FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



UVC Dose Mapping by Mobile Robots

Beatriz Pinto

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor at INESC TEC: Dr. Marcelo R. Petry

Supervisor at FEUP: Prof. Dr. António Paulo Moreira

June, 2021

© Beatriz Pinto, 2021

Abstract

Hospital-acquired infections are a persistent and increasing problem and their prevention involves disinfecting areas and surfaces. The necessity for effective disinfection methods has highly increased in consequence of the Covid-19 pandemic. An effective method is using UVC exposure because UVC radiation is absorbed by nucleic acids and, therefore, is able to inactivate microorganisms. This method also brings many advantages when compared with traditional disinfection methods. UVC disinfection can be performed by portable equipment that have to be moved by a person from place to place to disinfect an entire area, or by an autonomous mobile robot that requires minimal human intervention to completely disinfect an environment.

This dissertation is focused on the mapping module of mobile robots that disinfect an environment using UVC radiation. These mobile robots are able to move autonomously while mapping the surrounding environment and simultaneously disinfecting it. The robots keep track of the dose applied to each area of the environment in order to build a dose map and differentiate areas that are completely disinfected from those that are not. This solution has the advantage of the robot performing UVC disinfection without needing to stop in each area nor having previous knowledge of the environment.

The mapping module developed uses as a base the SSL_SLAM algorithm. This is a recent algorithm that is inspired by LOAM and OctoMap. The SSL_SLAM algorithm was modified to create the desired UVC dose map. Experiments of this solution were performed using rviz, a Robot Operating System (ROS) visualization tool, and the LiDAR Camera L515. The LiDAR was used to capture the necessary information for creating the map of the environment and rviz was used to visualize the dose map. The different experiments were performed using only the L515 sensor and not the entire disinfection system and under different conditions such as lighting conditions and moving speed of the sensor. The resulting maps of these experiments revealed limitations of the algorithm and points of improvement, and the best performance was obtained when there was minimal sunlight interference and low rotation speed of the sensor.

Keywords: UVC, dose, OctoMap, 3D map.

Resumo

As infeções adquiridas em ambientes hospitalares são um problema persistente e crescente e a sua prevenção envolve a desinfeção de áreas e superfícies. A necessidade de métodos de desinfeção eficazes aumentou muito em consequência da pandemia do Covid-19. Um método eficaz é a utilização de exposição UVC porque a radiação UVC é absorvida pelos ácidos nucleicos e, portanto, é capaz de inativar microrganismos. Este método também traz muitas vantagens quando comparado com os métodos tradicionais de desinfeção. A desinfeção UVC pode ser realizada por equipamentos fixos que têm de ser deslocados de um local para outro de modo a desinfetar toda uma área, ou por um equipamento móvel autónomo que requer intervenção humana mínima para desinfetar completamente um ambiente.

Esta dissertação foca-se no módulo de mapeamento de robôs móveis que desinfetam um ambiente utilizando radiação UVC. Estes robôs móveis são capazes de se mover autonomamente enquanto mapeiam o ambiente à sua volta e simultaneamente o desinfetam. Os robôs mantêm um registo da dose aplicada a cada área do ambiente de modo a construir um mapa da dose e diferenciar as áreas completamente desinfetadas das que não o estão. Esta solução tem a vantagem de o robô realizar a desinfeção UVC sem necessitar de parar em cada área nem ter conhecimentos prévios sobre o ambiente.

O módulo de mapeamento desenvolvido utiliza como base o algoritmo SSL_SLAM. Este é um algoritmo recente inspirado no LOAM e no OctoMap. O algoritmo SSL_SLAM foi modificado para criar o mapa da dose UVC desejado. Experiências desta solução foram realizadas utilizando o rviz, uma ferramenta de visualização do Robot Operating System (ROS), e a LiDAR Camera L515. O LiDAR foi utilizado para capturar a informação necessária para a criação do mapa do ambiente e o rviz foi utilizado para visualizar o mapa da dose. As diferentes experiências foram realizadas utilizando apenas o sensor L515 e não todo o sistema de desinfecção e sob diferentes condições tais como condições de iluminação e velocidade de deslocação do sensor. Os mapas resultantes destas experiências revelaram limitações do algoritmo e pontos de melhoria, e o melhor desempenho foi obtido quando havia uma interferência mínima da luz solar e baixa velocidade de rotação do sensor.

Palavras-chave: UVC, dose, OctoMap, mapa 3D.

Acknowledgements

To my supervisors, Doctor Marcelo R. Petry and Professor Doctor António Paulo Moreira, I appreciate and am very thankful for all the support, trust, availability, and guidance given to me throughout the development of this dissertation. Thank you for the opportunity to collaborate with you both and to develop this work.

Throughout my five year journey I was accompanied and supported by many special people that deserve my utmost appreciation: Ana Cruz, André Leão, Bruno Gonçalves, Bruno Maurício, Carolina Correia, Eduardo Fonseca, Francisco Parente, Hélder Pereira, Hugo Pires, João Couto, João Fernandes, João Pires, Leonardo Ribeiro, Michael Rosendo, Nuno Lobo, Nuno Minhoto, Ricardo Costa, Ricardo Sousa, Rodrigo Morais, Rodrigo Silva, and Tiago Couto. You all have impacted positively my journey in FEUP and I will always treasure the infinite hours spent in 328 studying, working or goofing around. You truly have given me the best moments of my life. To my best friends, Mikas, Bruno and Rodrigo I am extremely thankful for your friendship and for your overall support, I have to give you huge credit for the role you played in my achievements and successes.

I also could not forget to mention my family at NEEEC. You have helped me develop and push myself throughout the years and become a better version of myself. A special mention to Sofia Teixeira, José Pedro, Tomás Naldinho, Beatriz Loureiro, Inês Brandão, Miguel Almeida, Pedro Claro, Rita Azevedo, Francisco Silva, and Nuno Neto.

Furthermore, I am thankful for all the support and care my family has given me throughout all my life. Lastly and most importantly, I have to thank my sister Catarina Pinto, who is the most important person in my life and my biggest supporter, and has always been a sister, a mother and a best friend.

Beatriz Pinto

"Scientists study the world as it is, engineers create the world that never has been."

Theodore von Kármán

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives . . . 3
- 1.4 Structure of the document . . . 4

2 Literature Review **5**
- 2.1 UVC Disinfection . . . 5
  - 2.1.1 Fundamental concepts . . . 6
  - 2.1.2 Mathematical Modelling . . . 8
  - 2.1.3 UVC inactivation effectiveness of the Covid-19 virus . . . 10
  - 2.1.4 Advantages and disadvantages . . . 12
  - 2.1.5 Safety . . . 13
  - 2.1.6 Existing solutions . . . 14
- 2.2 Mapping . . . 17
  - 2.2.1 Fundamental concepts . . . 18
  - 2.2.2 3D Mapping Techniques . . . 19

3 UVC Dose Map **24**
- 3.1 Technologies used . . . 24
  - 3.1.1 SSL_SLAM method . . . 25
  - 3.1.2 Depth Camera L515 . . . 28
  - 3.1.3 Robot Operating System . . . 28
- 3.2 UVC Dose Mapping Implementation . . . 29
- 3.3 Visualization Method . . . 35

4 Results **38**
- 4.1 Experiment 1 . . . 39
- 4.2 Experiment 2 . . . 40
- 4.3 Experiment 3 . . . 43
- 4.4 Experiment 4 . . . 44
- 4.5 Experiment 5 . . . 46
- 4.6 Discussion . . . 47

5 Conclusions and Future Work **49**
- 5.1 Conclusions . . . 49
- 5.2 Future Work . . . 50

References **53**

List of Figures

2.1 UV bands [1] . . . 6
2.2 Formation of thymine dimers as a result of UV radiation absorption, in DNA [1] 7
2.3 Limit UV exposure time. Source: ICNIR, International Comission on Non-Ionizing Radiation Protection [2] . . . 14
2.4 UVC based disinfection systems for air or surfaces [1] . . . 15
2.5 3D representations of a tree using a laser range sensor (from left to right): Point Clouds, Elevation map, MLS map, OctoMap (only the occupied volumes are represented) [3] . . . 22

3.1 2D example of cells whose UVC dose value can and cannot be updated because of the three conditions presented. Cells in red (A, B, D, E) cannot be updated. Cells in green (C, F) can be updated. . . . 33
3.2 (a) 3D map built with the SSL_SLAM method. (b) 3D dose map built of the same area without using markers. . . . 36
3.3 Example of the progression of the UVC dose value of a surface mapped. From (a) to (c) the dose value increases. . . . 36
3.4 Graph with all active ROS nodes and topics during the execution of the UVC dose mapping algorithm . . . 37

4.1 Environment 1 . . . 40
4.2 Evolution of the 3D dose map built using a test file provided by the authors of the SSL_SLAM method. The progress is represented from (a) to (d). . . . 41
4.3 Top view of the final 3D dose map built using a test file provided by the authors of the SSL_SLAM method (environment 1). . . . 41
4.4 Environment 2 . . . 42
4.5 UVC dose mapping with the L515 sensor attached to a remote controlled wheelchair in environment 2. . . . 42
4.6 UVC dose mapping with the L515 sensor hand-held in environment 2. . . . 43
4.7 Top view of the 3D maps built with the sensor attached to a remote controlled wheelchair, (a), and hand-held, (b), in environment 2. . . . 43
4.8 Top view of the map of environment 2 with a high sensor moving speed. . . . 44
4.9 Environment 3 . . . 44
4.10 Top view of the maps of environment 3, when moving the sensor at a low speed, (a), and at a higher speed, (b), except when rotating. . . . 45
4.11 Environment 4 . . . 46
4.12 Results from the mapping algorithm of environment 4 in the presence of sunlight. 46
4.13 Results from the mapping algorithm of environment 3 in extreme lighting conditions. 47
4.14 Mapping of environment 2 with an obstacle. . . . 47

4.15 Mapping of environment 2 with another obstacle on the ground. 48

Chapter 1

Introduction

The present chapter is divided into four sections. The first section (Section 1.1) presents the context for the dissertation and briefly presents a few existing solutions for the problem. The second section (Section 1.2) describes the motivation for developing an autonomous mobile robot that performs UVC disinfection. The third section (Section 1.3) presents the main objectives to be fulfilled in this dissertation as well as all the other associated goals. Lastly, the fourth section (Section 1.4) presents the structure of this document.

1.1 Context

Nosocomial infections, or in other words hospital-acquired infections, are a persistent problem found in hospital environments. Some of these infections may spread to outside of the hospital into communities and are increasingly caused by drug-resistant microbes [1]. An infection caused by a drug-resistant microbe cannot be effectively treated by medicine, which redirects the focus from treating to preventing. Contaminated surfaces are a way of spreading infections, by contact or through air. In order to prevent the spread of nosocomial infections, surfaces and air must be disinfected.

