomap that the exploration tree is built. The *extension range* parameter corresponds to the predefined length between each node of the tree. This parameter is particularly sensitive, since if it is too low, the tree will not cover a large area in the field; if it is too high, will not be able to project the tree through short passages or along corners.

Simulations were initially configured with a considerable range of values for each distinct parameter, to perceive which configuration of values lead to the most suitable results. We then further discretize the parameters according to the results obtained to find an improved set of parameters. The initial ranges of parameter values to be tested are presented in Table 4.1. Combinations of all these values were tested in simulations, and since the approach results in a stochastic process due to the use of RRT trees, a number of 10 simulations for each combination were set. Therefore, having 36 different combinations for each environment, we performed a total of 360 simulations for each scene.

The maximum time of each simulation was set to 15 minutes of simulation time. In case something unexpected happens, the simulation stops after this timeout and proceeds to the next test.

The volumetric map of each robot, V_{robot}, comprises occupied and free cells. Only the occupied cells are used to calculate how much of the scene was explored, since the obstacles and walls in the environment are generally considered static during the exploration mission, and the free cells can easily fluctuate, because, in a scene that is not closed (*i.e.* without ceiling) the free cells are dependent on the sensor FoV.

In order to provide a suitable analysis of the simulations, several data was gathered from each test, such as poses of each robot, the number of free and occupied cells of the scene, the state and time of exploration, etc. All this data is logged during the simulation mission and stored, so that further assessment of the simulation mission can occur offline.

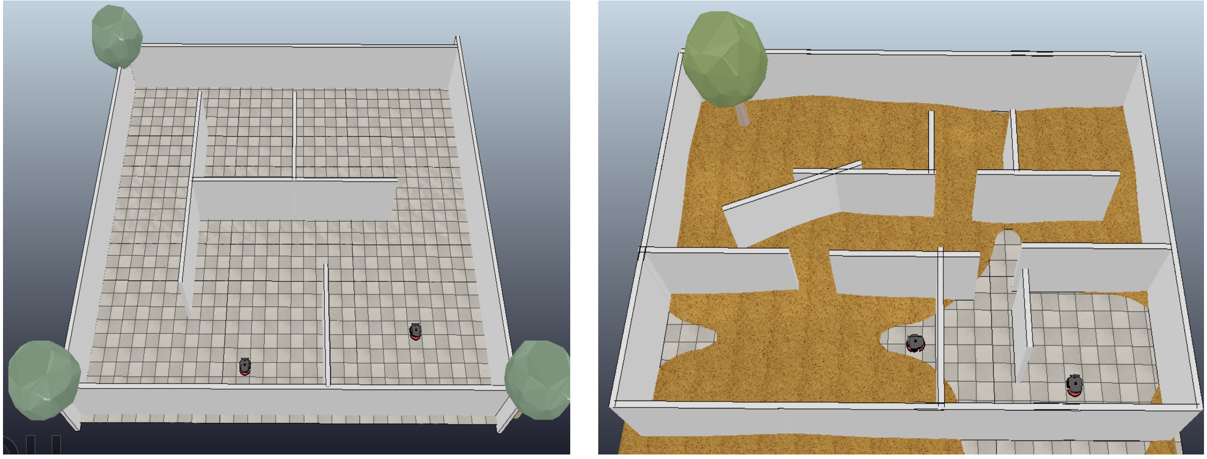
4.3 Simulated scenes

To perform simulations, several scenes were designed in V-REP. Figure 4.2 presents two of these scenes where the approach was tested using two robots. Three more variants of these scenes were implemented in order to test different simulation configurations.

In the *rooms* scenes (Figure 4.2a), a very simple environment with an open area is represented, where robots have plenty of space to move around. This scene has a flat surface, so robots will always keep the same plane for the traversability. In the *extreme* scene (Figure 4.2b), a denser scene is represented, where passages were designed to be tighter, making it more difficult for the robots to get through. Additionally, this scene has irregular terrain (illustrated with a brown texture) alongside with flat surface. Robots start their mission on the flat surface and soon conduct traversability analysis to check where they can move, enabling navigation in the irregular terrain.

Extending the exploration mission to more robots, in the *rooms2* scene, one more robot was added, as shown in Figure 4.3. Here, the *rooms2* scene is tested again with three robots and in addition to it, an irregular terrain is implemented (we named it *rooms3 3D* scene). This scene intends to compare how adding a new robot to the team will have an impact in the performance of exploration.

Another configuration to be tested was designed in another scene using four robots, as shown in Figures 4.4. Here, the robots were placed in different initial positions, one configuration with two pairs of robots starting in the same room, one pair in each room



(a) *rooms2* scene.

(b) *extreme* scene.

Figure 4.2: Scenes containing two robots.

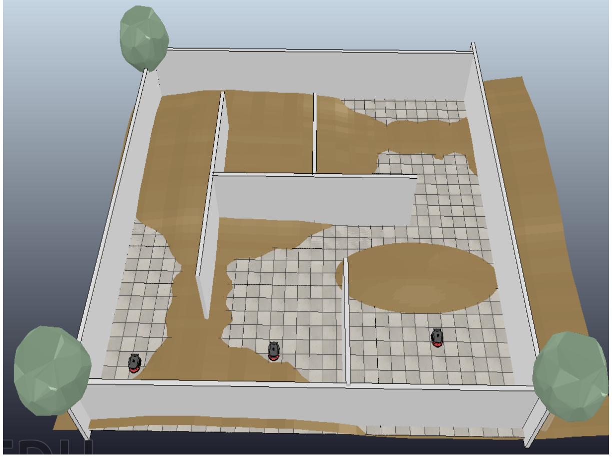
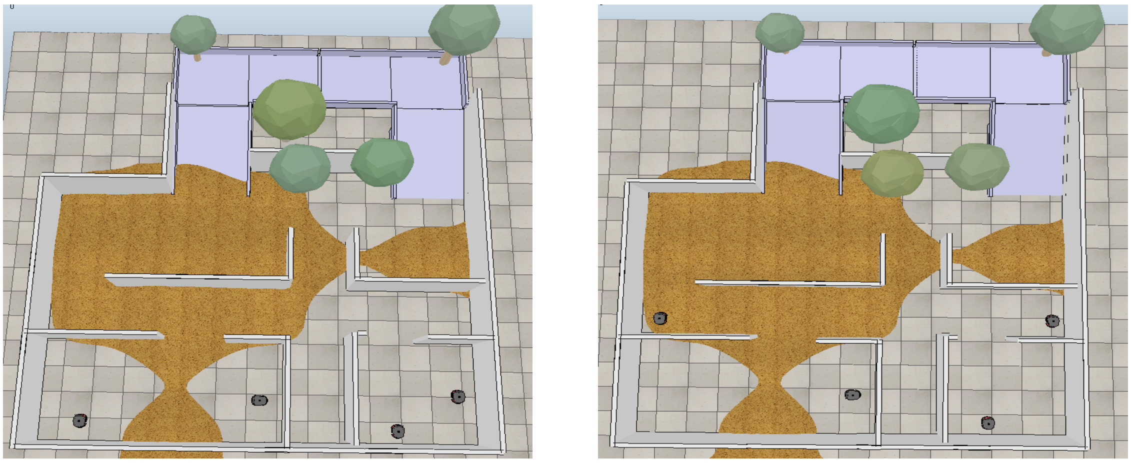


Figure 4.3: *rooms3 3D* scene - scene with 3 robots.



(a) *ramps* - scene with pair of robots in the same location.

(b) *ramps_2* - scene with pair of robots in different locations.

Figure 4.4: Simulated environments with irregular terrain and ramps containing four robots.

Scenes	**Width (mxm)**	**Area (m^2)**	**Volume (m^3)**	**Surface**	**Number of robots**
rooms2	15 x 15	225	540	Flat	2
extreme	14 x 10	140	336	Irregular	2
rooms3 3D	15 x 15	225	540	Irregular	3
ramps	17 x 14	214	514	Irregular	4

Table 4.2: Detailed specifications of the simulation scenarios proposed.

(Figure 4.4a), and the other with robots placed in distinct regions. For both configurations, the robots start in the same side of the scene, even if in different locations, so they can move to the opposite side of the environment, to the ramps.

In all the scenes presented, robots start always in the same location as they are located in the figures.

An overview of the specifications of the simulated scenarios described are presented in Table 4.2, considering their area, volume, number of robots and surface type.

4.4 Multi-Robot Exploration Simulations

Simulations were performed using the combination of parameters in Table 4.1. Each configuration was simulated 10 times. Therefore, for each configuration, we present the mean of the exploration time and the time spent by the simulator to run the corresponding exploration mission, denoted as t_{real} and t_{sim}, respectively. The percentage of occupied cells, *occ*, is also considered. Hereafter, the presentation form of a configuration test will be presented as a [expl_step; max_range; ext_range] set.

In the analysis of each trial, the simulation of the exploration mission is considered valid if the percentage of the number of occupied cells is at least 95% of the maximum number obtained in all simulation tests conducted in that scene. The simulation test is discarded if it does not explore 95% of the area within the 15-minute time window. Yet, it is important to underline that an exploration test can be successful without presenting 95% of the maximum number obtained in that scene. Even in the simulation environment, it was observable that the total number of occupied cells can differ from test to test. This metric is only used so that we can test different values for the parameters in order to obtain the most appropriate values for a greater number of valid tests. In the tables included in the next section, the

Scene	expl_step	max_range	ext_range	t_{real}	t_{sim}	*occ* (%)	valid
	(m)	(m)	(m)	(s)	(s)		
rooms2 (2 robots)	2	10	0.7	281	133	98.2	6/10
	5	15	0.7	290	142	98.3	7/10
	8	15	0.5	293	146	96.8	9/10
extreme (2 robots)	5	15	0.5	425	239	95.5	9/10
	8	15	0.5	441	248	96.6	6/10
	2	10	0.5	441	244	97.1	5/10

Table 4.3: Overview of the first tests with the application of the spatio-temporal filter.

number of valid simulation tests considered for each configuration is presented in column "valid".

4.4.1 Simulation Results

The first results, when the spatio-temporal filter was applied, can be observable in the Table 4.3. For these tests, the parameter values presented in Table 4.1 were used.

The ext_range has presented an average value in the *extreme* scene, which is suitable for the path constraints that it contains.

Following this, a new range of parameters has been selected, and the max_range was defined with a fixed value of 10. This is a reasonable value for the real depth sensors since depth estimation errors increase as the range increase. Moreover, this selection also considers the scenes structure that are being tested and their dimensions. Taking into account the results from Table 4.3, the new range of parameters for the exploration step are [3; 4] in order to present a RH-NBV exploration, and [0.35; 0.40; 0.45] for the extension range.

The results using these new configurations are presented in the Table 4.4.

The *ramps* and *ramps_2* scenes follows a stopping condition of only 90% of the explored volume. This is due to the scene size and the fact that it has more obstacle density than the remaining ones and consequently more cells obstructed by obstacles that may not be detected by the depth sensor leading to an increased variance of number of cells.

From the scenes *rooms2* and *rooms3 3D* it is possible to observe that the exploration time (t_{real}) has decreased when an additional robot was added to the team. Also, when analysing the exploration time of the scenes *ramps* and *ramps_2* it was possible to decrease the t_{real} placing the robots in sparse initial positions, covering more area at the beginning of the mission.

scene	expl_step (m)	max_range (m)	ext_range (m)	t_{real} (s)	*occ* (%)	valid
rooms2 (2 robots)	4	10	0.35	324	98.7	10/10
	3	10	0.45	325	98.9	9/10
	3	10	0.35	329	98.3	10/10
extreme (2 robots)	4	10	0.4	408	96.7	8/10
	3	10	0.4	451	100	9/10
	3	10	0.35	473	99.6	8/10
rooms3 3D (3 robots)	3	10	0.35	205	101	6/10
	4	10	0.35	205	99.5	6/10
	3	10	0.4	207	100	6/10
ramps (4 robots)	3	10	0.45	725	100	4/10
	4	10	0.35	727	100	5/10
	4	10	0.4	742	100	6/10
ramps_2 (4 robots)	3	10	0.35	610	99.6	4/10
	3	10	0.45	637	97.3	7/10
	4	10	0.35	659	96.9	10/10

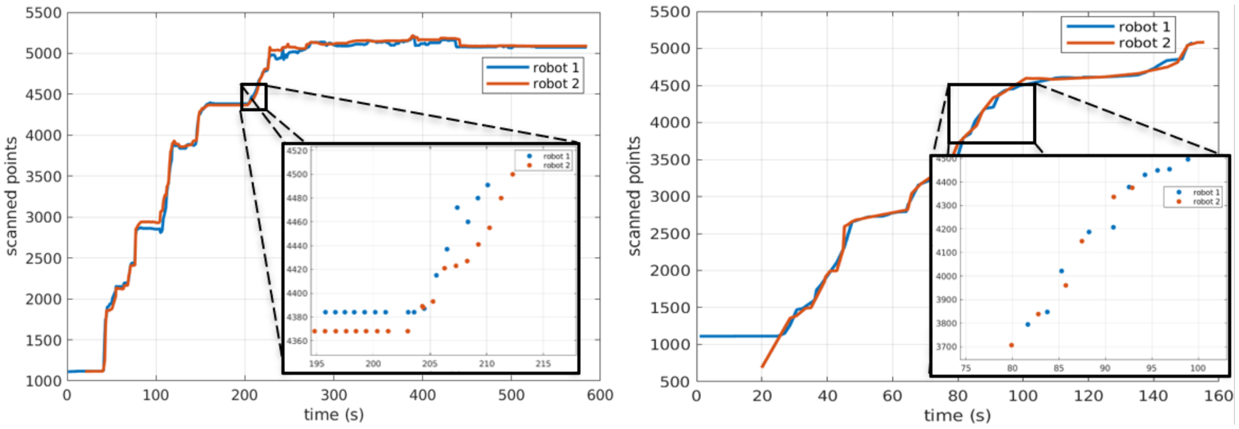
Table 4.4: Overview of the second tests with the application of the spatio-temporal filter.

Spatio-Temporal filter analysis

Regarding the spatio-temporal filter implementation, simple simulation tests without the filter applied were performed in order to analyse its influence in the robot's communication. For that reason, two graphs are presented in Figure 4.5 which compare the occupied cells in the local scene of each robot before and after the implementation.

Comparing the amount of scanned points integrated into the local map of a robot before and after the spatio-temporal filter is applied, it can be seen that both robots integrate approximately the same amount of points into their local map using the shared point clouds, with both robots growing their local maps similarly. Yet, when the amount of integrated data is analyzed, the number of messages exchanged between the two robots is significantly different. Looking into the zoomed portions of each graph, the sequence of point cloud integrations becomes more clear. Each point corresponds to one point cloud being integrated into the robot's local map. Before the filter was applied, the frequency of integrated data is periodic, and most of these without any new information about the environment. When the filter is applied, the frequency of integration is not periodic. There is much less data being integrated, leading to better management of the maps and protecting the system from data overload, specially when more robots are added to the team.

Data quantity was extracted in these simulations. Without the spatio-temporal filter, the rate of integration of scanned points of each robot was about 1 Hz, and in the simulation



(a) Without the spatio-temporal filter. Result with [2; 10; 0.7] set.

(b) With the spatio-temporal filter. Result with [2; 10; 0.7] set.

Figure 4.5: Amount of occupied cells integrated into the local map of a robot before and after applying the spatio-temporal filter in the *rooms2* scene.

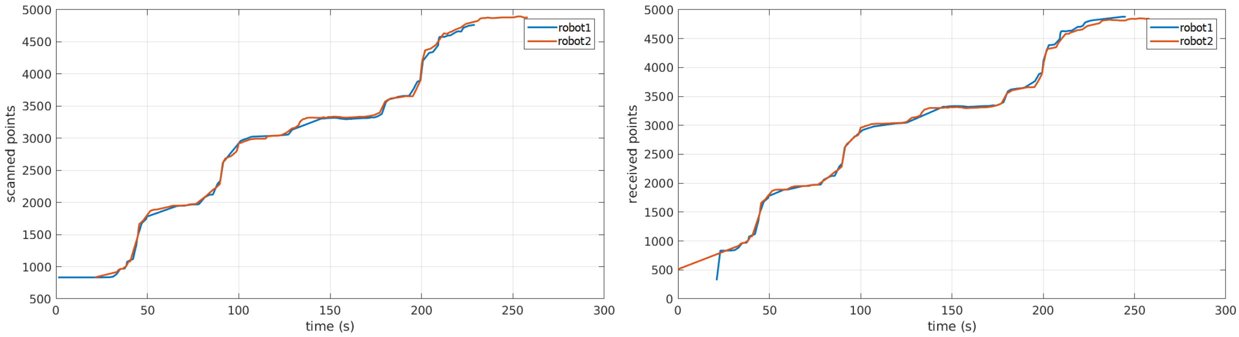
with the filter, those values were reduced to 0.54 Hz, corresponding to a mean reduction by 46%. The difference before and after the filter is even more noticeable in the amount of the received point clouds from other robots. Without the filter, each robot was receiving a point cloud at a rate of about 21 Hz. After the filter was applied, each robot was receiving point clouds at a rate of about 0.4 Hz, corresponding to a mean reduction by 98%. Therefore, the spatio-temporal filter allowed for a strong decrease in the required processing load as well as the amount of data shared among robots.

The graphs presented in Figures 4.6, 4.7 and 4.8 show the amount of points scanned integrated into the local map (Figures 4.6a, 4.7a and 4.8a) and those received from the teammates along the simulation (Figures 4.6b, 4.7b and 4.8b).

In the graph of Figure 4.7b, a little discrepancy is shown in the first 50 seconds after the simulation starts. This happens when a scan corrects the previous integrated one. Here, an amount of occupied cells can be scanned and integrated into the local map, then, in a new scan, if the same robot or another robot detects those same points as free cells, they are updated as free and the previously occupied cells will be removed.

The graphs presented (Figures 4.6, 4.7 and 4.8) describe the behaviour of the approach, showing that all the robots presented in the mission have the same amount of information about the environment, proving the efficient of communication between them.

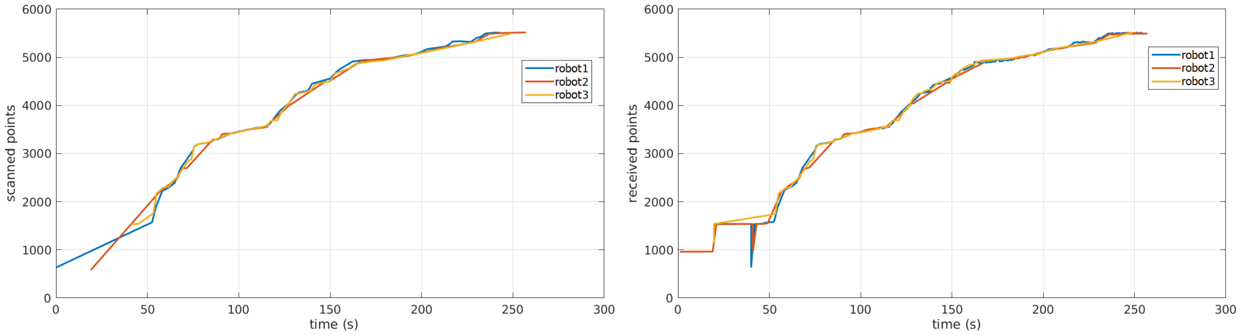
At the beginning of each mission, there may be a relevant variance of the number of points integrated, but that is because the robots are starting the simulation, and most data acquired is new. While robots are launching the system, scans may start to be acquired, but



(a) Points scanned by each robot integrated into its local map.

(b) Points received from the teammates inserted into the robot's local map.

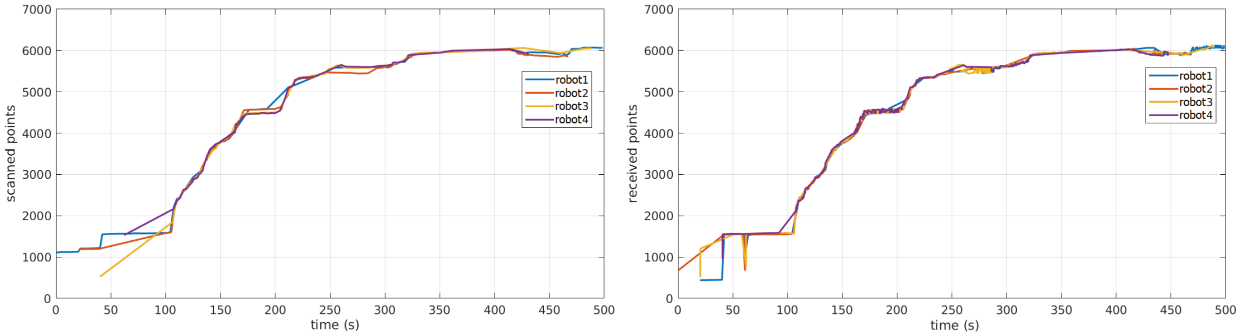
Figure 4.6: Scanned and received maps of each of the two robots in the *extreme* scene. Results with [2; 10; 0.7] set.



(a) Points scanned by each robot integrated into its local map.

(b) Points received from the teammates inserted into the robot's local map.

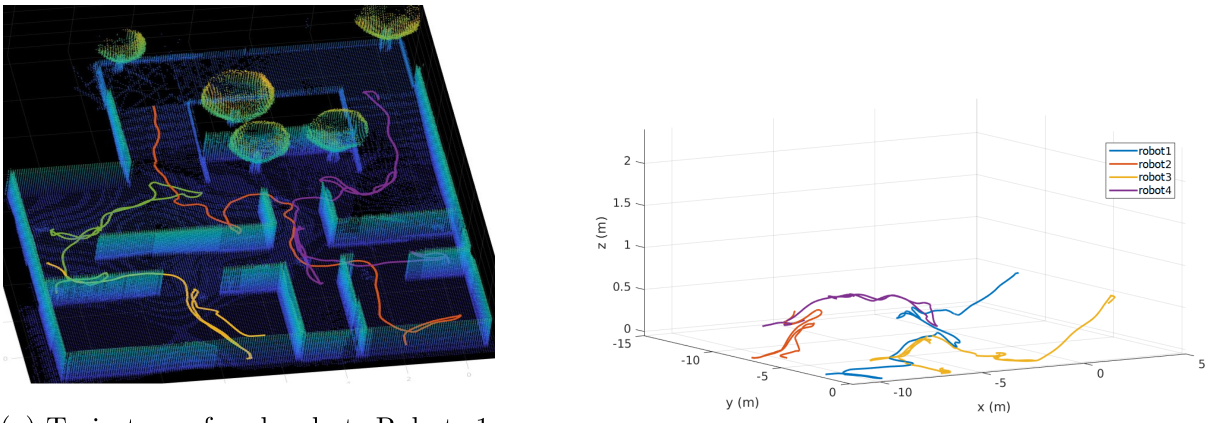
Figure 4.7: Scanned and received maps of each of the three robots in the *rooms3 3D* scene. Results with [3; 10; 0.35] set.



(a) Points scanned by each robot integrated into its local map.

(b) Points received from the teammates inserted into the robot's local map.

Figure 4.8: Scanned and received maps of each of the four robots in the *ramps* scene. Results with [4; 10; 0.35] set.



(a) Trajectory of each robot. Robots 1, 2, 3 and 4 represented in orange, purple, yellow and green, respectively.

(b) Trajectory of each robot in 3D.

Figure 4.9: Trajectory path performed by each robot in the *ramps_2* scene (2D and 3D perspective). Result with [4; 10; 0.35] set.

the exploration mission only starts after all robots are ready. Only when this happens, do the robots start to exchange information between them.

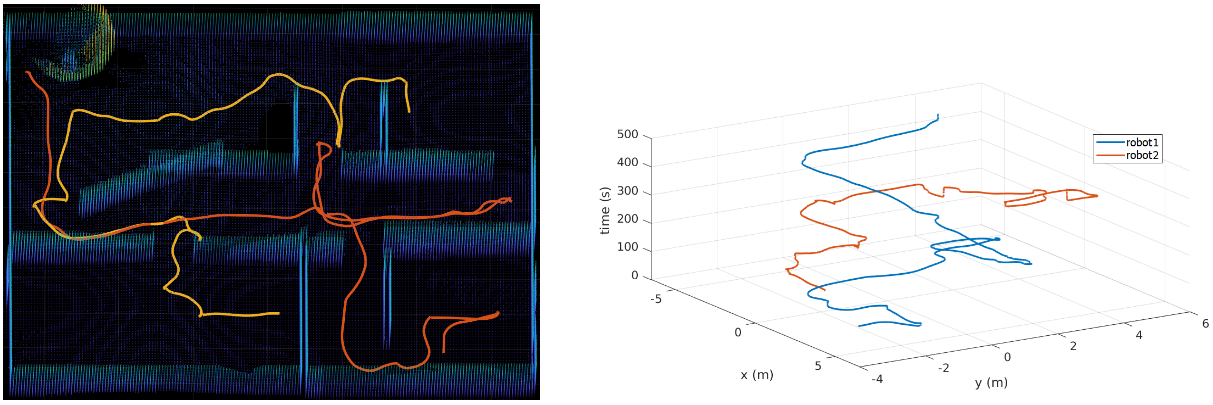
Robot Trajectories

Finally, in order to evaluate robot interference, the path taken by each robot was tracked during each simulation. Note that they have a threshold of distance to preserve from each teammate. In Figure 4.9, we present graphs of paths followed by robots from two different perspectives in the *ramps_2* scene, one showing the paths in top view of the scene, in 2D (Figure 4.9a), the other showing the 3D trajectories (Figure 4.9b).

In Figure 4.10a, the trajectory paths on *extreme* environment are presented. When comparing to paths presented in Figure 4.10b, where the trajectory paths are distributed along simulation time, it is possible to see that the distance between teammates is respected.

As seen from the figures, robots never interfered with each other regarding their distance or goal node. When there is potential for conflict, robots re-negotiate in order to have distant goals and paths. Also, from Figure 4.9b, it can be seen that the approach can easily project the robots in an irregular terrain.

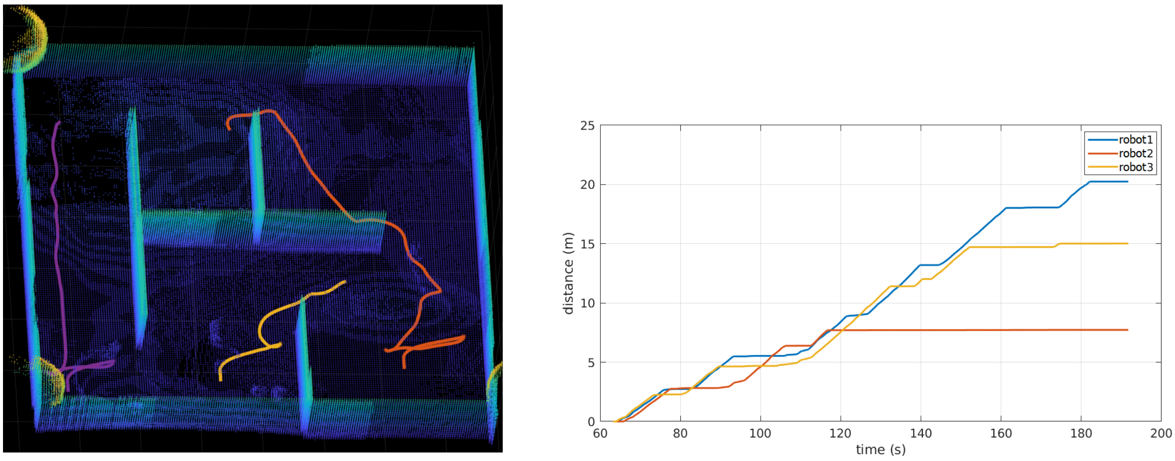
In the *rooms3 3D* scene, it is observable that a third robot does not really add value to the exploration of that environment. This can be seen in Figure 4.11, where robot 2 only explored the interior of the scene while teammates explored the rest of the entire environment. As soon as teammates explore the neighboring areas of the robot 2, this robot soon realizes that



(a) Trajectory of each robot. Robot 1 represented in orange and robot 2 in yellow.

(b) Trajectory path along the mission.

Figure 4.10: Trajectory path performed by each robot in the *extreme* scene. Result with [2; 10; 0.7] set.



(a) Trajectory of each robot. Robot 1 represented in orange, robot 2 in yellow and robot 3 in purple.

(b) Distance travelled by each robot.

Figure 4.11: Trajectory path performed by each robot and their travelled distance in *rooms3 3D* scene. Result with [3; 10; 0.35] set.

it has no more area to explore. A reasonable selection of how many robots need to be part of the multi-robot team should be taken into account in order to not have robots contributing little to the exploration mission.

4.5 Summary

From the results presented in this chapter, we can conclude that the system is able to employ a team of robots to autonomously perform 3D exploration, with no human intervention and providing a fully detailed 3D scene of the explored environment. The cooperation between robots has demonstrated that the time of exploration can be significantly reduced as long as more robots are added to the team. As the information is shared, all robots have a common vision of the environment with approximately the same amount of occupied cells, therefore, the same amount of explored area.

Also, exploration missions can be concluded with no interference between robots, as they prevent collisions and avoid areas already explored. There are times that robots can eventually move to an already explored area. However, that happens because some unmapped cells, yet to discover, are detected in the exploration model.

Furthermore, the spatio-temporal filter showed to be efficient in saving network bandwidth, while sharing information that robots need to finish the exploration with success.

5 Experimental Validation and Results Analysis

Real tests are the final proof of every developed system. Even if the system presents efficient results in a simulation environment, this may not be considered as a completely functional system and may overlook important practical aspects that are difficult to simulate. When performing tests in a simulation environment, reduced number of errors are expected, if any. A real environment test has huge of external factors which may induce errors to the sensory system, as light conditions or reflections of the materials (depending on the sensing system used), or to the actuation system, as odometry or localization errors which accumulate along time, may have negative impact on the results. Another sensitive case is the noise obtained from the sensor sources, which limits the performance that can be obtained from them, compromising the final results. The real proof of a functional system relies on the capability of the system to overcome noise and potential errors and noise and to suppress them.

5.1 Setup of Experiments with Real Robots

Test areas

The first experiments with real robots took place at the Institute of Systems and Robotics shared experimental area (ISRsea), a dedicated room for tests of this type. A labyrinth was created so that robots could explore, plan and navigate through it. The scene created can be seen in Figure 5.1. This room has a size of 13 x 5 meters with a height of 4 meters.

The downside of this room is that robots are restricted to navigate in a flat surface, without any irregular terrain. Therefore, a more complex scenario was also tested, the *A3 Auditorium* of the Department of Electrical and Computing Engineering of the University of Coimbra (Figure 5.2), which contains some ramps. This environment serves the purpose to fulfil the specifications of robots to navigate on an irregular terrain. The *A3 Auditorium*

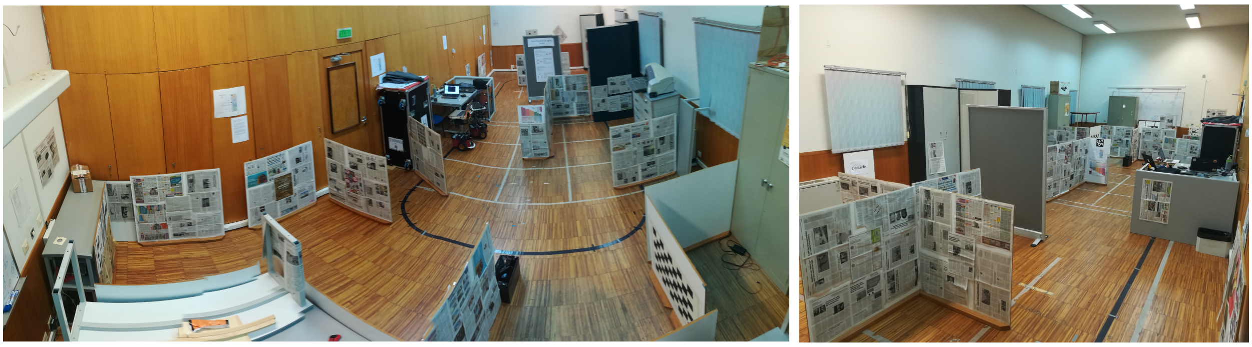
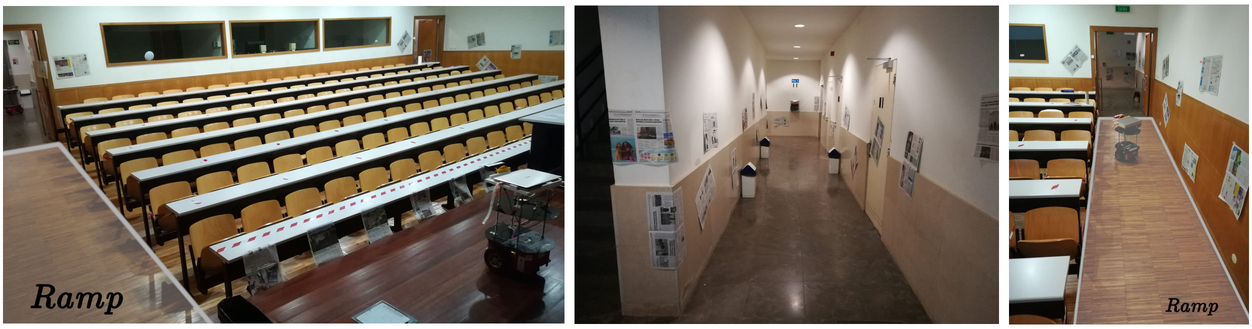


Figure 5.1: *ISRsea* scenario.