The importance of disinfecting health care facilities, as well as other indoor environments that aggregate multiple people such as airports or nursing homes, exponentially increased during the year 2020, as a result of the Covid-19 world pandemic. The best measure against the novel coronavirus is prevention. To prevent the virus from spreading, disinfection is essential, especially in hospitals, which are over capacitated, particularly in human resources.

In hospitals, traditionally, the disinfection process is based on cleaning by chemical means. This process can be hazardous, since the products may be toxic or corrosive. An effective alternative is UVC disinfection.

For disinfecting air and surfaces, UV (ultraviolet) light with wavelengths between the germicidal range ($200 - 320$ *nm*) is used, which corresponds to the UVB and UVC bands. UVC ($200 - 280$ *nm*) irradiation is particularly effective in inactivating viruses and bacteria because it

is absorbed by the nucleic acids (DNA and RNA) [1]. By causing photochemical changes in nucleic acids, that are responsible for microbial replication and protein synthesis, electromagnetic radiation can destroy the ability of microorganisms to reproduce, inactivating them.

Besides being highly effective in inactivating microbes, the use of UVC in disinfection has other advantages. UV light has a high replacement cycle, which means it is economical and environment friendly [4]. The use of a UV disinfection system does not require human contact with surfaces and reduces the exposure of operators to toxic or corrosive agents. It is important to note that UVC radiation immediately dissipates, so it is safe to enter a room immediately after disinfection.

UVC radiation, which is invisible to the human eye, also has a few significant disadvantages. Unprotected exposure to UVC radiation can be dangerous to the human health, even if unprotected exposure during small periods of time is tolerated. Some of the negative effects of unprotected UVC exposure are eye damage, skin burns, and even potentially skin cancer [1]. These negative effects are not verified when using a UVC lamp with a wavelength of 222 *nm* [5] [6], however further long-term studies have to be made.

The most common UVC disinfection systems found are Cooling coil disinfection systems, After Hours UVGI (ultraviolet germicidal irradiation) systems and Portable UVGI systems [1]. Most of these are fixed, not portable, and require an operator to manually activate and deactivate them, monitor, or manually move them from point to point where disinfection is required. Because these systems require human intervention they are subjected to human error. An innovative and possibly more effective solution is the use of autonomous mobile robots to perform the task of disinfecting surfaces in a given area using UVC light.

In this dissertation, the proposed mapping module of an autonomous mobile robot, which would be a complete disinfection system, generates a map of the surrounding environment and estimates the required UVC dose to be applied to each cell of the map in order to improve the radiation distribution, allowing the robot to move instead of having to stop for several minutes. The complete disinfection system would use UV lamps, should automatically interrupt the irradiation upon human detection, should be capable to localize itself and navigate through the contaminated rooms and autonomously disinfects the area.

1.2 Motivation

As stated before, hospital-acquired infections are a persistent and increasing problem that justifies the need for disinfection systems in health care facilities. The importance of disinfecting these facilities exponentially increased during the year 2020, as a result of the Covid-19 world pandemic. The novel coronavirus is a highly contagious disease that can be transmitted through the air or by direct contact with contaminated surfaces. When an infected person sneezes, coughs or even exhales, the air around them gets contaminated and the heavier droplets fall onto the floor and surfaces [7]. A non-infected person may become infected when in contact with the contaminated air or surfaces.

UVC disinfection systems have been shown to be extremely efficient in inactivating viruses and bacteria. Recently mobile robots have been developed for performing UVC disinfection. The approaches that already exist are not able to guaranty that the UV irradiation dose is uniform in every surface, and many of the mobile robots are not autonomous and therefore require an operator to manually transport them from point to point and monitor them. These problems present the need for an autonomous mobile robot for performing UVC disinfection.

Another limitation of the existing solutions is that the robots are required to stop in a certain location and not move for several minutes. Through mapping and estimating the required UVC dose in each point of the map of the environment, the disinfection procedure can be more efficient and the applied dose more uniform.

Motivated not only by the Covid-19 pandemic, but also by the problem of the increasing number of drug-resistant microbes and of hospital-acquired infections, autonomous mobile robots equipped with UVC disinfection units present themselves as a possible solution. In this dissertation, the mapping module of such disinfecting robots is presented. A 3D map of the environment should be built and the required UVC exposure dose for each point of the map should be determined and updated, which would result in a uniform UV distribution. This mapping algorithm is one of the modules that have to be integrated to later achieve the disinfecting mobile robot.

1.3 Objectives

As explained previously, there is a necessity for an improved disinfection system, such as an autonomous mobile robot that uses UVC radiation to disinfect surfaces and areas of an indoor environment. This mobile robot should autonomously build a map of the surrounding environment and determine the required UVC dose in each point of the map in order to apply the minimal UVC dose required to every surface and maximize efficiency. The main objective of this dissertation is to develop a UVC dose mapping algorithm that integrates the mobile robot described. This mapping module to be developed should track the UVC dose applied to each surface and represent the current state of the disinfection process. This UVC dose map could be used to autonomously optimize the trajectory and total disinfection time (future work) and without it, the control system of the disinfecting robot (to be developed in the future) would not be able to guarantee complete disinfection of the environment.

There are other overall objectives that a system as the one described would be able to achieve, such as the reduction and prevention of infectious diseases caused by viruses or bacteria in hospital environments, redirecting and freeing human resources from disinfection procedures to other essential tasks, and the reduction of exposure by health care professionals to corrosive and toxic products used in traditional disinfection systems.

1.4 Structure of the document

This document is divided into five chapters. Chapter 1 contextualises the dissertation's theme, presents the motivation for the development of the proposed solution, the goals to be achieved, and the document's structure. Chapter 2 presents in detail concepts regarding the two main themes related with the development of the proposed solution (UVC disinfection and Mapping) as well as the literature review for the same themes. This chapter is divided into two main sections, one regarding UVC disinfection (Section 2.1) and one regarding mapping (Section 2.2). Following the introductory chapters, Chapter 3 presents the development process and implementation of the desired mapping algorithm. It is divided in three sections, Section 3.1 presents the main technologies used for the development and validation tests of the algorithm, Section 3.2 describes in detail the mapping algorithm, as well as its functionalities, characteristics and desired behaviour, and Section 3.3 presents the visualization algorithm developed to allow visual monitoring of the disinfection process. Chapter 4 presents the results obtained through the validation tests performed. Sections 4.1, 4.2, 4.3, 4.4 and 4.5 present the visual results of the validation tests (the 3D maps built), and Section 4.6 discusses and evaluates them. Finally, Chapter 5 presents the final conclusions regarding the mapping module developed and its performance (Section 5.1), and suggests the future developments that would improve the mapping algorithm developed and that would expand it in order to achieve a complete autonomous disinfection system (Section 5.2).

Chapter 2

Literature Review

As stated previously, a possible solution for the problem presented is an autonomous mobile robot that performs UVC disinfection. This robot has innovative features, which are a result of its main functionality: the robot builds a map of the surrounding environment and determines and updates the minimal required UVC dose for each point of the map. In this dissertation, the mapping module of the mobile robot, which is the complete disinfection system, was developed.

In this chapter, the studied concepts and solutions found regarding key aspects for the development of the proposed system are presented. One key aspect is understanding how UVC can be effective in inactivating microorganisms and how to use UVC light for this end. For this reason, firstly, the fundamental concepts of UVC inactivation (Subsection 2.1.1) will be presented, followed by the mathematical model of UVC disinfection (Subsection 2.1.2), effectiveness of UVC in inactivating SARS-CoV-2 (Subsection 2.1.3), advantages and disadvantages of UVC disinfection (Subsection 2.1.4), safety measures (Subsection 2.1.5), and existing UVC disinfection systems (Subsection 2.1.6). The other key aspect of the proposed system is the development of a mapping system. For this reason, mapping concepts will be presented (Subsection 2.2.1), followed by 3D mapping techniques (Subsection 2.2.2) and 3D mapping techniques in ROS (Subsection 2.2.2.1).

2.1 UVC Disinfection

UVC disinfection has been used for the past years to help stop the spread of hospital-acquired infections. These infections are a persistent problem and an increasing number of drug-resistant microbes have been registered. Several studies verify the effectiveness of UVC disinfection in inactivating microorganisms in water, air, and surfaces. The following sections will be focused on the use of UVC for disinfecting surfaces. One of the key elements in UVC disinfection is discovering the required UV exposure dose to be applied to the contaminated areas. Recently, because of the Covid-19 pandemic, the necessity for a disinfection method that does not require direct contact with contaminated surfaces has highly increased. UVC disinfection systems do not require direct contact with infected surfaces. In addition to this, UVC disinfection is a non-chemical disinfection method, which gives it another advantage in relation to traditional disinfection methods.

The most effective way to combat the Covid-19 pandemic, as well as many other epidemics and pandemics, is by prevention, and prevention passes through disinfecting contaminated areas and surfaces, since the novel coronavirus is able to remain infective for 24 *h* to 72 *h* on surfaces [8].

The basics for UVC disinfection, including the process of how UVC light inactivates microorganisms, will be presented firstly. As stated previously, determining the appropriate UV dose is a key factor in UVC disinfection, for this reason a section for presenting the concept of UV exposure dose and the mathematical models for UVC disinfection will follow the initial section. UVC disinfection has advantages and disadvantages when compared to other disinfection methods, which will also be discussed. A section that covers the necessary safety measures for using UVC disinfection systems will also be presented. Lastly, the solutions already in existence in the market will be presented, focusing on solutions that involve mobile autonomous robots.

2.1.1 Fundamental concepts

Ultraviolet (UV) radiation spectrum may be divided in several subtypes such as UVA, UVB, UVC and VUV. Definitions are given in figure 2.1. UV radiation with a wavelength bellow 320 *nm* is considered actinic, which means UVB, UVC and VUV radiation cause photochemical reactions [1].

The range between 200 and 320 *nm*, which encompasses UVB and UVC, is called the germicidal range because UV light inside this range is absorbed by the nucleic acids of microorganisms causing structural changes in DNA and RNA and may lead to the inactivation of the microorganism [9]. The inactivation of a microorganism means that it is no longer able to replicate and multiply. If it cannot replicate, then it cannot cause infections or disease, even if it is still metabolically alive.

<table>
<tr><th>Band</th><th>Wavelength, nm</th><th colspan="3">Type and classification</th></tr>
<tr><td>UVA</td><td>320–400</td><td colspan="3">Non-germicidal (Near-UV, Blacklight)</td></tr>
<tr><td>UVB</td><td>280–320</td><td>Erythemal</td><td rowspan="2">Germicidal</td><td rowspan="3">Actinic</td></tr>
<tr><td>UVC</td><td>200–280</td><td>Ozone-producing</td></tr>
<tr><td>VUV</td><td>100–200</td><td colspan="2">Vacuum ultraviolet</td></tr>
</table>

Figure 2.1: UV bands [1]

Disinfection implies a reduction in the microbial population, which can be achieved by using UV radiation in the germicidal range. To understand UV disinfection, it is needed to firstly present the structure of DNA and RNA of microorganisms.

Deoxyribonucleic acid (DNA) and Ribonucleic acid (RNA) are macromolecules involved in many cellular processes and are responsible for microbial replication protein synthesis. DNA has a double helix structure of two strands of nucleotides, while RNA has a single strand structure. Nucleotides are composed by deoxyribose (in DNA) or ribose (in RNA), a phosphate group responsible for the connection of the nucleotides along the strand, and one out of the four nitrogenous bases [1]. Regarding DNA, these four bases are adenine, cytosine, guanine, and thymine.

Adenine forms bonds with thymine and cytosine forms bonds with guanine. Guanine-Cytosine (G-C) bonds are stronger than Adenine-Thymine (A-T) bonds due to an extra hydrogen bond. As a result, DNA structural stability is linked with the number of existing G-C bonds [10]. RNA nucleotides are the same as DNA nucleotides apart from thymine which is replaced with uracil. RNA is less stable than DNA [10].

The process of UV disinfection starts with the absorption of UV photons by the nucleotides present in the nucleic acids, when a microorganism is irradiated with UV light [9]. The absorption of UV light causes photochemical reactions in the nucleotides. Some of the bonds between the bases in the nucleic acids may be broken and bonds between adjacent bases in the same strand may be formed. In other words, the absorption of UV light may result in the formation of intrastrand dimers, which can cause damage to the nucleic acids, if a critical number of dimers is formed. These damages may result in microbial inactivation or inability to effectively reproduce. This process is the fundamental mechanism of UV disinfection. In DNA, the most common dimers formed are thymine dimers (as a result of A-T bonds being the weakest links in the DNA structure), followed by cytosine dimers. Figure 2.2 demonstrates the formation of thymine dimers. In RNA, the predominant dimers are uracil dimers [1]. The formation of dimers (dimerization) also depends on the DNA conformation (A or B). The B conformation is fully hydrated and therefore has a higher resistance to UV.

Other photoproducts besides dimer formation can also contribute to cell death, such as photohydration. Photohydration occurs when cytosine or uracil forms a bond with elements of water molecules.

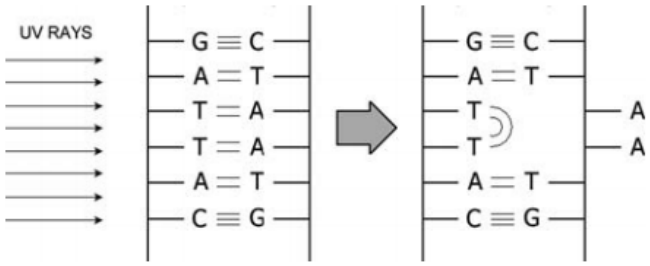


Figure 2.2: Formation of thymine dimers as a result of UV radiation absorption, in DNA [1]

The ability of UV radiation to damage DNA, either by the formation of dimers or photoproducts, depends on two factors, the DNA sequence and flexibility. Only the bases that are in a geometry capable of forming a dimer are able to photoreact [1].

The four DNA nitrogenous bases have absorption peaks within the UVC range ($200-280$ *nm*). Thymine and cytosine have strong peaks around $265nm$, which is within the peak range of UVC germicidal effectiveness ($260-265$ *nm*) [1].

Microbes have mechanisms to protect themselves from UV radiation (photoprotection), such as having UV absorbing proteins, chromophores which are also capable of absorbing UV

photons, or enzymes that repair the bonds broken and facilitate photoreactivation (theoretically, using high levels of UV exposure can cause extensive damages that inhibit photorepair mechanisms) [1]. These mechanisms impact the necessary UV exposure dose.

The germicidal effectiveness varies from species to species of microorganisms, some are more susceptible to UVC disinfection than others. According to the International Ultraviolet Association (IUVA), all microbes, including strains of coronavirus other than SARS-CoV-2, tested so far can be inactivated using an appropriate UV dose [11]. The germicidal effectiveness of UVC radiation is influenced not only by the exposure time, distance to source and type of microorganism, but also by temperature, UV intensity, wavelength, absorbing properties of the surface the microorganism is in, and the ability of UV light to reach the microbe. [12].

2.1.2 Mathematical Modelling

Disinfection is usually modelled in terms of the survival rate or inactivation rate of the microorganisms. It is a logarithmic process. Modelling this process serves as a base to projecting the disinfection system. The main model used for UVC disinfection is exponential decay and the slope of the logarithmic curve is the UV rate constant. A high value for the UV rate constant implies fast decay and low UV resistance. The UV rate constant units are m^2/J.

The degree of inactivation of a microorganism is determined by the UV exposure dose [9]. UV exposure dose is given by multiplying the exposure time to UV radiation by the irradiance. Irradiance is defined as "*The UV power incident on an infinitesimally small area dA of a flat surface from all upward directions divided by dA*" [9].

$$D = I_r E_t \qquad (J/m^2) \quad (2.1)$$

$$I_r = \frac{P}{A} \qquad (W/m^2) \quad (2.2)$$

Where D is the UV exposure dose, measured in J/m^2, E_t is the exposure time, measured in seconds, I_r is the irradiance, measured in W/m^2, P is the UV bulb power, measured in W, and A is the irradiated area, measured in m^2. The area of the irradiated surface is often considered to be the area of a spherical surface. Equation 2.1 may be directly applied when the irradiated surface is exposed, because in this case the irradiance is relatively constant.

Different models may be used for UVC disinfection, such as Single Stage decay model or Two Stage decay model. The latter is used when a small fraction of the microbial population has a higher resistance to UVC radiation than the rest of the population, meaning the UV rate constant is not the same for inactivating the different parts of the population. The single stage equation is 2.3 and the two stage decay equation is 2.4 .

$$S = e^{-kD} \qquad (2.3)$$

$$S = (1-f)e^{-k_1 D} + fe^{-k_2 D} \tag{2.4}$$

Where S is the survival of the population (a fractional value), k is the UV rate constant for the single stage decay, k_1 and k_2 are the UV rate constants of the first and second stage, respectively, for the two stage decay, and f is the UV resistant population fraction.

While building the UVC disinfection model, some effects might have to be considered, such as shoulder curves, relative humidity, photoreactivation (minimum effect for higher UV doses), temperature (may impact the UV lamp output by overheating), and others. Shoulder curves start with a horizontal slope, representing a lag in the response to UV radiation. According to Casarett 1968, this lag implies the necessity for a threshold dose or supports that repair mechanisms occur and are effective at lower doses of UV radiation [1]. As stated previously, the DNA conformation has an impact on the dimerization process, influencing the microbe resistance to UV radiation, and for this reason relative humidity effects may have to be considered.

On average, viruses are more resistant to UV radiation than bacteria, and fungal spores are considerably more resistant than bacteria. On surfaces, bacteria UV rate constants appear to be highly dependent on the type of microorganism, and viruses are shown to be five times more resistant than bacteria, however the data sets on which this information is based are limited [1]. When surface rate constants are unavailable, water rate constants can be used.

Various factors determine the resistance or sensitivity of a microorganism to UV radiation. These factors impact the value of the UV rate constant. For non-spore virus and bacteria, a few are listed bellow.

- Physical size;
- Molecular weight of the nucleic acid;
- DNA conformation;
- Percentage of G-C bonds;
- Percentage of A-T bonds;
- Percentage of potential dimers;
- Presence of chromophores, UV absorbers or repair enzymes.

An important indicator of the project of the UV system is the D_{90}. It is used to evaluate the survival rate of the microbial population. D_{90} represents the required dose to inactivate 90% of a given microbial population (survival of 10%), which represents a 90% disinfection rate. D_{90} can be calculated by equation 2.5.

$$D_{90} = \frac{-ln(1-0.9)}{k} \qquad (J/m^2) \tag{2.5}$$

As expected, a higher UV constant rate, that indicates fast decay, results in a lower D_{90} value, meaning the microbe is less resistant to UV exposure. Other common indicators are D_{99}, representing a survival rate of 1% and $D_{99.9}$, representing a survival rate of 0.1%. Using these indicators to analyse the inactivation rate of a microbial population rather than using the UV rate constant is less ambiguous when the decay curves include shoulders or more than one stage.

Table 2.1 presents values for the required dose for inactivating different strains of coronavirus, some of these values were determined in recent studies, and all of them were determined using a wavelength of 254 *nm*.

Table 2.1: Required dose to inactivate 90% of different strains of coronavirus

Microbe	**Medium**	$D_{90}\ mJ/cm^2$	**Source**
Middle East Respiratory Syndrome Coronavirus (MERS-CoV)	surface	2.81	[13] [14]
Middle East Respiratory Syndrome Coronavirus (MERS-CoV EMC 2012)	liquid	27.47	[15] [16]
Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV)	surface	1.8	[13] [14]
Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV Frankfurt 1)	liquid	17.54	[15] [17]
Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV Hanoi)	liquid	22.67	[15] [18]
Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV P9)	liquid	40.5	[15] [19]

The D_{90} values for different strains of coronavirus varies greatly, which is apparently caused by the experimental conditions selected in each study [15] such as the medium, however it is shown that it is possible to inactivate the virus. Although these results are not related to SARS-CoV-2, different strains of coronavirus share the same structural and life cycle features [20], for example SARS-CoV genome has an 80% shared identity with SARS-CoV-2 [21]. However it is important to note that though the structural proteins might be the same for different strains of coronavirus, the nucleotide sequence is not [20]. Joining this information with the information provided by recent studies, SARS-CoV-2 inactivation through UVC radiation is possible. For an inactivation of 90% of SARS-CoV-2, [15] suggests a required UVC exposure dose of 3.7 mJ/cm^2, [21] suggests a dose of 40 mJ/cm^2 for complete virus inactivation and [13] suggests a dose higher than 20 mJ/cm^2 for 99.9% of virus inactivation. In the next section, UVC effectiveness in inactivating SARS-CoV-2 will be addressed and further explored by presenting recent studies.

2.1.3 UVC inactivation effectiveness of the Covid-19 virus

UVC radiation is a widely used effective method for inactivating pathogens. All virus are susceptible to UV light, however UV sensitivity is highly variable. Previous studies on SARS-CoV demonstrate that this virus is susceptible to UV radiation, and establish a minimum value of UVC irradiation of 90 $\mu W/cm^2$ [7]. The novel coronavirus is a simple stranded RNA virus, with a

molecular structure very similar to SARS-CoV. Recent studies released have been verifying the effectiveness of UVC in inactivating SARS-CoV-2 (Covid-19 virus).