(a) Interior of the *A3 Auditorium*.

(b) Back corridor.

(c) Side ramp.

Figure 5.2: *A3 Auditorium* scenario.

has the size of 13 x 15 meters and 3.5 meters of height.

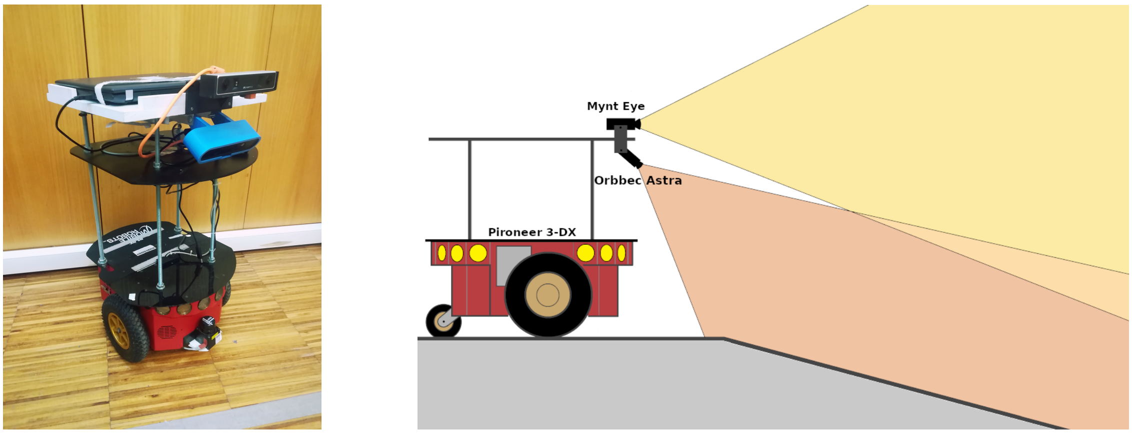
Since the robots do not have the capability to climb stairs, the stage area of the auditorium (viewpoint from Figure 5.2a) is not considered for the exploration mission.

Some newspaper sheets were posted to cover placards (obstacles) and walls in order to suppress the lack of textures and help the depth recognition from the sensors (stereo camera sensor).

Robot Configuration

Three units of the Pioneer 3-DX mobile robot were used, likewise in the simulation tests. The Pioneer robots were equipped with two cameras (Figure 5.3a), the Orbbec Astra S and the Mynt Eye S1030 (Section 3.3). The performance of the Intel RealSense D435i has been analysed and is discussed later on, with this being replaced by the Mynt Eye camera. Besides stereo limitations that were counterbalanced by the structured light RGB-D camera sensor, the usage of two cameras allowed to increase the FoV, providing more information about the environment at each instance.

The Astra camera was positioned looking down, so that when the robot approaches different surface levels, such as descending ramps (Figure 5.3b), the robot is able to perceive it



(a) Robotic System.

(b) Illustration of the FoV of both sensors operate.

Figure 5.3: The Pioneer 3-DX mobile robot platform.

in the FoV.

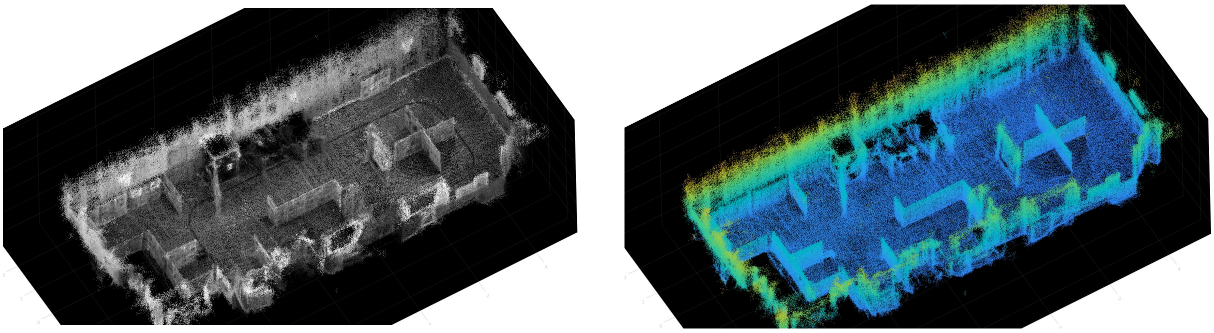
Another important aspect defined in the robot configuration was the speed cruise of the mobile platform. This was set to 30 cm/s so that when stopping, the robot does not fall forward due to its elevated structure. This is a relevant precaution when operating robots like the Pioneers in complex and irregular terrains. Also, the pitch angle interval to be used in the traversable map partitioning was set to [-30°; 30°]. Any surface outside of this interval is considered as obstacle. The existent ramps in the *A3 Auditorium* have a slope of 5°, so the interval is enough to enable the robot to perceive them.

SLAM and Localization-only mode

The purpose of these experimental tests is to prove that robots are capable of exploring an unknown environment autonomously while sharing information between them and avoiding conflicts such as collisions or nearby targets selection. Since the focus of this work is the exploration itself and not robot localization, a global map of the environment was provided to all robots so that they start with the same reference frame and self-localize in it. This way, robots have their local pose in the same common frame of reference. This map is only used for localization, while the scene to be mapped and explored is still unknown for the purpose of the experiments.

The Mynt Eye camera was used to previously map the scene and saved in form of a database to be loaded by all the robots. This mapping process consists of navigating with the camera through the scene, trying to capture all possible features[1]. Examples of the

[1] Video of mapping the environment using the Mynt Eye S1030 available: `https://www.youtube.com/watch?v=PQXuCM05AsE&list`



(a) Monochromatic map.

(b) Colormap aligned in zz axis, from bottom (blue) to top (yellow).

Figure 5.4: 3D Maps of the ISRsea environment.

reconstructed scene are presented in Figure 5.4.

5.1.1 Odometry calibration

The Pioneer 3-DX is a differential mobile which means that wheels can be controlled independently. The kinematic model of a differential mobile robot is given by the following relations:

$$\begin{bmatrix} x(k+1) \\ y(k+1) \end{bmatrix} = \begin{cases} x(k) + D(k) \cdot cos(\theta(k+1)) \\ y(k) + D(k) \cdot sin(\theta(k+1)) \end{cases} \tag{5.1}$$

$$\theta(k+1) = \theta(k) + \frac{D_l(k+1) - D_r(k+1)}{L} \tag{5.2}$$

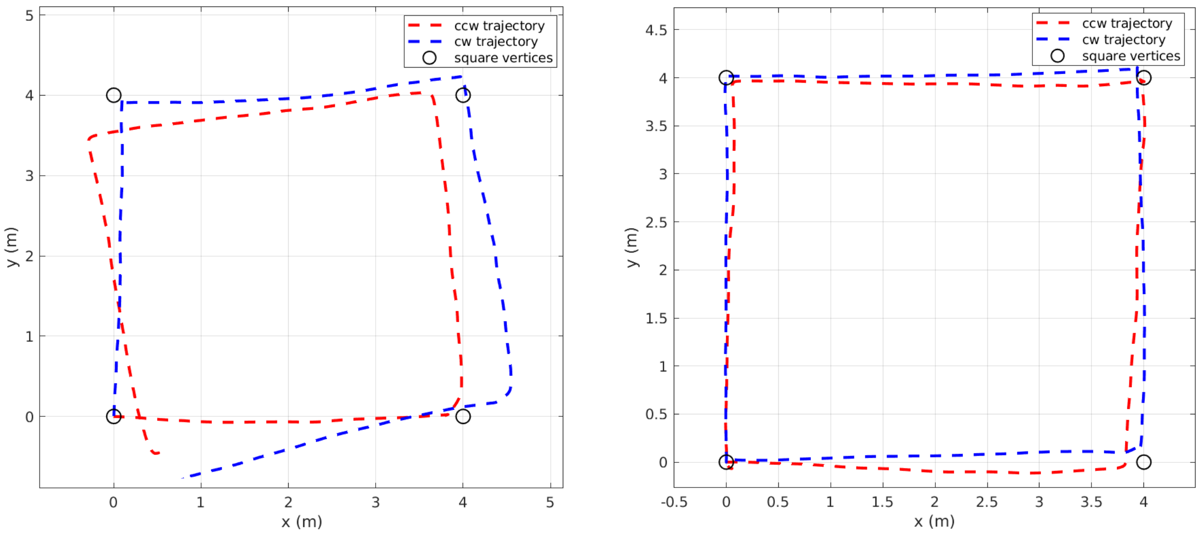
with D,

$$D = \frac{D_l + D_r}{2} \tag{5.3}$$

and D_l and D_r,

$$D_l = \frac{2\pi N_l}{E_l} R_{el}, \qquad D_r = \frac{2\pi N_r}{E_r} R_{er} \tag{5.4}$$

Where $x(k)$ and $y(k)$ are the coordinates of the center of the robot; $D(k)$ the distance travelled by the robot; $\theta(k)$ angle between the robot and x-axis; D_l and D_r distance travelled by left and right wheel, respectively; N_l and N_r number of ticks counted by the left and right encoder's wheel, respectively; R_{el} and R_{er} radius of the wheel; L, the distance between wheels and E_l and E_r number of ticks of the encoders by revolution.



(a) CW and CCW trajectories before odometry calibration.

(b) CW and CCW trajectories after odometry calibration.

Figure 5.5: Correcting odometry using the UMBMark method [15].

After some tests performed, one of the robots would become lost in particular zones of the environment. Even if, after some time, it managed to relocate, this could result in some uncertainties to the constructed map. These events seemed to happen with a systematic pattern, which could indicate a systematic wheel odometry error of the Pioneer.

Using the UMBMark method described by Johann Borenstein and Liqiang Feng [15], it was possible to measure and correct this odometry error. The test consisted in a robot performing a square trajectory and analysing the location of the final point, which should be close to the initial point. This trajectory should be performed clockwise (CW) and counter-clockwise (CCW). The difference between the final point (0, 0) and the initial point (0, 0) should be as small as possible. Graphs in Figure 5.5 present the CW and CCW results with uncalibrated odometry (Figure 5.5a) and with calibrated odometry (Figure 5.5b). The odometry calibration has consisted in tuning the *DriftFactor* parameter [46] - responsible for adding or subtracting increments in the left wheel encoder's ticks to correct tire circumference differences and consequent translation and rotation drift. This parameter will directly interfere in the N_l.

Before the odometry calibration, the error estimation in the square trajectory was about 0.894 meters, between the final and initial points of the CW and CCW trajectories (Figure 5.5a). After odometry calibration, the error was significantly reduced to 0.083 meters (Figure 5.5b).

Visual Odometry	Pros	– Fast Motion Estimation.
	Cons	– Small certainty on environments with lack of textures. – Leads to drifts on the created map.
Visual + Wheel Odometry	Pros	– Higher certainty.
	Cons	– If lost, the localization system fed by combined odometry may not re-locate again. – Huge drifts, especially during rotation. – May result in map inconsistencies, when the wheels slide.

Table 5.1: Comparative results when using visual and visual + wheel odometry.

5.1.2 Visual Odometry vs Wheel odometry

Visual odometry relies on estimating motion from a stereo camera, or RGB-D depth images, to compute the odometry of the robotic system [47]. In wheeled odometry, the pose is estimated by the motion of rotary encoders mechanically coupled with the robot's wheels. In this project, odometry would be estimated using a combination of both, so that visual odometry may provide a precise motion by observing a sequence of images of the environment and the rotary encoders could provide the motion of the robot when a low number of features are detected. Combining them, each one may suppress the weaknesses of the other.

Due to the odometry errors presented in the previous section, tests using only visual and using the combination of both were performed to understand which would be the most suitable option.

Relevant observations are summarized in Table (5.1). These observations relies on visual results from using between visual odometry and combined visual and wheel odometry.

When adding the wheel odometry, this may compromise the estimation on 3D surfaces, delaying that estimation when changing from one surface to another which, for the proposed robotic system, does not have much impact. In fact, both have shown to be very effective for the purpose of this work. Since the map is previously provided to each robot, if this is a high quality map, the motion of the robotic system will be easier to estimate, especially by the camera sensor, which is crucial to estimate motion on irregular terrains, since wheel odometry would only estimate motion in a 2D plane (x, y, θ).

The 3D pose estimation in the environment is computed using combined visual and wheel odometry. The RTAB-Map ROS node subscribes an RGB-D image from the stereo camera and the wheeled odometry from the robot which then updates the pose estimation into a transformation (TF) as /map –> /odom (see Figure 5.6). Considering such inputs, the pose

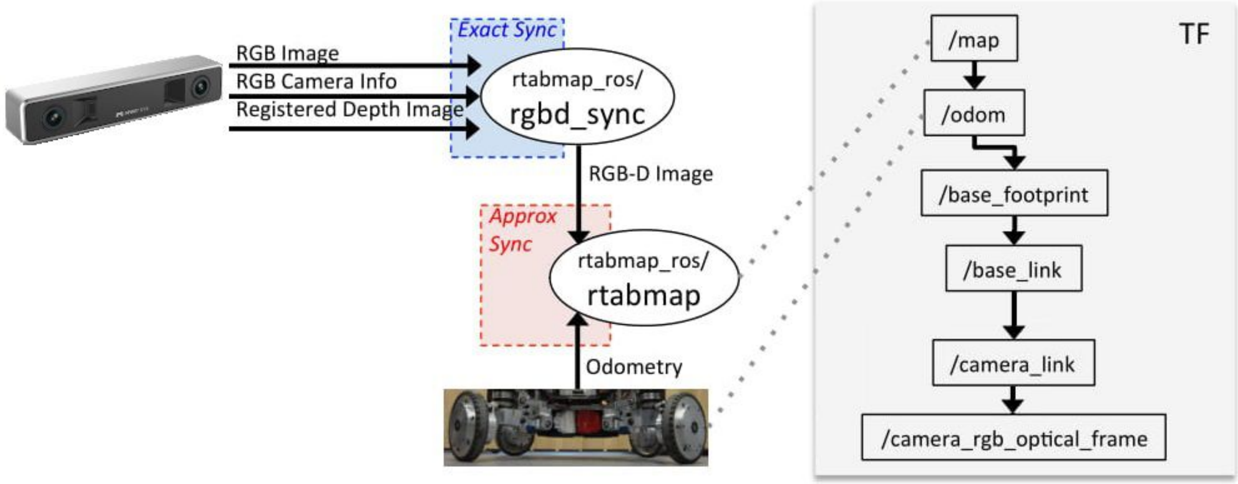
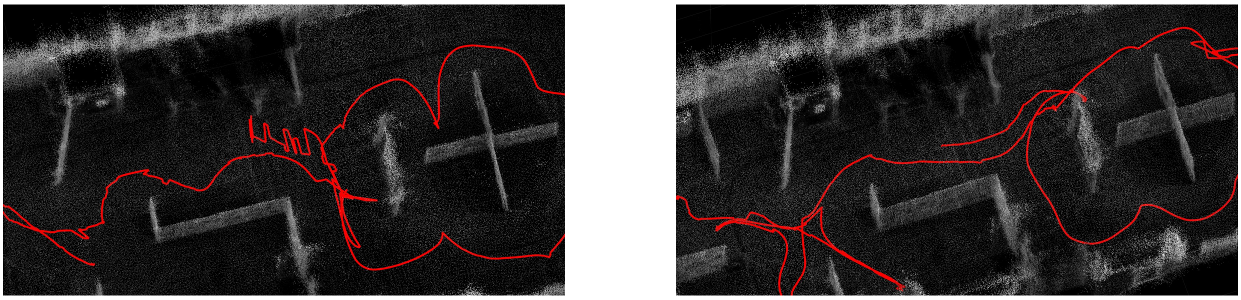


Figure 5.6: Synchronization example of a stereo camera and odometry. In this case, odometry is computed through wheel encoders. Camera messages are synchronized together using rgbd_sync ROS node before synchronizing the resulting RGB-D image message with the other sensors (which can have different publishing rates). On the right is an example of the resulting TF tree for this sensor configuration, with transforms linked by a dotted line to corresponding publishing ROS nodes. Reproduced from [16].