Using a low pressure mercury lamp, the most common used in UVC disinfection systems, emitting UV radiation with a wavelength of 254 nm and average irradiance of 1.082 mW/cm^2, it was found that for lower concentrations (for example, the average concentration found on hospital rooms) of SARS-CoV-2 a dose of 3.7 mJ/cm^2 would completely inactivate the virus. However, for higher concentrations (for example, the ones found in Covid-19 terminally ill patients), a dose superior to 16.9 mJ/cm^2 would be required [22]. The distance between the irradiated virus and the light source was not disclosed. Note that this study hasn't been peer reviewed and was released on June 2020.

Another study, published on July 2020, using a UV LED light of 280 nm and 3.75 mW/cm^2 placed within distance of 20 mm of the virus, found minimal cytopathic effects on irradiated cells (cells die due to inability to reproduce) after 1 second (3.75 mJ/cm^2), with a reduction of 87.4% of the virus concentration, and after 10 seconds (37.5 mJ/cm^2), with a reduction of 99.9% of the virus concentration. This was based on an in vitro experiment and the sample of SARS-CoV-2 used was isolated from an infected patient [23].

In August 2020, another article supporting the high susceptibility of Covid-19 to UV light was published [24]. In this study, a sample obtained from an infected patient was irradiated with UVC and UVA light, which was proven to be much less effective as expected. The UVC light intensity used was 1.940 mW/cm^2 and a wavelength of 254 nm. The distance from the light source to the irradiated virus was 3 cm. After 9 minutes of exposure, the virus was completely inactivated, corresponding to a required UV dose for inactivation of 1048 mJ/cm^2. This study concludes that the use of UVC irradiation to inactivate the novel coronavirus is a reliable method not only for health care facilities disinfection but also for preparing inactivated SARS-CoV-2 material for research purposes, because it was found to be an effective method even when inactivating higher viral load levels that are found in research laboratories. It is important to note that the methods used in the experiments from the different studies are quite different, which may contribute to the difference between the required doses presented.

On December 2020 another article was released stating that the use of UVC radiation allowed for a rapid and complete inactivation of the virus. This study analysed the effects of UVC light on wet and dried droplets of SARS-CoV-2 using a mercury lamp with a peak wavelength of 254 nm and irradiance of 0.849 mW/cm^2. For the dried virus, complete inactivation was achieved in 9 seconds, and for wet virus in 4 seconds. A reference to another study that used a wavelength of 280 nm and wet droplets and inactivated the virus in 10 seconds using a UV dose of 3.75 mW/cm^2 is made. To conclude, the article declares the need for further studies on the required UV exposure dose to inactivate SARS-CoV-2 that explore effects such as humidity and surface design (surfaces may have crevices that are not reached by the UV rays) [25].

Most of the studies performed use lamps that emit light with a peak wavelength of 254 nm, which is also the most common used in UVC disinfection systems. A few studies suggest that using 222 nm has the same efficiency as 254 nm and has the advantage of being less harmful to

the skin and eyes [26].

To summarize, recently many studies have emerged suggesting the effectiveness of UVC in inactivating the novel coronavirus and supporting the use of UVC disinfection systems to combat the spread of the virus and help control the current pandemic. However, the data available is still limited and there is considerable variability in the reported required UV dose. This variability might be attributed to deficiencies in the experimental methods used and some important details of the experiments were not disclosed [8]. It is also important to note that all the presented studies were performed in a laboratory environment and further experiments must be executed in real world environments. Nonetheless, based on available evidence, UV radiation should be effective in inactivating the SARS-CoV-2 as well as other microorganisms.

2.1.4 Advantages and disadvantages

UVC radiation is highly effective in inactivating microorganisms and presents several advantages that justify its use in disinfection systems. One key advantage is its high effectiveness in inactivating microbes, which contributes to the prevention of infectious diseases caused by viruses or bacteria and reduces their transmission. Given the current world pandemic caused by the novel coronavirus, this is an extremely important characteristic.

Another important advantage is that UVC radiation immediately dissipates, which means that it is safe to enter a room that was being disinfected immediately after the disinfection process ends. This is extremely important in the case when a person accidentally enters a room currently being disinfected (admitting the UVC disinfection system detects the person and automatically turns off).

The traditional process of disinfecting rooms and equipment is by chemical disinfection. Chemical disinfection exposes professionals to corrosive and toxic agents that are prejudicial to human health. UVC disinfection does not require the use or transportation of any toxic or corrosive agents, and although disinfection by UVC irradiation might be more efficient if followed by traditional disinfection processes, UVC disinfection still reduces or eliminates the exposure of health care workers to harmful agents. Also, UVC disinfection does not generate chemical residues or by-products [27].

Another advantage is that it does not require contact with infected surfaces, which contributes to the reduction of disease transmission. Surface disinfection is advantageous when compared to other mediums because it attacks microbial growth at the source [1]. It is important to reference that contaminated surfaces are often a source for airborne microbes.

UVC disinfection is easy to use and requires low maintenance (although maintenance also depends on the type of UVC disinfection system). Finally, the disinfection process is relatively fast, in the range of seconds [9] (when the surfaces are very close to the light source, as they were in the presented studies).

Using UVC for disinfection accommodates many advantages as demonstrated, however it is important to state the disadvantages of using UV radiation and UVC disinfection systems. Firstly,

the major disadvantage of UV radiation is the threat it poses to human health. Unprotected exposure to UVC radiation is harmful to the human health even if tolerated during minor periods of time. A few of the negative effects of unprotected UVC exposure are eye damage, skin burns, and even potentially skin cancer. Out of these, the most serious effect is eye damage. Prolonged UV exposure may cause photokeratitis, conjunctivitis, corneal injuries, and even cataracts [1]. Regarding skin damage, some of the effects include erythema, photosensitivity, skin aging, immune system damage, and as stated before skin cancer [1]. Smaller wavelengths, such as 222 *nm*, are not as harmful to human health and do not cause the presented damages. UVC radiation is invisible to the human eye, which can facilitate the occurrence of unprotected and prolonged unprotected exposure.

Another possible disadvantage appears when using medium or low-pressure mercury UV lamps, which are commonly used in UVC disinfection systems. The disadvantage associated with the use of these lamps is mercury. The breakage of a lamp, results in a potential mercury hazard [9]. If the breakage occurs during lamp usage, then the mercury should be in vapor form, which constitutes an inhalation hazard. In addition, these lamps have low mechanical stability, and some may produce ozone [28]. UV-LED might be a better substitute.

Other disadvantages include the variability of the effect of UV radiation from species to species and its dependence on the growth medium, and characteristics of the microorganism such as strain and density, among others.

Lastly, an important disadvantage is that UVC disinfection is only effective when UV radiation is applied directly at the target, which means the surface design, shadows, pores, and orifices affect the germicidal effectiveness of UVC.

2.1.5 Safety

Several disadvantages of UVC disinfection involving threats to the human health were previously presented. UV radiation with wavelengths within the range between 180 *nm* and 320 *nm* constitutes a concern for biological safety and health [1]. This range includes UVB and UVC. For this reason, safety is an important aspect to take in consideration when projecting and handling a UVC disinfection system. There are three types of safety measures to be taken, namely engineering controls, operational protocols, and personal protective equipment [1].

Unprotected exposure to UVC radiation is tolerated during short periods of time, however it is to be avoided. Figure 2.3 presents values for the limit of exposure time to UVC radiation. The damages caused by unprotected UV exposure target mainly the eyes and the skin and its effects may take several hours, 1 h to 12 h [29], to be noticeable. Many of the ocular damages are not permanent and may disappear within a few hours or days. The threshold dose for keratoconjunctivitis is 40 J/m^2 [1]. Regarding skin damage, UVB is more harmful than UVC, however the areas of the environment subjected to any UV exposure should be properly labelled and warning signs should exist [29]. For unprotected exposure to UV radiation with wavelengths between 180 *nm* and 400 *nm*, the UV exposure dose should not exceed 30 J/m^2 [1].

When using mercury UV lamps, very high temperatures may be reached during operation, which implies a risk of burns. For this reason, the lamps should be allowed to completely cool down before maintenance [9].

Duration of exposure per day	Effective irradiance	
	E_{eff} (W m^{-2})	E_{eff} (μW cm^{-2})
8 h	0.001	0.1
4 h	0.002	0.2
2 h	0.004	0.4
1 h	0.008	0.8
30 min	0.017	1.7
15 min	0.033	3.3
10 min	0.05	5
5 min	0.1	10
1 min	0.5	50
30 s	1.0	100
10 s	3.0	300
1 s	30	3,000
0.5 s	60	6,000
0.1 s	300	30,000

Figure 2.3: Limit UV exposure time. Source: ICNIR, International Comission on Non-Ionizing Radiation Protection [2]

2.1.6 Existing solutions

Various systems based on UVC disinfection principles are already in use, including in hospitals or other health care facilities. In health care units, these systems are useful for disinfecting material, personnel equipment and areas. Some of these systems include lamps, fixtures, ballasts, and even reflective surfaces to improve the UVC effectiveness by enhancing the irradiance field and exposing otherwise shadowed areas to UV [1]. The most common material used for this purpose is polished aluminium, and it has a UV reflectivity of 75% [1]. These systems may be lamp fixtures mounted on walls or ceilings, or be portable, either manually or autonomously. Some of these systems used for air or surface disinfection are presented on figure 2.4.

To disinfect medical equipment, vials, or surgery tools a UV cabinet can be used, it disinfects the materials within 30 to 60 minutes. It must be used in combination with chemical disinfection, because it has been shown that a residual level of contamination is typically still present after UV disinfection when using a UV cabinet [1].

Regarding some of the systems presented in figure 2.4, cooling coil disinfection is among the most successful applications of UVC disinfection, because it can save energy. Lower room disinfection systems are mainly used to disinfect the floors, since bacteria and other microorganisms tend to gravitate towards the floors. Overhead surgical site disinfection systems should have two operation modes, one for disinfecting the room when it is unoccupied and one for when it is occupied. UV absorbers can be used to control the UV irradiance of lower and overhead systems

Classification	System
Air disinfection	In-duct air disinfection
	Recirculation units
	Upper room systems
	UV barrier systems
	Overhead tank disinfection
Surface disinfection	Equipment & packaging disinfection
	Cooling coil disinfection
	Lower room disinfection
	Overhead surgical site disinfection
	Area/Room disinfection
	Food surface disinfection
	Hand wands
	Mold remediation systems

Figure 2.4: UVC based disinfection systems for air or surfaces [1]

to minimise UV exposure. Area disinfection systems can be portable or permanent and may be turned on and off automatically using timers, for example, to disinfect the area in specific times of the day. These systems emit very high levels of UV irradiance. Health care wands can be used in health care settings or even for a home use.

A wide variety of solutions for portable UVC systems already exists in the market. Many are hand wands, that can be used by an untrained user at home or for disinfecting a patient's personal items in a hospital, for example. More recently, other solutions for disinfecting rooms and surfaces have arisen. One example of these solutions is THOR UVC by FINSENTECH [1].