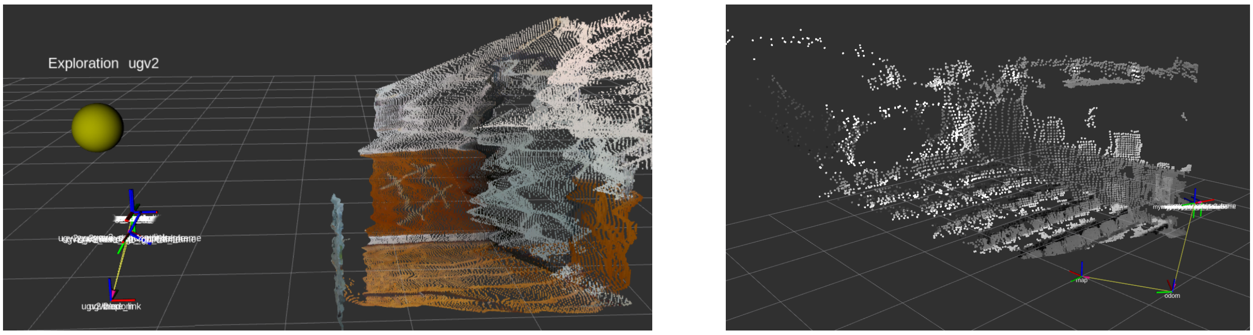
(a) Visual odometry.

(b) Combined visual and wheel odometry.

Figure 5.7: Examples of path performed by robots using combined visual and wheel odometry.

estimation is performed by the RTAB-Map ROS package. Detailed information about this package can be found in [16].

The main difference in the pose estimation is exemplified in Figure 5.7. Using only visual odometry, the pose estimation tends to fluctuate between hypotheses unlike when using combined visual and wheeled odometry which utilizes an extra sensor source for motion estimation and allows the system to estimate the robot's pose more accurately. The fact that the Pioneer 3-DX has a high accuracy wheel odometry in short-time intervals also helps to maintain the pose estimation throughout the mission.



(a) Intel Realsense D435i.

(b) Mynt Eye S1030.

Figure 5.8: Illustrative point clouds of Realsense and Mynt Eye.

5.1.3 Mynt Eye S1030 vs Intel Realsense D435i

The Realsense D435i camera was integrated into the robotic system in order to check its feasibility on exploration missions. Yet, it was soon noticed that this camera would not be effective in the tests. In Table 3.1, presented in Section 3.3, the Realsense camera presents some weaknesses compared to the Mynt Eye, because the FoV and range are the most relevant technical specification for exploration.

Besides those specifications, the Realsense has shown a huge distortion, losing depth precision as the range increases. This may occlude some passages, such as corners (Figure 5.8a), preventing the robot from going through those passages and leading it to finish the exploration too soon.

Besides having a higher range and FoV than the Realsense, the Mynt Eye camera does not present significant distortion and it is possible to extract a clear and defined point cloud (Figure 5.8b). Whereas the Mynt Eye camera happens to gain some distortion after 10 meters, Realsense happens to gain after 4 meters, a very noisy distortion. This camera has the downside that, as depth estimation relies on stereo, some features may be projected beyond the walls, if these features are lacking. The fact that Mynt Eye has a higher distance between the two lens (denoted as baseline) than the Realsense helps to perceive more distant features when triangulation of images is applied. While Mynt Eye's baseline is 120 mm, Realsene's baseline is 50mm.

Summing up, the Mynt Eye turns out to be a suitable option for exploration missions since its wide FoV allows to acquire greater information about the environment at once. Additionally, its maximum range (18 meters) and minimum distortion at those higher distances, enable clear and defined depth estimation. Although the Realsense camera may be very useful for robotic applications, it is not the most appropriate option when large areas

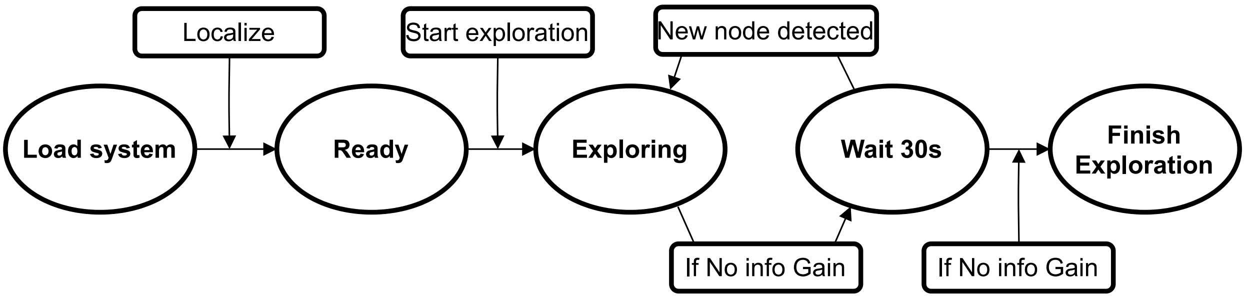


Figure 5.9: Workflow of exploration mission on each robot.

need to be scanned, due to the significant distortion.

Real system configuration

Considering the ablation study described in the previous chapter, the parameters optimization, were taken into account for the real-world tests. Therefore, the exploration step was set to 2 meters. The maximum range was set to 7 meters, although the proposed cameras comprehend a considerable range, especially the Mynt Eye which is the one used to validate the system. This range should be moderated in order to not overload the computation process of each machine; and the extension range was set to 0.5 meters.

The exploration mission on each robot was designed to work as follows: all robots initialize the system and localize in the environment; when the exploration planner is ready, a start manual command is given to all the robots and then they start to explore; when exploring, thy periodically check their exploration status (*"planning"*, *"forwarding"*, etc.); if a robot has the status of *"No Information Gain"* it has likely explored everything, so it waits 30 seconds until checking the status again, if the status remains the same, the mission ends for that robot, if not, it checks again every 30 seconds (see Figure 5.9).

Note that, while exploring, robots are constantly exchanging messages with their teammates, planning targets, avoiding conflicts, moving around the environment, etc.

5.2 Experimental Tests

Intensive experimental tests took place across five months to test cameras, odometry, network configuration, SLAM implementation, tune safety and trajectory parameters, etc. In order to store data of the system in run time, rosbag[2] recording was running for each test, on each robot of the exploration mission performed, storing all necessary data/information to

[2] Rosbag: ROS package used to record datasets in real time and playback offline.

be further analysed and discussed. For this dissertation, all the tests presented later on have a dataset file already analysed. In this document, we present 23 real-world successful tests. For these tests, a single dataset file can range from 500 Megabytes up to 2.3 Gygabytes. In total, 61.2 Gigabytes of datasets were generated for these final tests, which are stored in a specific Desktop of the MRL-ISR so that future researchers may appeal to them for further analysis and comparisons.

Evaluation metrics

In order to validate the approach tested with real robots, several metrics were used. This validation relies on the exploration mission time (t_{expl}), as a function of the number of robots being used; the number of occupied cells explored by each robot (occ_{expl}); the time that each robot takes to explore 95% of the environment ($t_{expl_{95}}$); the time that robots spend between the instant of time when 95% of the scenario has already been explored until the instant of time when they finally finish the mission (t_{end}); the distance travelled by each robot (d_{trav}); the ability of robots to avoid conflicts, *i.e.* if the robots do not collide with others and if they plan when a conflict occurs; and the information shared, measured as the number of points processed and integrated in each robot's map.

Denote that, while in a simulated environment the explored map may have the approximate result, in the real world environment, this is not likely to happen and this map may have a more fluctuating value. For that reason, (occ_{expl}) it is presented in form of a percentage when multi-robot missions are applied and discarded for the single-robot missions.

5.2.1 Single-Robot Experiments

First, single-robot tests were performed, mostly because of the difference between the three computers that would be used on top of the robots in the tests. Two of them were provided by the MRL-ISR and another was a personal laptop. Specifications of these machines are presented in Table 5.2. These laptops are well skilled concerning performance, which is fundamental for the tests, since it is a relatively heavy system that is running in these machines.

These tests were performed with the same robot, starting in approximately the same pose and the only differentiating factor was the usage of those three machines. Tests were performed three times with each machine.

Each machine has its specifications that directly reflect on the exploration performance.

	CPU	**CPU freq**	**GPU**	**RAM**
PC 1	i7-6500U	2.5 GHz	Intel HD Graphics 520	16 GB
PC 2	i7-6500U	2.5 GHz	Intel HD Graphics 520	16 GB
PC 3	i7-7700HQ	2.8 GHz	Nvidia 1050 GTX	16 GB

Table 5.2: Specifications of each machine used in exploration tests.

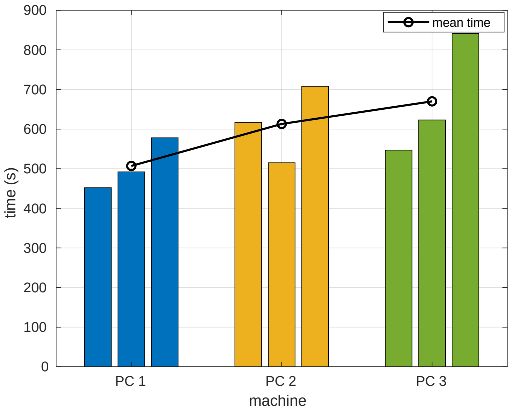
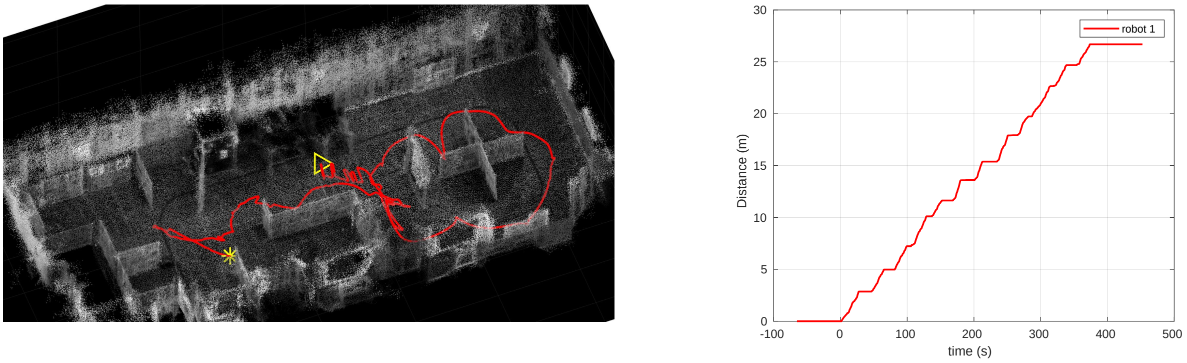


Figure 5.10: Bar graph with each test on each machine.

	PC 1	**PC 2**	**PC 3**
min time (s)	452	515	547
max time (s)	578	708	841
mean time (s)	507	613	670

Table 5.3: Table with overall time results performed on each machine.



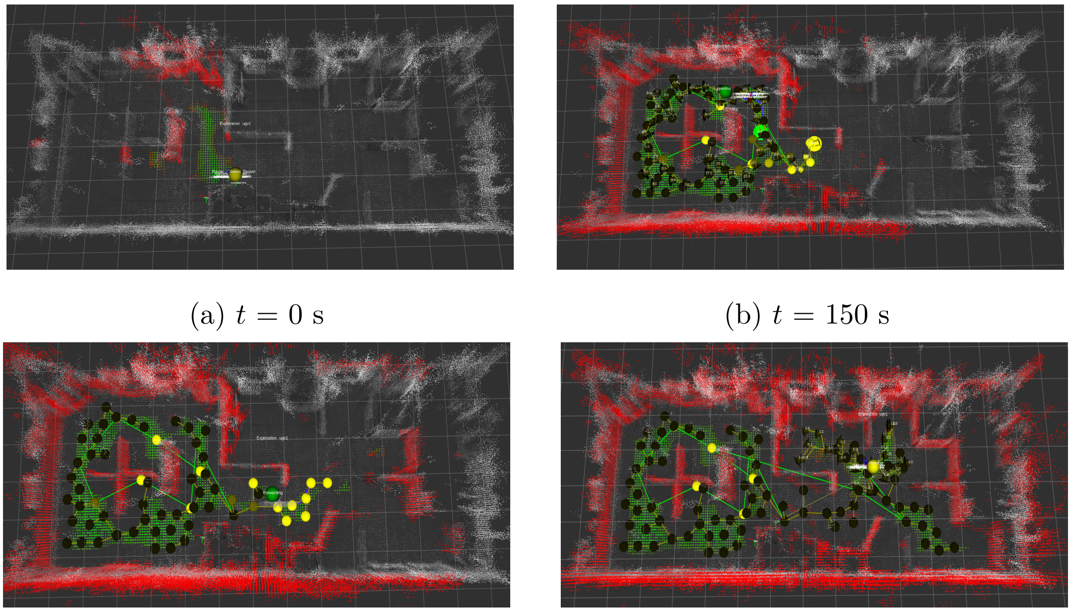
(a) Robot's path in the environment - Initial and final positions marked with ▷ and $*$, respectively.

(b) Distance travelled during the mission.

Figure 5.11: Path followed and distance travelled by the robot during the exploration mission.

Although the *PC3* is a more powerful computer than the others, it was observed that its hardware entered in power save mode when using battery only, slowing down the computation resources needed, which increased the time needed to explore. Regardless of the machine used, and evaluating all the tests, it is assumed that each exploration mission with a single-robot system may occur with a mean time of 597 seconds (detailed values presented in Table 8.1, page 83).

In Figure 5.11a, it is possible to see the trajectory performed by the robot through the



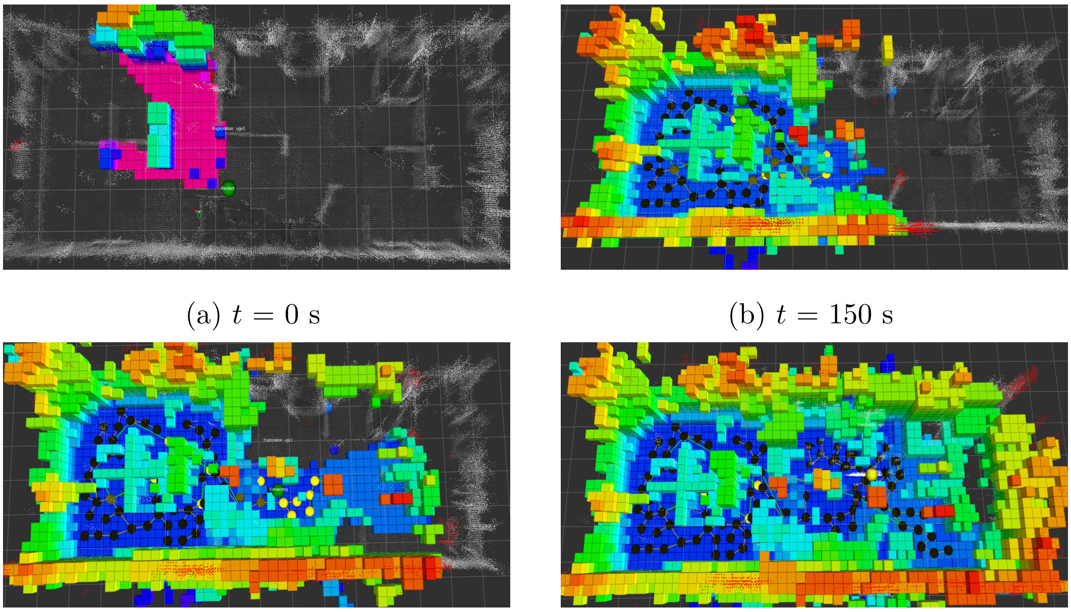
(a) $t = 0$ s

(b) $t = 150$ s

(c) $t = 300$ s

(d) $t = 450$ s

Figure 5.12: Segmented point cloud during the exploration mission. Obstacles in red and traversable area in green.



(a) $t = 0$ s

(b) $t = 150$ s

(c) $t = 300$ s

(d) $t = 450$ s

Figure 5.13: Occupied cells of the volumetric map during the exploration mission in the form of octomaps.

scenario, from the start to the end point. Since visual odometry was being used for these tests, an erratic oscillation was observable in the beginning of the trajectory due to the pose estimation uncertainty. After the robot moved for a while, the pose estimation was adjusted and the path was already well defined. Due to the RH-NBV strategy, the robot did not need

Nr of robots	$t_{expl_{mean}}$ (s)	$t_{95_{mean}}$ (s)	occ_{expl} (%)	$t_{end_{mean}}$ (s)	$d_{trav_{robot}}$ (m)	$S_{t_{expl}}$	$S_{t_{95}}$	$S_{t_{end}}$
1	597	431	-	166	34	1	1	1
2	582	301	**98**	137	21	0.51	0.72	0.61
3	**414**	**216**	85	**102**	**17**	0.48	0.67	0.54

Table 5.4: Summary of the performed tests with the results with superior performance highlighted. Speedup values shown at the right-side of the table.

to move till the end of each corridor to explore the entire scene.

Figure 5.11b illustrates the performance along the trajectory, determining when and for how long the robot moves and stops. When the robot stops, it has reached the goal and it is planning a new goal to move to. Once the exploration agent does not find a new target to visit, the robot finishes the exploration mission.

Figures 5.12 and 5.13 present the sequence of the map generated by the robot in four stages, from the start of exploration ($t = 0$ s) until the end of it ($t = 450$ s).

5.2.2 Multi-Robot Experiments

Using the same environment (Figure 5.1), multi-robot exploration missions were performed. In order to establish communication between robots running an independent ROS system and enabling them to share ROS topics, the ROS multimaster_fkie package mentioned in Subsection 3.4.1 was used.

Table 5.4 presents the summary of performed tests with one, two and three robots in the team. Detailed tables of each test are presented in Appendix 8.1.1, on page 83. The mean time of an exploration mission ($t_{expl_{mean}}$) was reduced as a new teammate was integrated into the robotic team. Besides that, also the time that a robot took to explore 95% of the environment (t_{95}) was reduced by about 30% as a new robot is added to the team. In fact, t_{95} was reduced by 30.1% with 2 robots and 28.2% when extending from 2 to 3 robots. These two factors and the distance travelled ($d_{trav_{robot}}$) of each robot are proof that the cooperation between robots leads to each robot exploring different areas of the environment, which are not explored yet.

An effective tool to measure the performance of the exploration as more robots are added to the team is the Balch's speedup metric [48], which is expressed as

$$S[i] = \frac{\frac{P[1]}{i}}{P[i]}, \tag{5.5}$$

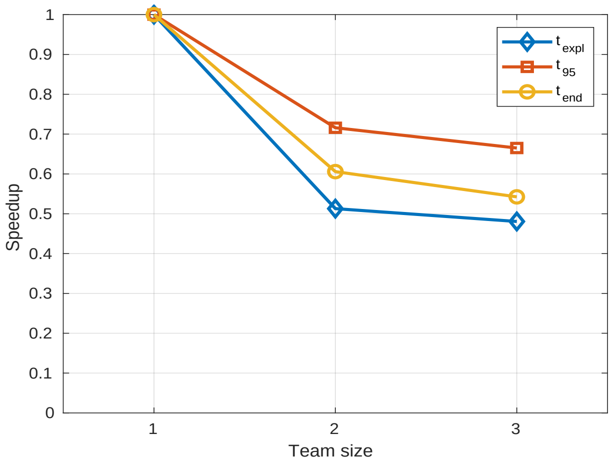


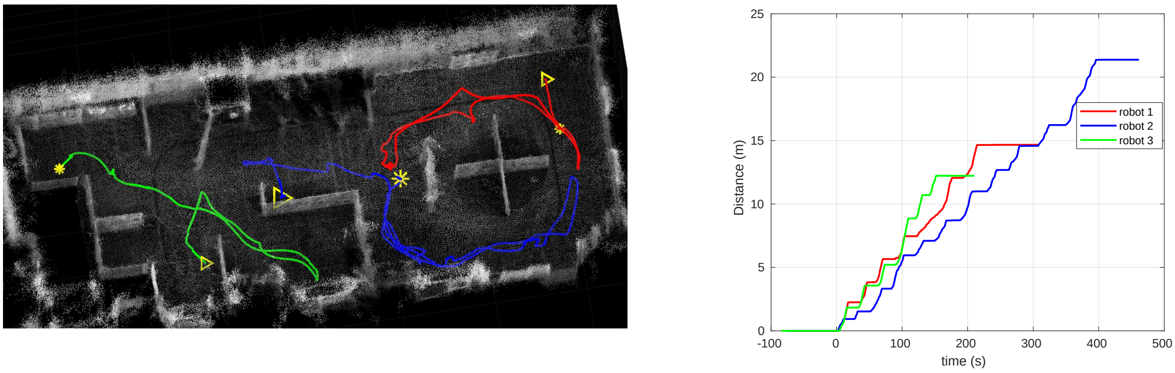
Figure 5.14: Speedup graph of t_{expl}, t_{95} and t_{end} with 1, 2 and 3 robots in the ISRsea scenario.

where $P[i]$ is the performance for i robots. So, if two robots complete the exploration mission exactly twice as fast as one robot, the speedup is 1.0 (higher numbers are better). Table 5.4 presents the results of this measure for the *ISRsea* scenario.

Taking into account the values of the t_{exp}, t_{95} and t_{end} presented in Table 5.4, a speedup graph of the ISRsea scenario is illustrated in Figure 5.14. It is possible to conclude that the system rapidly enters in a sublinear performance ($S[i] < 1$). As a new robot is added to the team, the exploration time will not necessarily be reduced in a linear way. At some point, the performance of exploration will decrease with the addition of more robots in the environment, as they will spend more time managing conflicts compared with the "useful" time spent in exploration itself. This sublinear speedup is commonly observed in multi-robot system deployments. It tends to be more noticeable in more constrained environments where robots are more likely to interfere among themselves, or in larger teams. The time to explore 95% of the environment (t_{95}) and the time that a robot takes to finish the exploration mission (t_{end}) follows a similar trend, in which the t_{95} presents a performance closer to the linear behaviour since the robots share their information about the environment. As robots are added to the team, they will sooner have more complete information about the explored environment, thus obtaining the 95% exploration ratio more quickly.

When analysing Figure 5.15a, it is possible to observe the exploration trajectories performed by each robot which, by further analysing Figures 5.16, 5.17 and 5.18, shows that each robot has integrated information in their local maps that was acquired by others. All the robots begin with their local maps and, as the mission goes, those maps grow with their scans and the information provided by the teammates.

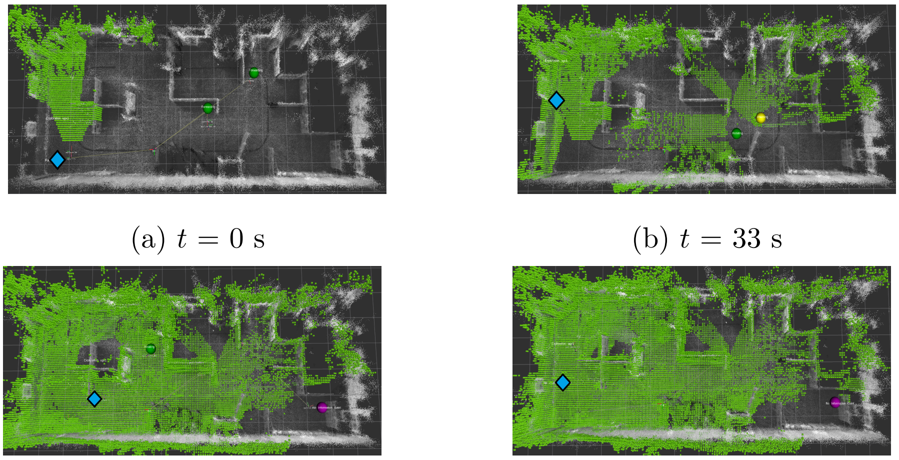
For these tests, a public Wi-Fi network of the Institute of Systems and Robotics (ISR)



(a) Robots' path in the environment - Initial and final positions marked with ▷ and $*$, respectively.

(b) Distance travelled during the mission.

Figure 5.15: Multi-robot exploration with 3 robots.



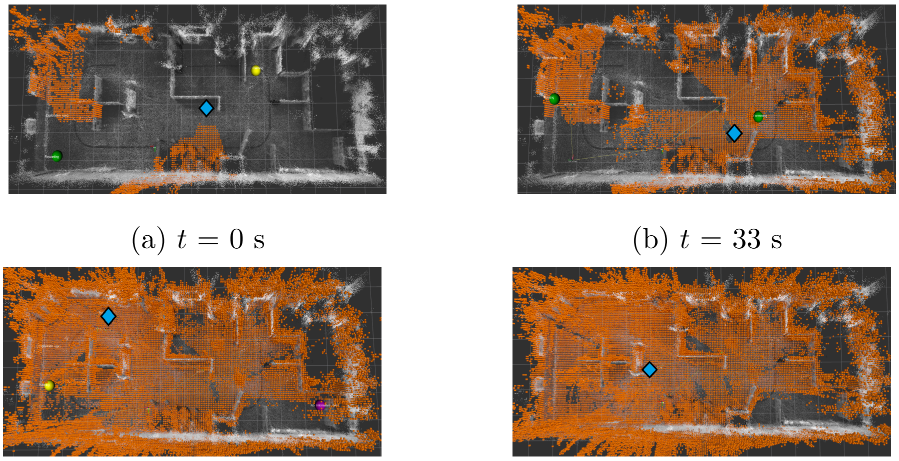
(a) $t = 0$ s

(b) $t = 33$ s

(c) $t_{95} = 177$ s

(d) $t = 308$ s

Figure 5.16: Sequence of the local map of robot 1. the blue marker represents robot 1.



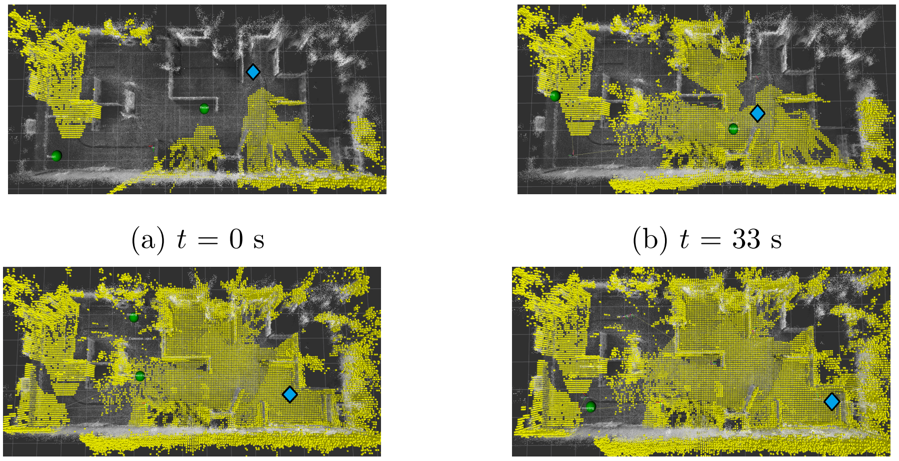
(a) $t = 0$ s

(b) $t = 33$ s

(c) $t_{95} = 237$ s

(d) $t = 461$ s

Figure 5.17: Sequence of the local map of robot 2. The blue marker represents robot 2.



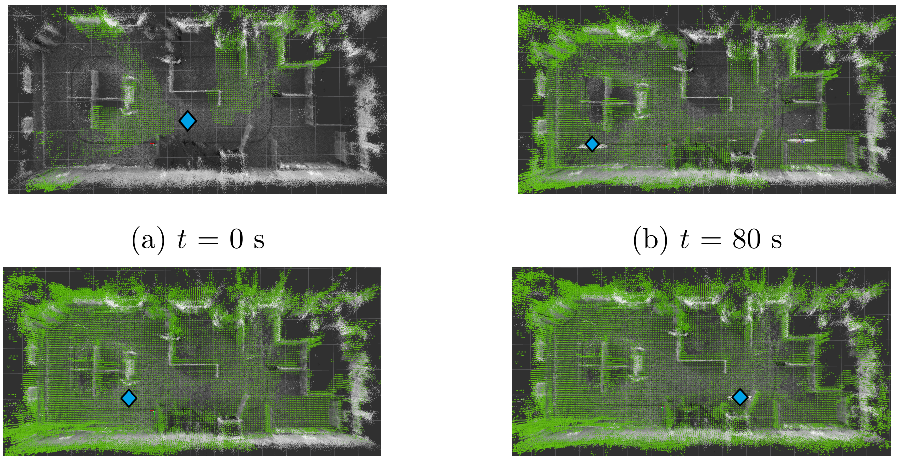
(a) $t = 0$ s

(b) $t = 33$ s

(c) $t_{95} = 140$ s

(d) $t = 210$ s

Figure 5.18: Sequence of the local map of robot 3. The blue marker represents robot 3.



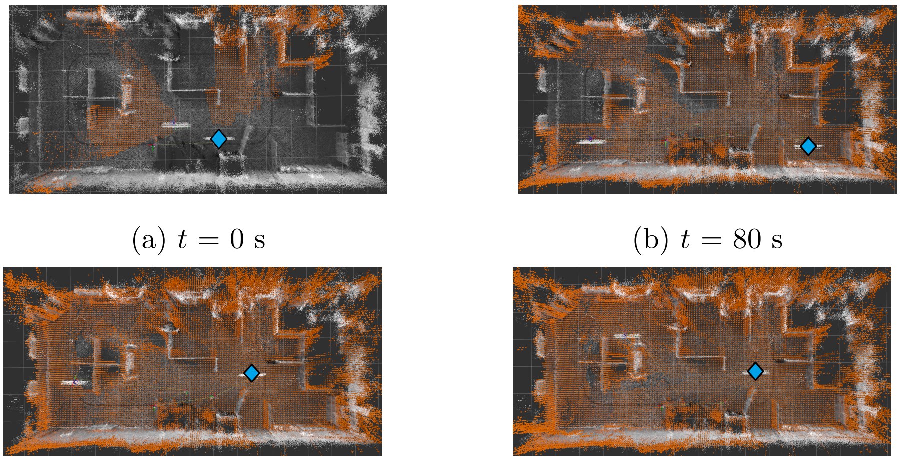
(a) $t = 0$ s

(b) $t = 80$ s

(c) $t_{95} = 415$ s

(d) $t_{expl} = 585$ s

Figure 5.19: Sequence of the local map of robot 1. The blue marker represents robot 1.