THOR UVC is a system for disinfecting air and surfaces using UVC radiation. It is not autonomous, which means it requires human interference to be moved from location to location and placed in the desired space. It has a unique patented design that reduces shadowing, which improves UVC effectiveness; it has four motion sensors that turn off the lights if any motion is detected (this is a good health protection feature); it has an adjustable height, from 1.1 m to 2.25 m, which allows the system to reach more areas of the room and gives this system an advantage when comparing to other existing solutions; it has six castor wheels to facilitate movement; it is not wireless; it auto generates cleaning reports; it claims that inactivates bacteria within minutes and that a daily use of this technology reduces the acquisition of hospital-acquired infections by 37%. It is important to note that the values presented are regarding bacteria inactivation and not general microbial inactivation.

Another portable solution is described in an article from 2014 [30]. This article presents an automated mobile unit that uses UVC light with a wavelength of 254 nm and studies its effectiveness in disinfecting patients' rooms after the patients were discharged and the room cleaned by conventional means. This unit allows an operator to set the UV exposure dose to be applied and it must be manually placed in the desired space. The UV system terminates the disinfection process

[1] https://www.finsentech.com/products/thor-uvc/

when all the eight sensors that detect how much UV light bounces back from the surrounding surfaces, have determined that the required UV dose has been applied to the surfaces in the room. To verify the systems effectiveness, two experiments were conducted. The first one placed the unit in the centre of the patients room with the bathroom door open, and the second one placed the unit in the centre of the patients room with the bathroom door closed and then placed it in closer location to the bathroom and opened the bathroom door. The first experiment required 67.8 minutes to disinfect the room, and the second required 83.7 minutes. These experiments were focused on the inactivation of *Clostridium difficile* (a bacteria) spores. This study concluded that the portable UV light system was able to reduce colony counts to the minimum on the surfaces that were directly exposed to UVC light and significantly reduced the colony counts on shadowed areas.

Recently, alternative and innovative UVC disinfection systems have been developed, such as mobile autonomous robots. A few of these solutions will be presented.

On September 2020, an article regarding an UV disinfection robot was published [4]. This robot follows a pre-defined path, has three 20 W UVC lamps, and has PIR sensors on top of the robot for detecting humans or potentially animals. Upon human or animal detection, the robot automatically turns off the UVC lights and on again once nothing is detected. The article recognizes the height of the robot as a limitation, since the UV radiation can only reach the areas up to the robot's height. This robot is able to inactivate the coronavirus in 57 seconds or more.

On October 2020, another article was published also presenting and autonomous mobile robot using UVC disinfection, the i-Robot UVC [28]. This robot has eight vertical UVC LED lamps and two more on the top, it has temperature, humidity, position, and motion detection sensors, it automatically turns off the lights upon human detection, and it can inactivate 99.999% of bacteria. The robot navigates around the room to be disinfected to minimise the distance between the robot and the surfaces and to minimise shadowing. I-Robot UVC can be manually controlled by an operator through a dashboard or it can function automatically, it has a LiDAR sensor that scans the environment to create a digital map.

Some of the most popular autonomous mobile robots are UVD Robots [2]. UVD Robots started developing autonomous mobile UVC disinfecting robots in 2016 and has currently two models, B and C. The main motivation for developing these robots are hospital-acquired infections and the extra costs they can bring. Depending on exposure time, these robots can inactivate 99.99% of pathogens. The robots are autonomous but can also be driven from a distance via their tablets; the operation starts with an operator sending the robot to a desired location via its tablet; the model C of the robot is able to create a heat map and an exposure map; for safety, the robots are equipped with sensors that detect when someone enters a room and the robot automatically turns off the lights; also for safety, if the robot detects an object above 15 cm, a person or a wall that is too close, it stops; it comes with a docking station for recharging, which the robot finds autonomously.

Another autonomous disinfecting robot has been developed by INESC TEC since 2020, the

[2] https://www.uvd-robots.com/about-us

RADAR project [3]. This robot navigates through the environment autonomously while disinfecting the area uninterruptedly and claims to disinfect surfaces and air up to 99.9%. This robot has already been tested in a hospital environment. The techonology developed by INESC TEC was licensed to DCSI PRO, that launched a first version of the autonomous robot called UV-C Robot [4]. This robot should disinfect an area of 25 m^2 in 4 minutes, is equipped with 900 W UVC lamps, is able to detect people and obstacles, has an autonomy of 2 hours, and has already been tested in an airport.

A few other existing solutions for UVC disinfection mobile robots are worth referencing, such as Sunburst UV Bot, LD UVC by Omron and ZenZoe. Sunburst UV Bots [5] also has safety mechanisms that detect when people are close and turn off the lights autonomously; it eliminates 99.99% of virus and bacteria; it can be programmed to follow a specific route and to disinfect a specific area; it is meant to be used in hospitals, commercial spaces, and others. LD UVC [6] has aluminium reflectors to reach shadowed areas and emits a voice message and turns on a buzzer to indicate that the disinfection process will begin and the UVC lights will be turned on. ZenZoe [7] offers three types of manual portable devices and one autonomous mobile robot for disinfecting areas using UVC light that uses real-time free SLAM navigation technology. ZenZoe solutions can inactive up to 99.99% of pathogens, including SARS-CoV-2, and is able to disinfect surfaces in a single pass. For a room of 25 m^2, it requires 8 minutes to disinfect. A few articles studying autonomous mobile robots solutions and verifying their effectiveness have also been published [31] [32].

2.2 Mapping

The main objective of this dissertation is to develop the mapping module of a mobile autonomous robot that uses UVC light to perform disinfection. The principles and existing solutions for UVC disinfection have already been presented. The robot must move through the environment autonomously, which means it has to be able to localize itself. In addition, the robot must build a map of the environment in order to calculate and apply the necessary UV dose to each area and surface present in the environment the robot is in. Because map building is a crucial aspect of the development of the mapping module, it is important to have an overview of the existing mapping algorithms, especially 3D mapping algorithms/techniques. The robot must build a 3D map of the environment because it has to disinfect surfaces with different heights and shapes, and not only circulate around the room. The map of the environment built by the robot will be used to estimate the required UVC dose to be applied to each point in the map. The robot will be used indoors, so the explored algorithms are focused on indoor applications.

[3] https://www.science4covid19.pt/radar-robo-autonomo-para-desinfecao-em-ambiente-hospitalar/
[4] https://www.m-tec.pt/2020/12/04/uv-c-robot-presentation-oporto-airport/?lang=en
[5] https://kazerobotics.com/products/sunburst-uv-bot/
[6] https://web.omron-ap.com/sg/ld-uvc/
[7] https://zenzoe.tech/robot/

In the following sections, an overview of mapping concepts and mapping algorithms will be presented (Subsection 2.2.1). 3D mapping techniques will be focused on (Subsection 2.2.2) and 3D mapping techniques available in ROS will be briefly presented (Subsection 2.2.2.1).

2.2.1 Fundamental concepts

A very important task of a mobile robot is to build a map of its surrounding environment. An extensive knowledge of the environment, indoor or outdoor, is required when the robot is involved in sophisticated manipulations of its environment [33]. Map building is essential for the navigation competence of a robot, because without a map the robot cannot localize itself in the environment.

Navigation requires the success of perception, cognition, motion control and localization [34]. Localization allows the robot to determine its position in the environment. To do this, the robot must build a map and then identify its position relative to the map built. A navigation map-based approach includes localization and cognition modules [34].

Map building may be done manually and not by the robot; however, this is a very difficult task with several disadvantages: it requires exact measures [33] and the map is dependent on the perception of the map builder [34]. The better alternative to manual map building is automatic, where the robot builds the map of its surrounding environment other than being given the map already built. The logical step after mobile robots appeared, was to supply them with sensors such as distance sensors to enable automatic map building by providing the necessary data [33]. Because of the sensors' limited range, the robot has to move around the environment while building the map, in other words, the robot has to be able to localize itself while moving and building the map [34]. The Simultaneous Localization and Mapping (SLAM) class of algorithms are used to solve the problem of concurrent mapping and localization. At a fundamental level, this problem can be viewed as a maximum likelihood estimation problem and solved by finding the most likely map [35]. Sensors are also accompanied by uncertainty because they are cannot provide completely accurate data. To deal with these uncertainties, most algorithms used are based on probabilities to describe the robot's position and environment [36].

Real-world environments are dynamic and not static. Many existing map representations do not differentiate between the permanent and transient obstacles that exist in a dynamic environment [34]. Permanent obstacles can be walls and transient obstacles can be people. A mobile robot operating in a real-world environment should be able to distinguish between these different types of obstacles. In a static environment, the robot explores unknown areas of the environment, and creates a spatial representation (map) with the collected data. This type of representation only distinguishes between three types of areas: occupied, unoccupied and unexplored areas [36]. Common mapping techniques for this type of representation are occupancy grid, geometric maps and landmark representation [36]. However, the algorithms used for static representations are not able to provide an accurate representation of dynamic environments, because areas with moving obstacles will possibly be wrongly classified as occupied or free areas [36]. In a dynamic environment, obstacles are classified as static, low dynamic (for example, doors) or high dynamic (for example, people). Because the desired complete mobile robot will disinfect the environment using

UVC radiation, people are not expected to be present, which means that the desired disinfection system will likely be used in static environments.

The goal of map building is to explore unknown regions and build a spatial model of the environment using sensors, being the most popular cameras and laser range finders [37]. The approaches that use cameras and laser range finders, vision and laser range finding respectively, have a few problems worth mentioning. Vision is often computationally expensive and is sensitive to illumination variations [38]. 2D laser range finders have limitations that only allow building 2D maps [38], which are not as representative of the real world as a 3D map would be. However, this last problem has been mitigated by placing the laser scanner on a rotating axis and expanding the distance measurements into 3D [33], for example, or by using 3D laser range finders.

3D mapping allows the creation of more complex and closer to the reality maps [33]. They provide a volumetric representation of complex environments which is important for a number of robotic applications, such as aerial robots and robots with manipulators [3]. 3D mapping has not been has extensively studied as 2D mapping, and few available efficient implementations exist [3].

Building a 3D map is much more complicated than building a 2D map, because the introduction of a new dimension is accompanied by new degrees of freedom in the movement of the robot [33]. More degrees of freedom increase the complexity of estimating the localization of the robot with some level of accuracy. Besides the added complexity, 3D map building is also associated with a slower scan acquisition time and the geometric increase in data to be processed [38], which are disadvantages because autonomous navigation may require real-time data [39].

Initially, 3D mapping used laser range finders and stereo cameras and focused on 3D reconstruction of the unknown environment [40]. Sensors used for acquiring data for 3D mapping are 2D laser scanners placed on rotating axis, CCD (charge-coupled device) array cameras, triangulation-based camera-beam systems, and time of flight cameras [33]. Another sensor used can be a 3D laser range finder. More recently, RGB-D cameras, that provide visual and depth information, have been used to collect data for building indoor 3D maps [37].