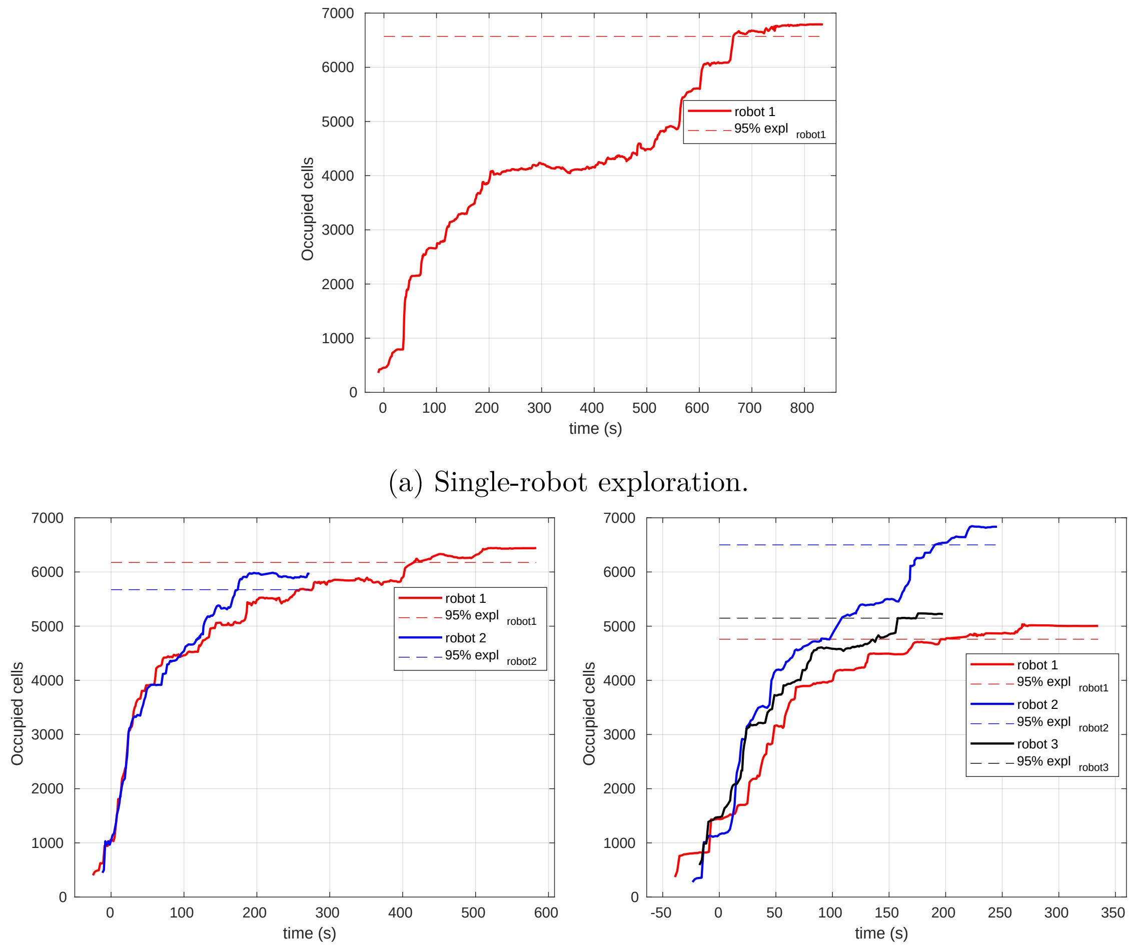
(a) $t = 0$ s

(b) $t = 80$ s

(c) $t_{95} = 173$ s

(d) $t_{expl} = 273$ s

Figure 5.20: Sequence of the local map of robot 2. The blue marker represents robot 2.



(a) Single-robot exploration.

(b) Multi-robot exploration with 2 robots.

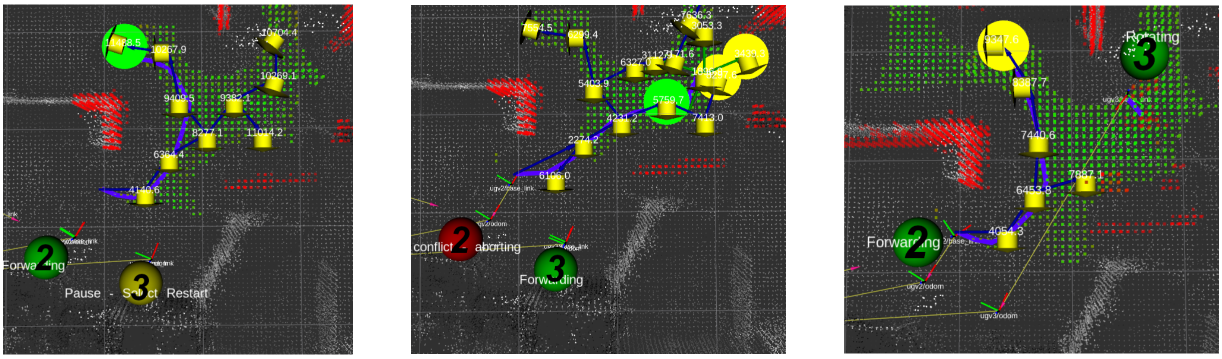
(c) Multi-robot exploration with 3 robots.

Figure 5.21: Number of occ_{expl} of each robot during exploration missions and $t_{expl_{95}}$ of the volumetric map.

was used for inter-robot communication. Due to the network being public and not dedicated, the traffic of the network interfered with the communication and did not allow to share all the data between teammates. Yet, this did not have a critical impact when the robotic team consisted of just 2 robots (see Figures 5.19 and 5.20).

Besides the time, the main difference between single and multi-robot exploration relies on how new information is acquired at each instance during the mission. With each robot exploring a distinct area, they will rapidly acquire significant portions of the scenario, which will reduce the time to reach the 95% of the environment explored ($t_{expl_{95}}$). Furthermore, as they explore distinct regions, each robot will have to explore their surroundings, so any node that can contain new information is quickly achieved, reducing by far, the time that robots need to finish the exploration mission.

As expected, not all the robots finish the exploration at the same time. After a robot explores its surroundings and no new information is detected, the mission ends for the robot.



(a) Robot 2 selects target goal.

(b) Robot 3 selects target goal and robot 2 detects a conflict.

(c) Robot 2 selects new target goal.

Figure 5.22: Illustration of management of conflicts. Monochromatic point cloud represents the map previously mapped; points in green represent the traversable region; red points represent the obstacles; nodes are expanded from the roots of each robot, in which the best node of the tree and the next planned node are presented in yellow and green, respectively; The blue line illustrates the path that robot will follow to reach the selected node.

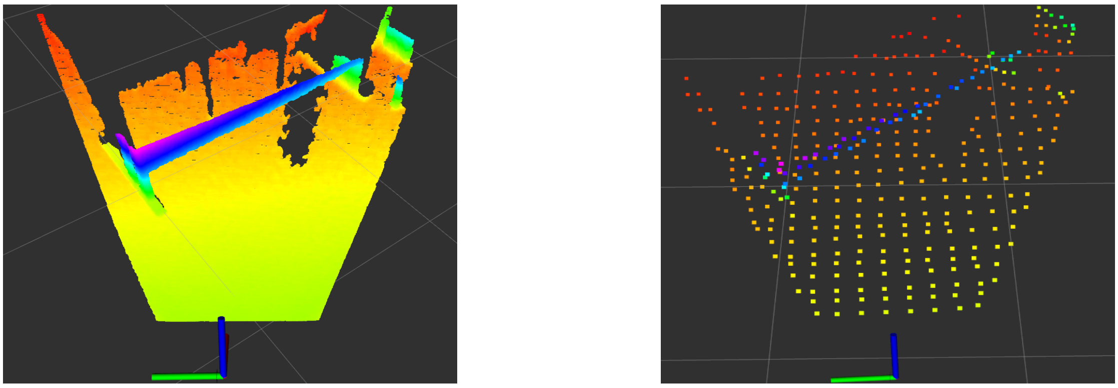
Conflict Avoidance

Robots successfully manage to select their target goals and followed path in order to avoid conflicts and interference with their teammates. The information shared endows each robot to “know” exactly where the teammates are, to where they are moving and the path they will follow. Figure 5.22a shows an example of two robots side by side represented in yellow and green in the left side of the figure, at the beginning of the mission, promoting a metric conflict. After the conflict is detected (Figure 5.22b), robot 2 stops and manages to assign a different target in order to avoid collision and proceed with the exploration (see Figure 5.22c). This conflict happens because both robots perceive the same portion of the environment and the local maps overlap.

Adding another robot will increase the probability of conflicts, especially if the size of the scenario is restricted to many more robots. Having more conflicts, instead of decreasing the time, the robots will spend more time managing the conflicts rather than exploring. In order to add a new robot, specifications of size and structure of the scenario should be considered.

5.2.3 Experiments in a More Complex 3D Environment

The *A3 Auditorium* scenario comprehends the inside of the Auditorium and a back corridor that connects the two side ramps (see Figure 5.24a). The robot is able to plan its movement



(a) Original point cloud.

(b) Decimated point cloud.

Figure 5.23: Illustration of original (a) and decimated point cloud (b).

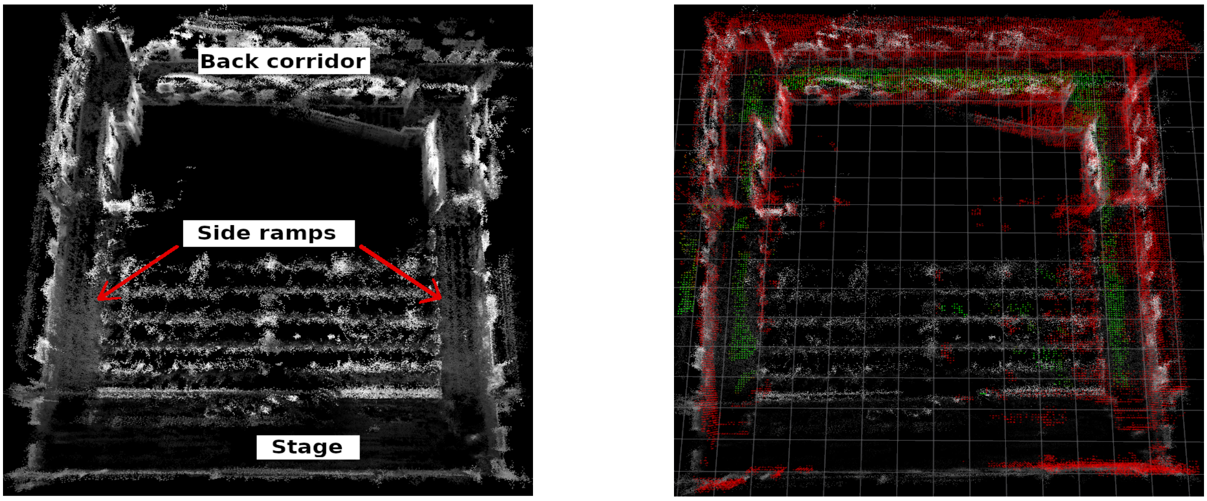
through the side ramps and a back corridor. This scenario presents a very difficult challenge, with several limitations on the network usage, such as bandwidth and signal quality and lack of textures on the walls. For that reason, a dedicated network was deployed for these tests. Yet, because of the lack of signal quality outside of the *A3 Auditorium* (back corridor), the available bandwidth would still be an issue.

In order to suppress this limitation and spare bandwidth, to the point cloud of observations was subject to a decimation process, thus reducing its size and the bandwidth needed. This decimation had a factor of 10. Therefore, a point cloud with a resolution of 640x480 pixels turns out to be converted into a point cloud with a resolution about 64x48 pixels (see Figure 5.23). In real-world observations, the final decimated point cloud turns out to maintain acceptable density of depth information. Even if some small details are missing in the observations, these are acquired as soon as the robot moves.

Figure 5.24b shows that traversable region is successfully projected along the side ramps, all through the corridor. Therefore, robots are able to explore, plan and navigate through an irregular terrain.

Figure 5.25a illustrates the trajectory followed by two robots in the scenario and, next to it, we present the graph of movement performed by them (Figure 5.25b). As expected, each robot has the ability to explore its surroundings, once robot 2 does not find any new target to explore, it finishes the mission.

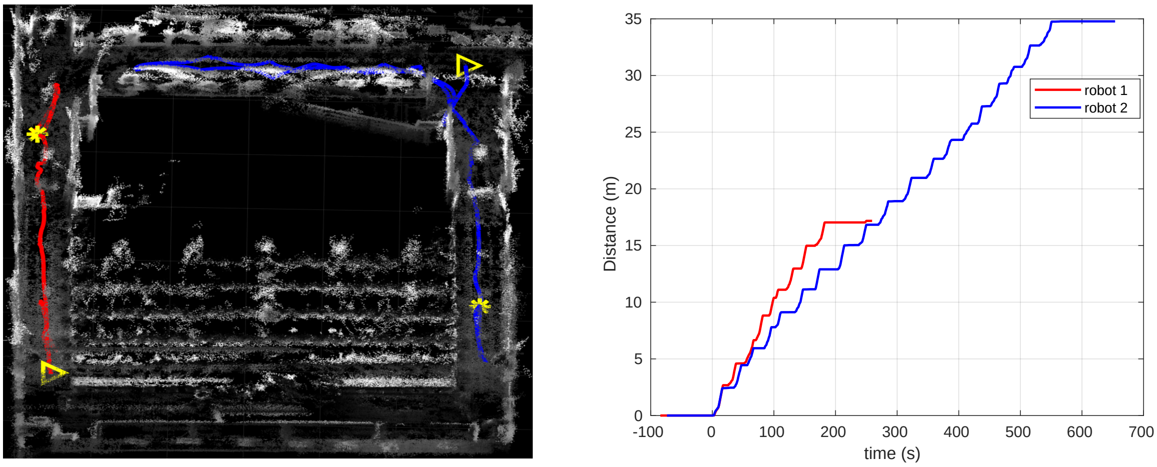
Figure 5.28 shows how occupied cells are progressively integrated in the volumetric map of each robot along the mission. It is possible to see that the number of occupied cells being integrated into the robots follows the same behaviour, so there is a high probability that robots are correctly sharing their maps between them. Also, as soon as robot 2 finishes the



(a) Descriptive illustration of the *A3 Auditorium*.

(b) Explored map in the *A3 Auditorium*.

Figure 5.24: Illustrations of the *A3 Auditorium* scenario.



(a) Robot's path in the environment - Initial and final positions marked with $\triangleright$ and $*$, respectively.

(b) Distance travelled during the mission.

Figure 5.25: Multi-robot exploration with 2 robots in the *A3 Auditorium* scenario.

exploration mission, the amount of occupied cells that are being integrated in the local map of robot 1 softens immediately.

A third robot was added to the exploration team. Each one had to be spaced around the scenario so that they could cover a larger area.

Figure 5.27a presents the paths performed by each robot when using three robots in the *A3 Auditorium* scenario. The distance travelled by each robot is presented in Figure 5.27b.

Due to the geometry of the environment, the robots may not explore further regions (where teammates already explored), therefore they tend to perform a more erratic be-

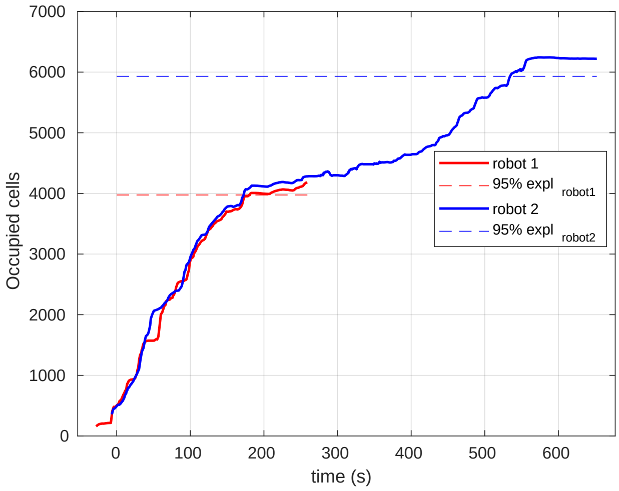
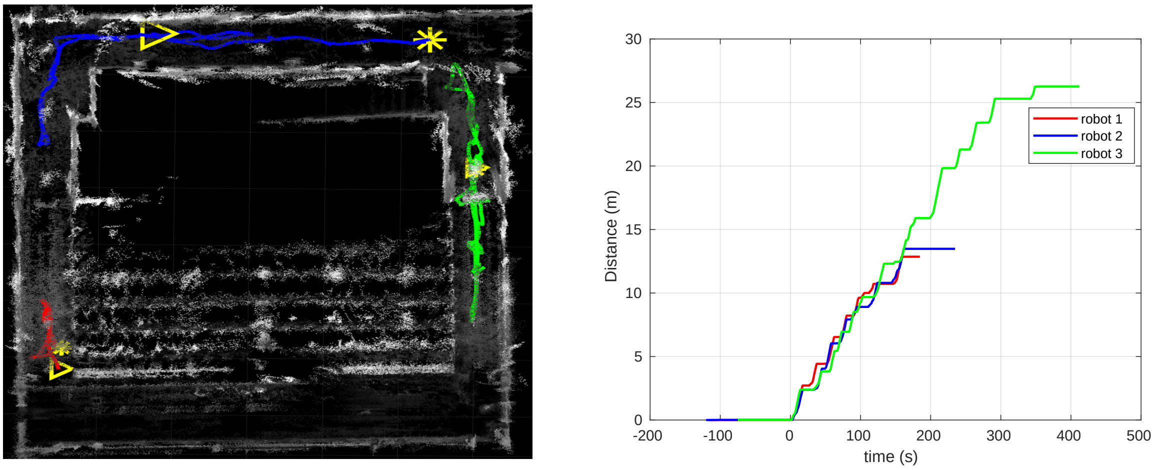


Figure 5.26: Number of occ_{expl} of each robot during exploration missions and $t_{expl_{95}}$ of the local map.



(a) Robots' path in the environment - Initial and final positions marked with $\triangleright$ and $*$, respectively.

(b) Distance travelled during the mission.

Figure 5.27: Multi-robot exploration with 3 robots in the *A3 Auditorium* scenario.

haviour on assigning the nodes to visit. This increases the estimation error on wheel odometry because robots execute pure rotations to change directions, which degrade their pose estimation.

Having localization estimation errors have caused some misalignment when mapping the scene, which in turn has brought problems for other robots such as occluding passages. Although a robot could get an occluded passage, the others could remain exploring, resulting in a total exploration of the *A3 Auditorium* (see Figure 5.27a). Also, as soon as the occluded robot received that one passage was no longer available (due to the misalignment), it immediately changed its trajectory to avoid hitting the obstacle.

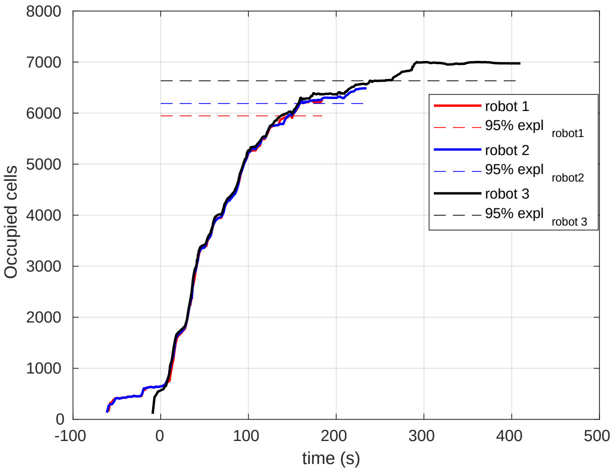
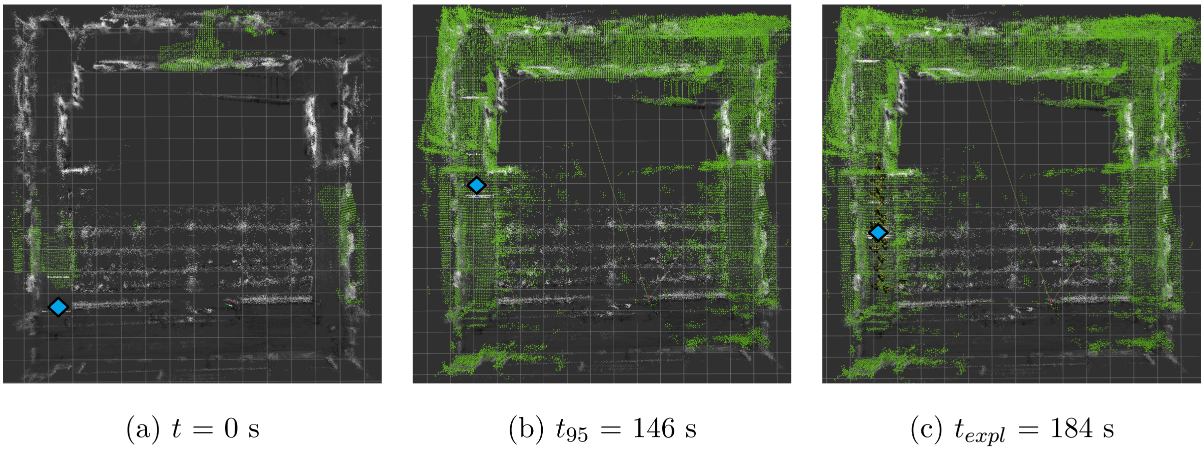


Figure 5.28: Number of occ_{expl} of each robot during exploration missions and $t_{expl_{95}}$ of the volumetric map in the *A3 Auditorium*.



(a) $t = 0$ s

(b) $t_{95} = 146$ s

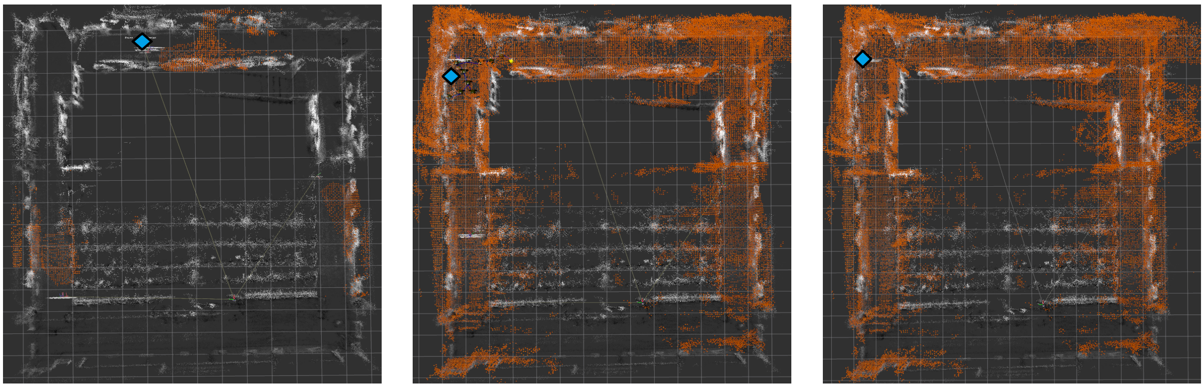
(c) $t_{expl} = 184$ s

Figure 5.29: Sequence of the local map of robot 1. The blue marker represents robot 1.

These errors may happen in environments where the visual sensors have lack of geometrical features. In fact, the addition of newspaper sheets and other objects to complement the scenario, helped to improve the localization of the robot. Yet, did not totally solved the problem of lack of features.

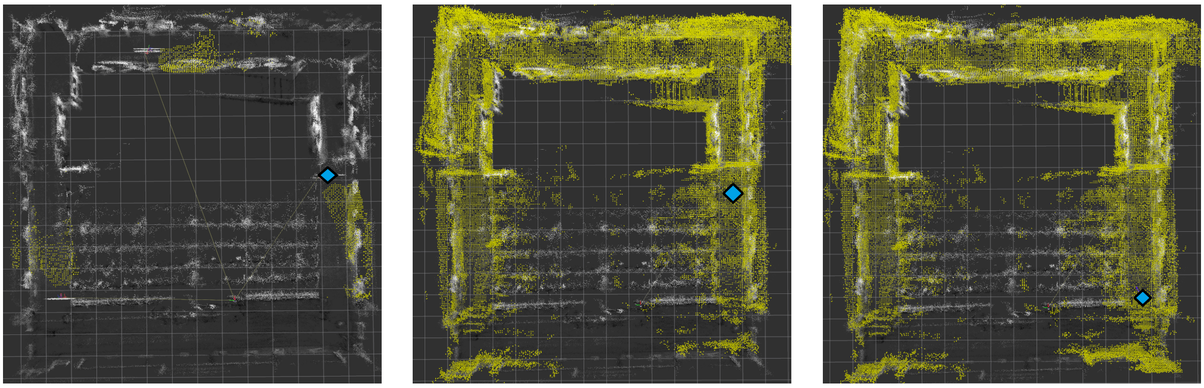
Besides the initial communication issues, once they have establish connection, the information was successfully shared, as it can be verified in Figure 5.28. All three robots have the same amount of points, with no data losses. To complement this information, Figures 5.29, 5.30 and 5.31 demonstrates the sequence of how robots' local map grows along the time.

As can be seen in the previous figures, the upper left corner of the *A3 Auditorium* environment resulted to be defected by an in-turn rotation of one of the robots (robot 2). Since they have successfully shared their observations with others, all the robots had that same defect in their local maps. Consequently, robot 1 turned out to not move through the passage between the ramp and back corridor because it was obstructed by robot 2 information. Yet,



(a) $t = 0$ s (b) $t_{95} = 158$ s (c) $t_{expl} = 235$ s

Figure 5.30: Sequence of the local map of robot 2. The blue marker represents robot 2.



(a) $t = 0$ s (b) $t_{95} = 238$ s (c) $t_{expl} = 410$ s

Figure 5.31: Sequence of the local map of robot 3. The blue marker represents robot 3.

Nr of robots	$t_{expl_{mean}}$ (s)	$t_{95expl_{mean}}$ (s)	occ_{expl} (%)	$t_{end_{mean}}$ (s)	$d_{trav_{robot}}$
2 robots	**600**	386	**89**	**78**	29
3 robots	691	**311**	88	86	**20**

Table 5.5: Summary of the performed tests - results with superior performance highlighted.

robot 1 has re-planned a new goal target and has maintained in the exploration mission, highlighting the flexibility of the approach even in challenging conditions.

Table 5.5 summarizes the multi-robot tests performed in this scenario with two and three robots. The detailed information of each test is presented in Appendix 8.1.2. When comparing 2 and 3 robots, the $t_{expl_{mean}}$ was not reduced because one of the robots spent more time managing the safe path to go through while acquiring observations and rewriting the passage to it.

Even that the time to finish the exploration has not been reduced, major variable as the time to explore the majority of the scenario ($t_{95expl_{mean}}$) was, leading to an improved

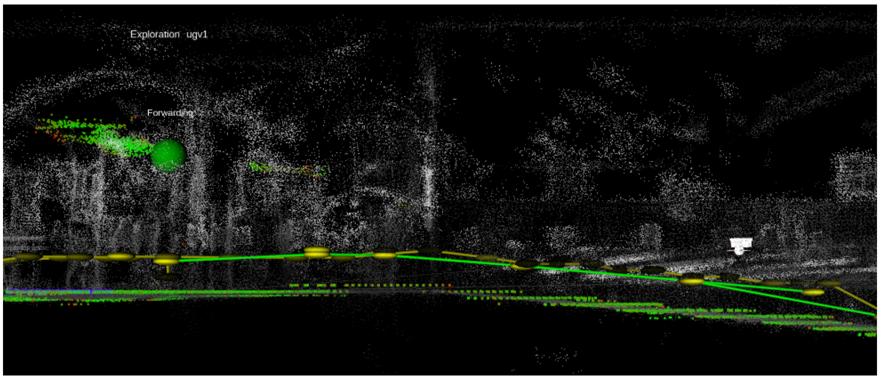


Figure 5.32: Side view of a ramp of the *A3 Auditorium* with traversable region (in green), in which exploration tree nodes are expanded. Point cloud in shades of gray represents the provided map to be used for the localization.

performance of the approach. The distance that every robot travels was also reduced, as expected.

5.3 Summary

Real-world environments introduce several challenges to the robotic system, especially when working with a team of cooperating mobile robots. For that, communication between robots is crucial so that data is not lost. This is taken for granted in simulations, as well as other aspects such as noiseless sensors, reliable and error-free localization, no wheels slippage, absence of the need to charge batteries and purely homogeneous team members.

In a controlled and open environment as ISRsea, network connection has proved to be sufficiently reliable. At that time, a huge amount of data was being shared between robots.

However, in the *A3 Auditorium* scenario, reliability of transmissions and traversability in irregular terrains proved to be the major challenge in real-world testing.

The decimation implemented to the point cloud has significantly reduced the size of the point clouds shared and, consequently, the bandwidth needed at each instance when point clouds are shared with others. The results when this was applied improved significantly, even in a constrained environment as the *A3 Auditorium* is. Whereas the previous point clouds used a bandwidth of 40 Mbps up to 70 Mbps, after the decimation, this was reduced to about 300 Kbps, a reduction of more than 99%.

In the later environment, the mobile robots have successfully managed to navigate through ramps, denoting the ability of the system to perform 3D navigation (see Figure 5.32).

The Localization-only mode technique used has resulted in some collateral errors, defecting the map created, which then resulted in different decisions of the robots. Nevertheless, it has not ruptured the exploration mission itself, causing only small misalignments in the final map of the environment.

Despite the impact of the above-mentioned challenges in the real-world multi-robot exploration, the results in these chapter are in line with the results in the simulated scenarios from Chapter 4, illustrating the feasibility of the multi-robot exploration approach proposed.

Some sketches of the experimental tests were recorded during these tests, which can be seen in the following link: `https://youtu.be/xwCnQtLCORI`.

6 Conclusion and Future Work

The multi-robot exploration research topic still has some challenges to be overcome. In this project, we took a step forward in that direction, dealing with a team of autonomous robots exploring an unknown environment through coordination and information sharing.

This chapter wraps up the main conclusion of the work conducted, summarizing the developed system along the course of this dissertation work, and interesting future research directions.

6.1 Overview

The presented approach in this dissertation was studied and tested in order to present the optimum results, leading to a suitable efficiency when having a team of mobile robots exploring an unknown environment.

The ablation study has shown the potential of the multi-robot coordination approach in order to explore the entire environment. Also, this allowed to study and analyse the implementation of the spatio-temporal filter which has shown to be a significant improvement on how shared data is integrated, avoiding unnecessary information, while optimizing computation time.

Following the results obtained in the ablation study, the approach was tested in real-world environments. As expected, this brought many challenges to suppress, such as camera sensor preparation, non-reliable communication network, robots odometry, logistic efforts, etc. Yet, after long periods of testing, the approach was analysed similarly to the ablation study, where the robots exhibited appropriate cooperation, and autonomous operation, while fully exploring the environment. These experiments have also confirmed that the proposed approach is agnostic and not specific to a given sensing configuration or robotic platform.

Once proper communication is established between robots under the same network, due to the two-level coordination approach, they can perform autonomous exploration with no collisions or robot interference. Every time a metric or topological conflict occurs between two robots, both decide which one should move to the selected target and which one should assigns to a new target.

The Pioneer 3-DX robot was not designed for navigation in irregular terrain, but even so, it has succeeded to move up and down the ramps presented in one of the real-world environments, leading to the validation of the ability and efficiency of the approach to plan and navigate in irregular surfaces.

Communication remains to be an enormous challenge when operating not only with a team of robots but operating in challenging environments, such as the *Auditorium* environment. Also, the capability of the robot to localize itself in an environment with lack of textures where the main source of odometry was visual odometry tends to induce errors and leads to some drifts in the created map. Adding to this, there is the probability of the wheel odometry acquiring an incorrect estimation when the robot performs long in-turn rotations.

The localization-mode technique utilized has proven not to be so effective as expected, much because of the lack of textures in the environment and the fact that pose estimation is acquired only by combined visual and wheeled odometry. The *Auditorium* environment is, in fact, very challenging for the combined visual and wheel odometry, as it entails similar regions in which similar depth data can be easily perceived. This has an impact on to the system, resulting in some glitches on the created map by the robots. Yet, as each robot obtains information from others, they stand out to respond to that defective map and continue the exploration mission.

For the onboard processing system, the use of laptops is not ideal, since their autonomy is very reduced and the irregularity of the movements and terrain may cause some progressive damage over time. Even so, if laptops are used, they should be equipped with a Solid-State Drive (SSD) instead of a Hard-Disk Drive (HDD) for increased robustness to damage. Denoting that using an SSD also increases the processor capacity of the computer.