For creating 3D maps, three requirements should be satisfied: probabilistic representation (consider the uncertainties associated with measurements obtained with sensors), representation of unmapped areas, and efficiency (regarding time and memory consumption, which is usually the biggest problem with 3D mapping) [3]. To effectively explore an unknown environment and build its spatial 3D representation, an efficient mapping algorithm should be used, the robot should be able to identify the new areas that need to be covered, and the robot should be able to plan an efficient path to collect data to build the map [37].

As stated previously, 3D mapping algorithms are the mapping techniques of interest to develop the mapping module for the desired autonomous mobile robot that uses UVC to accomplish disinfection.

2.2.2 3D Mapping Techniques

There are many existing mapping techniques for building a spatial representation of the environment, in 2D or 3D. The maps built can be categorized in three types: grid maps, feature maps and

topometric maps [41]. In a grid representation, the environment is divided forming cells (forming an equidistant grid) [42]. The resolution of this type of representation depends on the cell size, the resolution of the grid map limits the environmental representation and greatly impacts memory consumption. However, grid maps are the best choice when unstructured objects have to be mapped [41]. The most common shape for a grid map is square in 2D or cube in 3D [42] (it can be a three-dimensional array where each cell contains the probability of the robot being in the cell [34]). A feature map represents obstacles using geometry descriptions, such as lines, planes, or cubes [41]. Topometric maps are a topological representation (which represents the environments using graphs, where nodes can be places or landmarks and arcs connecting the nodes are direct paths between nodes [42]) with a direct spatial relation. When no spatial relationship exists, a topological map is achieved. All of these types of maps have advantages and disadvantages, however the most commonly used are grid maps.

A grid map is the most adequate environment representation to be used for the UVC dose mapping module of the mobile robot. The robot should build a map of the environment and calculate and update the required UVC dose to be applied for each cell of the grid map. For this reason, grid map representations will be the focus for the remainder of this section.

One of the most used techniques for map building is occupancy grid mapping. In an occupancy grid map, each cell is classified as being filled, if it is part of an obstacle/occupied space, or as being empty, if it is part of free and unoccupied space [34]. It has been used for localization, mapping, navigation, and safe path traversal, using 2D or 3D representations [41]. This technique is efficient when representing uncertainty, fusing multiple sensor measurements, and incorporating different sensor models [41].

The following four systems are all based on occupancy grip mapping and presented in [43]. The Robot Localization and Mapping technique using Sonar data ([44]), creates the map by collecting and processing measurements acquired with a sonar sensor that is rotated by a servo motor. The grid used is 2D, and the occupancy and certainty value obtained for each cell of the grid is then used for the robot's localization. Occupancy Algorithm is the second robot mapping technique presented in [43]. In this algorithm, the current location of the robot is combined with the sensor readings to determine the state of each grid: occupied, unoccupied, or unexplored. The occupancy of each cell is updated using the previous value and the value obtained by the sensor readings. Autonomous topological modelling of an environment ([45]) is the third algorithm presented. It uses SONAR and although it creates a grid map, a topological model of the environment is extracted from the grid representation, using cell decomposition to extract empty regions and normalised graph cut. Finally, Robot Map Creation Algorithm ([46]) using sensor data is the fourth algorithm presented. In this algorithm the robot is able to create the map while moving and localizing itself. The algorithm begins with the creation of a polar distance map using sonar sensor readings. While exploring the environment and building the map, the robot moves in several directions and stops to acquire new data that is incorporated in the existing map. The most likely location of the robot is extracted and then based on this, the sensor readings are shifted, and finally the map is modified. Many other grid-based mapping algorithms and techniques besides

these four presented exist and have been presented and reviewed in several articles.

Focusing on mapping techniques to build models of 3D environments, a few techniques that use RGB-D sensors are RGBDSLAM, SLAM6D and KinectFusion [47]. Out of these, KinectFusion is the most accurate and has a real-time performance, which may be a requirement in map building. KinectFusion can only build a limited sized map because otherwise it would be a memory costly method and is not very stable. An advanced KinectFusion method was developed and presented in [47]. This method adds two improvements to the KinectFusion algorithm, decreasing the modelling error from 4.5% to 1.5% [47].

Other three common mapping approaches are point clouds, elevations maps, multi-level surface maps and a volumetric approach.

Point clouds have been used in several 3D SLAM systems, they do not require the discretization of data, and the mapped area is not limited. However, because they store large amounts of measurements, they are not memory efficient (the memory consumption increases with the number of measurements). There is also no differentiation between free and unexplored areas of the environment and for this reason point clouds should only be used when the environment is static, high precision sensors are used and unknown areas do not need to be represented [3].

Elevation maps are efficient, but do not represent unexplored areas and cannot represent arbitrary 3D environments. Elevation maps are based on a 2D grid that stores an estimated height for each cell. 2.5D elevation maps use a 2D grid to store the height information for each cell, exactly one value is stored per cell [3]. One disadvantage is that it only discretizes the vertical dimension of the environment (only one level is represented), so this type of map is not sufficient to model multilevel environments [48]. Although an elevation map reduces memory consumption, it is not sufficient when the robot has to use the elevation map for navigation or has to integrate two different maps. Elevation maps and extensions of elevation maps still can only be applied to single surface environments [49].

Multi-level surface (MLS) maps are an extension of elevation maps that allow the modelling of environments with more than one level. These maps consist of a 2D grid of variable size, where each grid cell stores a list of surface patches. Each patch consists of the height (of the position of the cell in the map) mean, height variance and depth value [49]. A MLS map can be created using 3D measurements or by joining two other MLS maps [49]. Whenever a new measurement is inserted into the map, the map is updated [49]. When compared to other representations, MLS has the advantages of being able to represent multiple surfaces per cell and allowing mobile robots to operate in environments with multiple levels, but has the disadvantages of not being able to represent unknown areas, not being as simple, and not being a volumetric representation and instead a discretization in the vertical dimension.

As stated previously, an accurate representation of the environment is required when the mobile robot has to navigate in unknown environments that contain multiple surface levels [48]. This accuracy can be achieved using a 3D volumetric representation. Voxels approach can be used to discretize the mapped area. Voxels use a grid of cubic volumes with equal size [48]. This approach requires a large amount of memory and the knowledge of the extent of the area to be

mapped [3]. A more effective solution that requires less memory is the use of octrees. Another 3D mapping method, considered a key contribution to 3D mapping [40], is OctoMap. OctoMap uses octrees.

Octrees are a tree-based data structure that recursively subdivides the three-dimensional space into octants. With octrees the initialization of map volumes can be delayed until the measurements have to be integrated and the resulting map will only contain measured volumes [3]. Octrees allow for complete 3D models of the environment, are probabilistic, memory efficient (because the recursive process can be stopped when the desired voxel size is reached), and are even multi-resolution (the tree can be cut at any level to obtain a subdivision) [3], however it is more complicated to implement.

OctoMap is a 3D mapping technique based on octrees. It creates a volumetric 3D model of the environment based on probabilistic representation of occupancy [48]. It is an open-source framework, it supports multi-resolution map queries [3], it is able to perform lossless map compression [3], it is efficient and compact in memory [48]. It is compact in memory because mapping free space takes up less memory due to the tree structure used [50]. OctoMap satisfies the three requirements presented in the previous section and is easily integrated into an existing stack [50]. A more detailed description of this method can be found in [3]. Figure 2.5 presents a representation of an environment obtained using OctoMap, as well as other representations obtained with the methods previously mentioned.

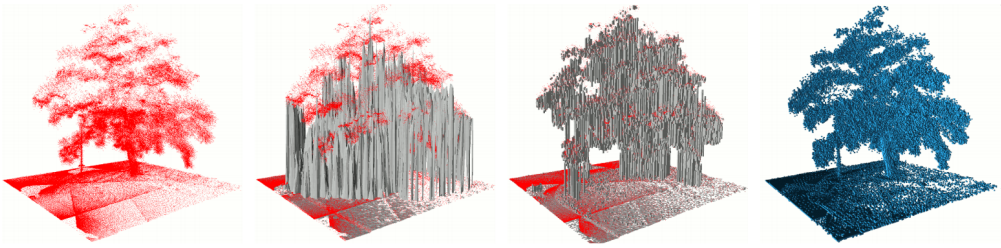


Figure 2.5: 3D representations of a tree using a laser range sensor (from left to right): Point Clouds, Elevation map, MLS map, OctoMap (only the occupied volumes are represented) [3]

2.2.2.1 3D Mapping Techniques in ROS

Robot Operating System [8] [9] (ROS) is a robotics software framework that encourages collaborative robotics software development. It is a collection of tools, libraries, and conventions that aim to simplify the development of complex robotic systems. ROS is also open-source. ROS will be the primary tool used in the development of the proposed UVC dose mapping algorithm.

[8] https://www.ros.org/about-ros/
[9] http://wiki.ros.org/

Regarding mapping, ROS already provides several open source algorithms, such as RTAB-Map and Hector Mapping. Regarding 3D mapping, ROS also supports several different systems, such as RGBD-SLAM, OctoMap, ORB-SLAM2, LSD-SLAM, ETHZ ASL ICP Mapping, Carthographer, HDL Graph SLAM, LOAM, or DSO [40].

An article from 2017 [50] compared the performance of three different open-source 3D Occupancy Mapping libraries available in ROS. These are OctoMap, SkiMap and OpenVDB. All three of these algorithms satisfy the common requirements of being able to set, store and retrieve data while not wasting memory and of allowing the insertion of data at arbitrary positions (because the size of the environment is initially unknown) without resizing or performing costly copies of data. SkiMap aims to be a faster alternative to OctoMap and uses skiplists. Skiplists are a datastructure that combine advantages of linked lists with the advantages of arrays and with the advantages of hash tables. OpenVDB is not widely used in robotics and was not developed for robotics, unlike OctoMap and SkiMap. However, the author found it to be ideal for 3D occupancy mapping because it is "*a hierarchical data structure and a suite of tools for the efficient manipulation of sparse, possibly time-varying, volumetric data discretized on a three-dimensional grid*". To compare the three algorithms, a RGB-D dataset was used. OpenVDB proved to be much faster than the other two algorithms, and OctoMap was slightly faster than SkiMap, even though the objective of SkiMap is to be a faster alternative to OctoMap.

Chapter 3

UVC Dose Map

As stated previously, this dissertation focuses on the development and implementation of a method to update the UVC dose value to be applied to surfaces in an unknown indoor environment. On the previous section, an introduction to UVC disinfection was made and several mapping solutions were presented. The chosen mapping solution that serves as the base for the system developed is the OctoMap algorithm, which builds a 3D occupancy grid map.

This chapter presents the development and implementation of the UVC dose mapping algorithm as well as an overview of the technologies used. Firstly, in Section 3.1, the technologies used for the development of this algorithm are presented, namely the recent SSL_SLAM algorithm that is based on OctoMap, the Intel RealSense Camera L515, which is the high-resolution depth camera used for capturing the surrounding environment, and the Robot Operation System (ROS). In Section 3.2, the algorithm implemented is presented and thoroughly explained. The Robot Operating System was the main tool for developing and testing the UVC dose mapping algorithm. To visualize the results of the test performed using this tool and the progress of the map building and UVC dose updating, a visualization algorithm was created and is presented in Section 3.3.

3.1 Technologies used

To achieve the objective of building a UVC dose map, the environment that surrounds the robot was divided into cells with adjustable size (in order to obtain more or less detailed maps) that have a UVC dose value associated. A suitable approach to the problem of building a 3D grid map is OctoMap, an algorithm that uses an octree to store the map built. However, OctoMap requires an external localization algorithm. SSL_SLAM [51] (Solid-State LiDAR Simultaneous Localization and Mapping) is a recent algorithm that incorporates localization and map building modules based on OctoMap. As the name suggests, SSL_SLAM was optimized for solid-state LiDAR sensors, such as the Intel RealSense LiDAR Camera L515. Systems using a SLAM algorithm and LiDAR sensor have the advantage of not requiring landmarks for localization in indoor or outdoor environments. L515 is considered a solid-state LiDAR because it has no mechanical moving parts, and because of its compact size, high resolution and optimization for indoor environments,

this depth camera presents itself as a good choice for the desired system. The main tool used to develop, test, and visualize the results of the algorithm created was ROS. In this chapter, the algorithm SSL_SLAM and sensor L515 used to build the required 3D map are presented and briefly explained. A brief introduction to ROS was previously given, however in this section relevant features of the Robot Operating System are better explained.

3.1.1 SSL_SLAM method

SSL_SLAM [51] is a simultaneous localization and mapping (SLAM) method proposed earlier this year by Han Wang and Chen Wang and was inspired by LOAM and OctoMap. The motivation for the development of this method was the necessity for optimized SLAM algorithms for solid-state LiDAR sensors, since many of the already existing SLAM algorithms are not capable of having real-time performance when using these sensors and are optimized for mechanical LiDAR sensors, which usually have a higher cost, superior size and weight. Solid-state LiDAR sensors are more cost effective, lighter (which makes then suitable for smaller robots), have higher update rate and angular resolution, but a smaller field of view [51].

The SSL_SLAM algorithm [1] is composed of three modules: feature extraction, odometry estimation and, finally, probability map building. In this algorithm there are two types of maps, the local map which is temporary and considers the points captured by the sensor in each moment, and the global map which represents the entire surrounding environment that has been explored and is built using the local map.

The first module is the feature extraction, which is rotation invariant and based on edge and planar matching. In this module, firstly, the points captured that are classified as noise are removed and not considered. The classification of noise is based on the measured distance of the points, for example, the points that are nearer the sensor's maximum detected distance are considered noise. Secondly, the point cloud captured by the sensor is projected into a 2D matrix. To achieve this, for each point (p_i) of each scan made by the sensor (P_k), the horizontal and vertical angles of the point are calculated. Then, the point cloud is segmented into cells whose dimensions are equal to half the total number of points in the vertical direction and to half the total number of points in the horizontal direction. After segmenting the point cloud, a mean measurement ($p_k^{(m,n)}$, where (m,n) are the cell coordinates in the 2D matrix) is calculated for each cell using the geometrical centre of every point inside that cell and then the local smoothness ($\sigma_k^{(m,n)}$) is calculated. The local smoothness indicates the sharpness of a surface, which means that a higher value indicates that the local surface is sharp and therefore the corresponding point is considered an edge point. Depending on the value of the local smoothness, each $p_k^{(m,n)}$ is either classified as an edge (higher value for the local smoothness) or a plane feature (lower value for the local smoothness).

$$\sigma_k^{(m,n)} = \frac{1}{\lambda^2} \sum_{p_k^{(i,j)} \in S_k^{(m,n)}} (||p_k^{(i,j)}|| - ||p_k^{(m,n)}||) \tag{3.1}$$

[1] https://github.com/wh200720041/ssl_slam

$$S_k^{(m,n)} = \{p_k^{(i,j)} | i \in [m-\lambda, m+\lambda], j \in [n-\lambda, n+\lambda]\} \tag{3.2}$$

In equations 3.1 and 3.2, λ is the searching radius for the nearby points of each $p_k^{(m,n)}$, and $S_k^{(m,n)}$ is the group of points that are inside the searching radius.

The second module is the odometry estimation. The objective of this module is to estimate the current pose of the robot based on the sensor's scans. For each received scan, a temporary local map, that divides into a edge and a planar map, is defined. For each edge point in the feature map previously found with the feature extraction module, its transform to the local map is calculated. Using the edge point and its transform, the two closest edge points to them in the local map are found and used to calculate the point-to-edge residual. The same process occurs with the plane points from the feature map, however there are found three points in the local map instead of two. These three planar points are used to calculate the point-to-plane residual. The odometry estimation is performed by minimizing these two residuals, a non-linear optimization problem. Algorithm 1 presents the summarized algorithm for this module, including the approach for solving the optimization problem.

Algorithm 1: Pose estimation for Solid-state LiDAR

Input : Current scan P_k, the robot trajectory $T_{1:k-1}$
Output: Current pose T_k

if *Not Initialized* **then**
  $T_0 \leftarrow 0$;
end
Calculate the local smoothness and extract edge features and planar features;
Compute initial alignment $T_k^0 \leftarrow T_{k-1}\ T_{k-2}^{-1}\ T_{k-1}$;
for *Pose optimization is not converged* **do**
  for *each point* $p_k \in P_k$ **do**
    Transform to map coordinate $\hat{p}_k \leftarrow T_k^{i-1}\ p_k$;
    if p_k *is an edge feature* **then**
      Compute edge residual f_ε and Jacobian matrix J_ε;
    end
    if p_k *is a planar feature* **then**
      Compute planar residual f_S and Jacobian matrix J_S;
    end
  end
  if *nonlinear optimization converges* **then**
    Compute T_k^i;
  end
end
Update current scan into local map and remove oldest scan from local map;
Return current pose T_k;

Finally, the third module is the probability map construction. The global map is only updated according to key frames instead of being updated every frame in order to increase efficiency, and an octree data structure is used to save the map. Using an octree also increases efficiency when searching, which is important because before inserting a new node it is necessary to check if that node has not yet been inserted, meaning the octree must be searched whenever the map has to be expanded. The key frames referred, which determine if the global map has to be updated, are the following: if the displacement of the robot is higher than a threshold value; if the variation of the rotation angle is higher than a threshold value; if the elapsed time since the last update is superior to a certain value that is defined by the processor used. These thresholds are defined based on the sensor's field of view. As noted previously, the construction of the map is based on the OctoMap algorithm. Both SSL_SLAM and OctoMap use an octree to store the map and each node of the octree represents either an occupied or free space (cell) in the environment that is being explored. The resolution of the map can be altered by changing the desired size of the cells that integrate the 3D map. To each node in the octree a probabilistic value, that represents the occupation probability of the corresponding cell, is associated. This probability is given by equation 3.3 (from [51], and in more detail [3]), where n is the octree depth, z_t is the current measurement, $z_{1:t-1}$ are previous measures, $P(n|z_{1:t})$ is the probability of node n being occupied, $P(n|z_t)$ is the occupancy probability of n given measurement z_t, $P(n|z_{1:t-1})$ is the previous estimate, and $P(n)$ is a prior probability (in this case 0.5).

$$P(n|z_{1:t}) = [1 + \frac{1-P(n|z_t)}{P(n|z_t)} \frac{1-P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1-P(n)}]^{-1} \tag{3.3}$$

The algorithm implemented in this final module receives information from the previous modules, such as the odometry estimation and the point cloud captured. The first step is to transform all coordinates (from the points and the pose of the robot) to the global map frame. The next step is to build a temporary local map in an octree structure using the received point cloud, but only if the displacement of the robot is higher than the defined threshold. The displacement is calculated using the current pose of the robot and the previous pose, which means that after building the local map and updating the global map the current pose of the robot has to be saved. After building the local map, the global map is updated. For each node in the local map, the global map is searched to check if the node already exists in the global map. If the node already exists, then its probabilistic occupancy value is updated using the value it has associated in the local map. If the node does not exist, then it is inserted in the octree that stores the global map information and its probabilistic occupancy value is set to the value it has in the local map. After iterating through all the nodes in the local map, the local map is eliminated, and the robot's current pose saved. The process repeats when one of the three conditions presented previously is true.

3.1.2 Depth Camera L515

The sensor chosen to incorporate the UVC Dose Mapping system was the Intel RealSense Camera L515[2], which is a LiDAR depth camera. This LiDAR (Light Detection And Ranging) sensor estimates the distance from the sensor to objects in the environment by sending a laser beam and measuring the travelling time that the light takes from reaching its target and returning [52]. L515 is able to capture depth from 0.25 *m* to 9 *m* at 95% reflectivity, and incorporates an RGB camera and an inertial measurement unit. Other components of this sensor are an IR laser to emit light, an IR photodiode to capture the reflected beam of light, a MEMS (Micro-Electro-Mechanical System) that scans the IR laser beam, a MEMS controller, and an Intel RealSense Vision ASIC (Application Specific Integrated Circuit) that processes the data received from the photodiode and outputs a depth point. Each depth point represents the distance from the sensor to a specific point, and the aggregation of all depth points calculated generates a point cloud that can be used to build a map of the environment.

The desired system is meant to be used in an indoor environment without moving objects or people. Since the L515 sensor is optimized for indoor environments, it can be an adequate choice for integrating the desired system. Another advantage of using this sensor is its ability to detect rapidly moving objects because it has a short exposure time of less than 100 *ns* per depth point, even though moving objects are not expected in the environment. The performance of this LiDAR is worse when there is interference of sunlight, however it can be improved by changing its parameters, such as sharpness or gain. This camera is also compact, with a diameter of 61 *mm*, and lightweight, with a weight of approximately 100 *g*, which means it can easily be used in a small robot. Another characteristic that makes this sensor adequate to be used in the desired system is the existence of software wrappers that support several programming languages and environments, such as ROS which is an important tool used in the development of the system. The L515 is also a solid-state LiDAR, the type of sensor the SSL_SLAM method was developed to be used with. Another feature worth noting of this high-resolution LiDAR sensor is that it requires less than 3.5 *W* of power consumption for depth streaming, which is considerably lower than other solutions.