To sum up, the presented approach turned out to be efficient and has shown appropriate management of a team of robots. As it is decentralized, it is possible for a robot to fail and even then, the others may proceed with the mission and finish as soon as no more information can be acquired by them.

6.2 Future Work

Following the developed work in the course of this dissertation, some features can be improved and implemented to better increase the efficiency of the system, suppressing some of the weaknesses presented and extending it to more complex scenarios.

Communication availability is crucial to this system, and for that reason, there is the need to perform a deeper analysis of the bandwidth requirements and the information that needs to be shared between robots. Implementing a technique to compress information in order to send it without data loss may allow the approach to be tested in other type of environments, where these constrictions are noticed.

Another way to extend this approach to hazardous environments is to integrate a strategy capable of maintaining connection stability. The integration of an Ad Hoc network where the robots create their network and establish communication on a peer-to-peer basis is an evident future direction of this work. Some relevant work in this area was presented by Slyusar, *et al.* [49], Araújo, *et al.* [50], Gomes, *et al.* [51] and Fernandes, *et al.* [52].

The SLAM technique should be improved, probably integrating a LiDAR, an Inertial Measurement Unit (IMU) and even a compass. Fusing multiple sensor systems may help to increase the capacity of robots to better perceive their motion while moving in complex environments. Using different sources of sensors to do it turns this technique more robust and fault-tolerant, able to suppress errors from other sensors. If the correct localization of the robot is maintained, the exploration mission will not suffer from this, the map will be successfully created with no misalignments and robots will perform accurate mapping along the mission.

In order to extend this approach into a deeper validation, a benchmark against other 3D multi-robot exploration approaches should be deployed. For instance, while developing the work in this dissertation, other multi-robot exploration strategies have been proposed such as [32]. It would be interesting to assess the performance of these strategies under the same conditions as ours. Moreover, it would be important to simulate several more robots within the exploration team to further assess its scalability.

The Realsense camera is an appropriate option to obtain a 3D dense reconstruction of objects placed at some proximity distance. As this camera uses the IR stereo principle, it shows great performance in outdoor environments and, for that reason, an optimal usage of this camera would be to perceive the floor in front of the robot, replacing the Orbbec Astra that is currently in use. This way, complex outdoor environments can be tested in order to

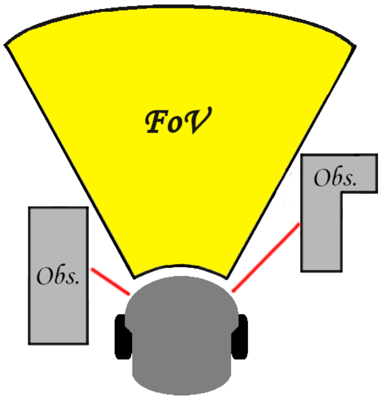


Figure 6.1: Robot using an extra sensing source in order to perceive closer objects that are not detected in the FoV.

validate this method.

In order for robots to move in tight spaces, it is important to integrate some sensors to better perceive the surroundings of the robot so that it does not move into an obstacle. The robot may not detect obstacles outside the field of view of the onboard sensors, which may lead to collisions or navigation deadlocks (see Figure 6.1). With the integration of a LiDAR, additional depth cameras or even sonars, it is possible to avoid those collisions, perceiving obstacles where the sensor cameras do not overlap. Another important avenue to explore would be to tune the trajectory controller in order to slow down the robot's rotations even more, favoring the odometry estimation.

Also, to expand the validation of the presented approach, experimental tests may take place in even more complex environments than the *Auditorium*, allowing the deployment of a larger team of robots. This can also be used to evaluate the scalability of the system under even more challenging conditions.

7 Bibliography

[1] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, “A UAV System for Inspection of Industrial Facilities,” in *2013 IEEE Aerospace Conference*, pp. 1–8, IEEE, 2013.

[2] S. Omari, P. Gohl, M. Burri, M. Achtelik, and R. Siegwart, “Visual Industrial Inspection Using Aerial Robots,” in *Proceedings of the 2014 3rd International Conference on Applied Robotics for the Power Industry*, pp. 1–5, IEEE, 2014.

[3] R. Khanna, M. Möller, J. Pfeifer, F. Liebisch, A. Walter, and R. Siegwart, “Beyond Point Clouds-3D Mapping and Field Parameter Measurements using UAVs,” in *2015 IEEE 20th conference on emerging technologies & factory automation (ETFA)*, pp. 1–4, IEEE, 2015.

[4] A. K. Rangarajan and R. Purushothaman, “A Vision Based Crop Monitoring System Using Segmentation Techniques,” *Advances in Electrical and Computer Engineering*, vol. 20, no. 2, pp. 89–100, 2020.

[5] F. Colas, S. Mahesh, F. Pomerleau, M. Liu, and R. Siegwart, “3D Path Planning and Execution for Search and Rescue Ground Robots,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 722–727, IEEE, 2013.

[6] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, “Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision,” *IEEE Access*, vol. 8, pp. 191617–191643, 2020.

[7] E. Olcay, F. Schuhmann, and B. Lohmann, “Collective navigation of a multi-robot system in an unknown environment,” *Robotics and Autonomous Systems*, vol. 132, p. 103604, 2020.

[8] M. García, D. Puig, L. Wu, and A. Solé, “Voronoi-Based Space Partitioning for Coordinated Multi-Robot Exploration,” *JoPha: Journal of Pysical Agents, ISSN 1888-0258, Vol. 1, Nº. 1, 2007, pags. 37-44*, vol. 1, 01 2007.

[9] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding Horizon "Next-Best-View" Planner for 3D Exploration,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1462–1468, May 2016.

[10] M. Selin, M. Tiger, D. Duberg, F. Heintz, and P. Jensfelt, “Efficient Autonomous Exploration Planning of Large-Scale 3-D Environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[11] B. Yamauchi *et al.*, “Frontier-Based Exploration Using Multiple Robots,” in *Agents*, vol. 98, pp. 47–53, 1998.

[12] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated Multi-Robot Exploration,” *IEEE Transactions on Robotics*, vol. 21, pp. 376–386, June 2005.

[13] R. Rocha, F. Ferreira, and J. Dias, “Multi-Robot Complete Exploration using Hill Climbing and Topological Recovery,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1884–1889, 2008.

[14] R. G. Colares and L. Chaimowicz, “The Next Frontier: Combining Information Gain and Distance Cost for Decentralized Multi-Robot Exploration,” in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 268–274, 2016.

[15] J. Borenstein and L. Feng, “UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots,” in *Mobile Robots X* (W. J. Wolfe and C. H. Kenyon, eds.), vol. 2591, pp. 113 – 124, International Society for Optics and Photonics, SPIE, 1995.

[16] M. Labbé and F. Michaud, “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[17] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *CIRA*, vol. 97, p. 146, 1997.

[18] M. Weinmann, B. Jutzi, and C. Mallet, “Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features,” *ISPRS Annals*

of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 2, no. 3, p. 181, 2014.

[19] A. Hermann, F. Mauch, K. Fischnaller, S. Klemm, A. Roennau, and R. Dillmann, "Anticipate your Surroundings: Predictive Collision Detection between Dynamic Obstacles and Planned Robot Trajectories on the GPU," in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–8, IEEE, 2015.

[20] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[21] R. P. Rocha, *Building Volumetric Maps With Cooperative Mobile Robots And Useful Information Sharing - A Distributed Control Approach based on Entropy.* PhD thesis, Faculdade de Engenharia do Porto, 2006.

[22] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A Probabilistic Approach to Collaborative Multi-Robot Localization," *Autonomous robots*, vol. 8, no. 3, pp. 325–344, 2000.

[23] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot SLAM system for 3D LiDARs," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1004–1011, IEEE, 2017.

[24] C. Dornhege and A. Kleiner, "A Frontier-Void-Based Approach for Autonomous Exploration in 3D," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 351–356, Nov 2011.

[25] M. Keidar and G. A. Kaminka, "Efficient Frontier Detection for Robot Exploration," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 215–236, 2014.

[26] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information Based Adaptive Robotic Exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 540–545 vol.1, 2002.

[27] C. Connolly, "The determination of next best views," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 432–435, IEEE, 1985.

[28] J. E. Banta, Y. Zhien, X. Z. Wang, G. Zhang, M. Smith, and M. A. Abidi, "A "Best—Next—View" Algorithm for Three—Dimensional Scene Reconstruction Using

Range Images," in *Intelligent Robots and Computer Vision XIV: Algorithms, Techniques, Active Vision, and Materials Handling*, vol. 2588, pp. 418–429, International Society for Optics and Photonics, 1995.

[29] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[30] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: A new tool for path planning," in *Proceedings. 1999 IEEE International Conference on Robotics and Automation*, pp. 473–479, 1999.

[31] M. Moors, *Koordinierte multi-robot exploration.* PhD thesis, Master's thesis, Department of Computer Science, University of Bonn, 2000.

[32] K. Ebadi, Y. Chang, M. Palieri, A. Stephens, A. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, L. Carlone, and A. a. Agha-mohammadi, "LAMP: Large-Scale Autonomous Mapping and Positioning for Exploration of Perceptually-Degraded Subterranean Environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 80–86, 2020.

[33] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[34] R. Rocha, J. Dias, and A. Carvalho, "Cooperative multi-robot systems:: A study of vision-based 3-D mapping using information theory," *Robotics and Autonomous Systems*, vol. 53, no. 3-4, pp. 282–311, 2005.

[35] L. Freda, M. Gianni, F. Pirri, A. Gawel, R. Dubé, R. Siegwart, and C. Cadena, "3D Multi-Robot Patrolling with a Two-Level Coordination Strategy," *Autonomous Robots*, vol. 43, pp. 1747–1779, Oct 2019.

[36] S. Thrun and A. Bücken, "Learning maps for indoor mobile robot navigation.," tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1996.

[37] J. Garcia-Luna-Aceves and H.-P. Dommel, "Tree-based Ordered Multicasting Method," Apr. 18 2006. US Patent 7,031,308.

[38] M. Skrodzki, "The k-d tree data structure and a proof for neighborhood computation in expected logarithmic time," *arXiv preprint arXiv:1903.04936*, 2019.

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

[40] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *ArXiv*, vol. abs/1405.4506, 2016.

[41] A. Tiderko, F. Hoeller, and T. Röhling, *The ROS Multimaster Extension for Simplified Deployment of Multi-Robot Systems*, pp. 629–650. Cham: Springer International Publishing, 2016.

[42] S. H. Juan and F. H. Cotarelo, "Multi-master ROS systems," *Institut de Robotics and Industrial Informatics*, pp. 1–18, 2015.

[43] L. Nogueira, "Comparative Analysis Between Gazebo and V-REP Robotic Simulators," vol. 2014, 2014.

[44] D. B. S. Portugal, *Effective cooperation and scalability in multi-robot teams for automatic patrolling of infrastructures.* PhD thesis, 2014.

[45] D. Portugal, L. Iocchi, and A. Farinelli, *A ROS-Based Framework for Simulation and Benchmarking of Multi-robot Patrolling Algorithms*, pp. 3–28. Cham: Springer International Publishing, 2019.

[46] MobileRobots, *Pioneer 3 Operations Manual.* 2006.

[47] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[48] T. Balch and R. C. Arkin, "Communication in Reactive Multiagent Robotic Systems," *Autonomous robots*, vol. 1, no. 1, pp. 27–52, 1994.

[49] K. Slyusar and M. Kulich, "Framework for ad hoc network communication in multi-robot systems," *Acta Polytechnica CTU Proceedings*, vol. 6, pp. 18–27, 11 2016.

[50] F. Araujo, J. Santos, and R. P. Rocha, "Implementation of a Routing Protocol for Ad Hoc Networks in Search and Rescue Robotics," in *2014 IFIP Wireless Days (WD)*, pp. 1–7, IEEE, 2014.

[51] A. F. S. Gomes *et al.*, *CHOPIN-MANET II–Comunicação Ad Hoc para Robótica de Busca e Salvamento.* PhD thesis, Universidade de Coimbra, 2015.

[52] A. Fernandes, M. Couceiro, D. Portugal, J. Santos, and R. Rocha, "Ad Hoc Communication in Teams of Mobile Robots Using ZigBee Technology," *Computer Applications in Engineering Education*, vol. 23, 04 2015.

8 Appendix

8.1 Detailed Experimental Results

For the next tables, when performing multi-robot exploration, note that $mean_{tot}$ of the t_{expl} is the only metric that uses the maximum time of the test (highlighted value) since it is the maximum time that defines the total time of the exploration mission t_{expl}. For the rest, the $mean_{tot}$ takes into account all the robots in test, presenting the mean time or distance that an element of the team took to conclude the mission.

8.1.1 ISRsea scenario

Machine	t_{expl} **(s)**	t_{95expl} **(s)**	occ_{95}	t_{end} **(s)**	d_{trav} **(m)**
PC1	452	310	6199	142	27
	492	352	7396	140	27
	578	424	6997	154	36
PC2	617	430	6649	187	35
	515	379	6719	136	27
	708	469	6615	239	34
PC2	547	460	5655	87	36
	623	393	5879	230	39
	841	665	6571	176	42
$mean_{tot}$	**597**	**431**	**6520**	**166**	**34**

Table 8.1: Results of exploration tests using one robot with different machines.

Test nr	t_{expl} (s)		t_{95expl} (s)		$occ_{explored}$ (%)		t_{end} (s)		d_{trav} (m)	
	rob1	**rob2**	**rob1**	**rob2**	**rob1**	**rob2**	**rob1**	**rob2**	**rob1**	**rob2**
1	**585**	273	**415**	173	**100**	92	**170**	100	32	**10**
2	**581**	320	**436**	179	98	**100**	**145**	141	28	**15**
3	**580**	287	**381**	222	**100**	97	**199**	65	28	**15**
$mean_{tot}$	**582**		**301**		**98**		**137**		**21**	

Table 8.2: Results of multi-robot exploration tests using 2 robots.

Test nr	t_{expl} (s)			t_{95expl} (s)			$occ_{explored}$ (%)			t_{end} (s)			d_{trav} (m)		
	rob1	**rob2**	**rob3**	**rob1**	**rob2**	**rob3**	**rob1**	**rob2**	**rob3**	**rob1**	**rob2**	**rob3**	**rob1**	**rob2**	**rob3**
1	308	**461**	210	177	**237**	140	76	**100**	68	131	**225**	70	15	21	**12**
2	308	**446**	354	193	**329**	328	74	**100**	92	115	**117**	26	19	20	**19**
3	**336**	247	199	**196**	190	158	73	**100**	79	**140**	57	42	25	12	**9**
$mean_{tot}$		**414**			**216**			**85**			**102**			**17**	

Table 8.3: Results of multi-robot exploration tests using 3 robots.

8.1.2 Auditorium scenario

Test nr	t_{expl} (s)		t_{95expl} (s)		$occ_{explored}$ (%)		t_{end} (s)		d_{trav} (m)	
	rob1	**rob2**	**rob1**	**rob2**	**rob1**	**rob2**	**rob1**	**rob2**	**rob1**	**rob2**
1	301	**654**	234	**584**	74	**100**	67	**70**	**14**	46
2	471	**493**	**415**	414	91	**100**	56	**79**	34	**31**
3	259	**652**	180	**534**	67	**100**	79	**118**	**17**	35
$mean_{tot}$	**600**		**386**		**89**		**78**		**29**	

Table 8.4: Results of multi-robot exploration tests using 2 robots.

Test nr	t_{expl} (s)			t_{expl95} (s)			$occ_{explored}$ (%)			t_{end} (s)			d_{trav} (m)		
	rob1	**rob2**	**rob3**	**rob3**	**rob2**	**rob3**	**rob1**	**rob2**	**rob3**	**rob1**	**rob2**	**rob3**	**rob1**	**rob2**	**rob3**
1	163	422	**973**	110	372	**844**	62	86	**100**	53	50	**129**	**9**	26	32
2	184	235	**410**	146	158	**238**	90	93	**100**	38	70	**172**	**13**	14	26
$mean_{tot}$		**691**			**311**			**88**			**86**			**20**	

Table 8.5: Results of multi-robot exploration tests using 3 robots.