3.1.3 Robot Operating System

As stated previously, the Robot Operating System (ROS) is the main tool used to develop the presented system. A brief introduction to this tool was given in Subsection 2.2.2.1. Many open source algorithms are available to be used in ROS, such as OctoMap and SSL_SLAM. Each one of the three modules described in SSL_SLAM are ROS nodes that communicate with each other using topics. Topics are a stream of messages, they are basically the way nodes can transmit messages. Each node can subscribe to topics to receive messages from other nodes and publish topics that may be subscribed by other nodes. For example, the ROS node that encompasses the probabilistic map construction, subscribes to the topic */odom* to receive odometry messages from the ROS node responsible for odometry estimation. In a launch file, which is a XML file,

[2]https://www.intelrealsense.com/lidar-camera-l515/

multiple nodes can be launched (run the program in the node) and parameters set. Parameters are essentially global variables accessible to all running nodes. ROS nodes can read, create and modify parameters. For this UVC dose mapping implementation many parameters were created by the ROS nodes. Parameters values can be easily modified through the launch file, so variables such as the UV light bulb power can be easily modified without having to change the program and recompile it. The parameters related to the UVC dose map building algorithm are the map resolution (can be modified to obtain more or less details of the environment), the maximum irradiance distance (depends on the lamps used), width, length and height of the robot (to avoid updating the UVC dose value of the cells that are bellow the robot or on its shadow), the required UVC dose to properly disinfect the environment, the UVC lamp power, and also the minimum displacement of the robot that should be considered for the map to be built (this value could change depending on the sensor used). L515 definitions such as maximum range are also parameters that can be adjusted in the launch file to improve the performance and the quality of the resulting 3D map.

3.2 UVC Dose Mapping Implementation

The main objective of this dissertation is to develop a UVC dose map, in other words, a 3D grid map that associates to each occupied cell that integrates the 3D map a UVC dose value. This dose value is updated if the corresponding cell is irradiated by UVC light. The update of the dose value of each cell must be simultaneous with the construction of the 3D occupancy grid map. To build the 3D occupancy grid map, the algorithm SSL_SLAM was used. As explained previously, this algorithm is divided in three modules, and the third is the one that constructs the map of the environment. This module is based on the algorithm OctoMap, which uses an octree to store the 3D map. The map built is a grid map, which means the environment is represented by cells (cubes) of adjustable size. To each cell a value that represents the probability of it being occupied space is associated. The occupancy value of a cell is calculated and updated with each iteration of the algorithm if any point belonging to that cell is captured by the sensor used. Each cell is represented in the octree as a node and has associated the occupancy value and the centre coordinates of the cell. Everything that has been seen at least once and processed is considered either occupied or free space, which means that every node in the octree represents a cell that has been classified as occupied or free. The probabilistic occupancy value is calculated using equation 3.3, and the threshold values that characterize a cell as either occupied or free can be adjusted using *void octomap::AbstractOccupancyOcTree::setOccupancyThres (double prob)*. The default threshold is 0.5.

The algorithm SSL_SLAM already builds the 3D map of the environment, however, to achieve the objective this algorithm had to be modified to incorporate the UVC dose calculations. As mentioned previously, the SSL_SLAM algorithm uses an octree to store the 3D map. The octree this algorithm uses is a ColorOcTree and each node of this octree is a ColorOcTreeNode. Several functions of the *octomap::ColorOcTree* and the *octomap::ColorOcTreeNode* classes are

used to simplify searching the octree and updating and inserting nodes. Each ColorOctreeNode stores the coordinates and occupancy value of each cell, as well as the RGB color of the cell. To store the RGB color, exists the *octomap::ColorOcTreeNode::Color* class that has three data fields, namely three *uint*8_*t* variables to save each of the RGB components of the color of the corresponding cell. For this application, the color of each cell was unnecessary information, so these three variables, that are associated with each existing node in the octree, were used to store the UVC dose value of each cell. This modification allowed to easily store the UVC dose values without having to create another data structure and allowed the use of all functions and advantages of the octree, such as efficient search. However, *uint*8_*t* variables are only able to represent values from 0 to 255 and the UVC dose can have any positive decimal value. For this reason, the three *uint*8_*t* variables were used to represent the dose value associated with each cell. To represent a decimal value using unsigned int variables, a maximum dose value was defined and the following calculations were made. Considering that the original scale of the UVC dose value has a range from 0 to *max_dose* and the new scale to which this value has to be converted into ranges from 0 to $2^{3\times8}-1$ (three variables, each with 1 byte which is equal to 8 bits):

$$X = X \cdot \frac{2^{3\cdot8}-1}{max\_dose} \tag{3.4}$$

$$r = (int)\frac{X}{2^{16}} \tag{3.5}$$

$$X = X - (r \cdot 2^{16}) \tag{3.6}$$

$$g = (int)\frac{X}{2^{8}} \tag{3.7}$$

$$X = X - (g \cdot 2^{8}) \tag{3.8}$$

$$b = (int)X \tag{3.9}$$

In equations 3.4 to 3.9, X represents the value to be saved, and r, g and b represent the three *uint*8_*t* variables. In equation 3.4, the dose value is converted from its original range to the representation range when using 3 bytes. Equation 3.5 saves in one of the three *uint*8_*t* variables (where the r component of the RGB value would be saved) the most significant byte of the dose value in its new range. A cast to integer is performed because *uint*8_*t* variables can only represent integers. The remaining steps (equations 3.6 to 3.9) save the intermediate byte in the variable g, and the least significant byte in the variable b.

As mentioned previously, the UVC dose value of each cell is updated whenever it is irradiated by UVC light. The complete desired system is a mobile robot that has UVC lamps to perform surface disinfection. According to equation 2.2, the irradiance depends on the area of the

irradiated surface, that is most commonly considered as a spherical area. When using a spherical model, it is considered that the irradiance is equally distributed in all directions, which simplifies calculations. The area of a spherical surface is given by equation 3.10, where A is the area of the spherical surface and *radius* is its radius. In equation 3.11, I_r is the irradiance and P is the total UV bulb power.

$$A = 4 \cdot \pi \cdot radius^2 \quad (m^2) \quad (3.10)$$

$$I_r = \frac{P}{4 \cdot \pi \cdot radius^2} \quad (W/m^2) \quad (3.11)$$

This demonstrates that the irradiance depends on the distance from the light source to the irradiated surface. The irradiance value is higher when the irradiated surface is near the light source and lower when it is further away. From a certain distance value, the irradiance is 0 or insignificant. For this reason, not all cells of the map are updated with every iteration of the algorithm. A maximum irradiance distance has to be defined, and only the cells that are inside the spherical area with a radius equal to this maximum distance value and a centre equal to the origin of the robot pose frame (in this case, because in this implementation and in the tests performed UV lamps were not used) can be updated. The maximum distance depends on the lamp used and on the design of the robot, so this value can be easily modified and it has a default value equal to 3.0 *m* in case the maximum distance is not specified.

Even if a cell is contained within the spherical area, it still may not be irradiated by UVC light and therefore its UVC dose value cannot be updated. If a cell is in a shadowed area, for example behind another occupied cell or bellow the robot, it is not irradiated by UV light. A cell is considered to be bellow or shadowed by the robot if its centre coordinates are within the footprint of the robot plus the surrounding shadow of the robot. To verify this, the width and length measures of the base of the robot are necessary and the centre coordinates of the cell have to be transformed from the global frame to the robot frame, because in the algorithm all the coordinates are in the global map frame. Using the odometry messages from the second module and the Eigen library, the third module of SSL_SLAM calculates the homogeneous matrix that represents the transformation from the robot frame to the global map frame. To transform a cell's coordinates from the global map frame to the robot frame, the homogeneous matrix that represents this transformation is required. This matrix is different whenever the robot moves, so it is calculated with each iteration of the algorithm (every time the UVC dose values are to be updated). This homogeneous matrix is equal to the inverse of the homogeneous matrix that represents the transformation from the robot frame to the global map frame. The transformation of the coordinates of each cell to the robot frame is achieved by multiplying the point they represent by the homogeneous matrix calculated.

$$p^{robot\,frame} = H^{-1} \cdot p^{global\,frame} \quad (3.12)$$

In equation 3.12, $p^{robot\,frame}$ is a column matrix with the three first lines equal to the desired centre

point of the cell in the robot frame and one line with the value 1, H is the homogeneous matrix from the robot frame to the global frame, and $p^{global\,frame}$ is a column matrix with the three first lines equal to the given centre point of the cell in the global map frame and one line with the value 1 (for example, $p^{global\,frame} = [x \quad y \quad z \quad 1]^T$).

In addition, a cell cannot be updated if an obstacle is preventing it from being irradiated by UVC light. It was considered that a cell was not being irradiated if another occupied cell was obstructing it. To establish this, 3D raycasting was performed by using the *castRay()* function from the *octomap::ColorOcTree* class. This function casts a ray from the origin with a given direction, returns in a *point3d* variable the centre coordinates of the first non-free cell found, and returns true if an occupied node was hit by the raycast. The origin of the raycast is the robot's current pose (origin of the robot frame), unknown cells are ignored, and the maximum range to which the raycasting is performed is equal to the radius of the spherical surface area (more time efficient and there is no necessity to continue to search for a node if it is not inside the irradiated area). The direction in which the rayscasting has to be performed is the direction of the vector formed by the current pose of the robot and the centre of the cell that is being analysed. For every cell whose centre is inside the irradiated area, the 3D raycasting is performed, and if the coordinates returned in the *point3d* variable are not equal to the centre coordinates of the cell that is being analysed, then it is because another cell was shadowing the first one and its UVC dose value should not be updated. Figure 3.1 exemplifies cells that can and cannot be updated according to the conditions described.

As explained previously, each cell is represented in the octree as a node. Firstly, when a new node is added to the octree, its UVC dose value is considered 0. When a point cloud is received, the octree is iterated and each node is examined to determine if its UVC dose value should be updated or not. If a cell is inside the irradiated area and is not shadowed by any obstacle or even by the robot, then the dose value has to be updated. With each update, a value is incremented to the UVC dose until it reaches the required dose. Different microorganisms have different D_{90} and D_{99} values, so the required UVC dose can be different in different disinfection situations. For this reason, the required dose is an adjustable parameter in this system. The estimation of the dose increment is given by equation 3.13, that results from equations 2.1 and 2.2.

$$\Delta d = \frac{P}{4 \cdot \pi \cdot radius^2} \cdot \Delta t \qquad (J/m^2) \quad (3.13)$$

Δd is the dose increment, P is the total bulb power, *radius* is the distance from the UV lamp to a specific cell, and Δt is the time period between dose increment calculations. The value of Δt is different for every cell because the first node is not updated at the same time as the last node in the octree (depending on the chosen map resolution, the time it takes to iterate through the octree varies and was considered in the Δt calculation), and is calculated using the time library <chrono>. In this implementation and in the tests performed, only the L515 sensor was used, and for this reason *radius* was considered as the distance from the sensor's current pose (instead of the UV lamp) to a specific cell. The value of *radius* is calculated for each cell by equation 3.14.

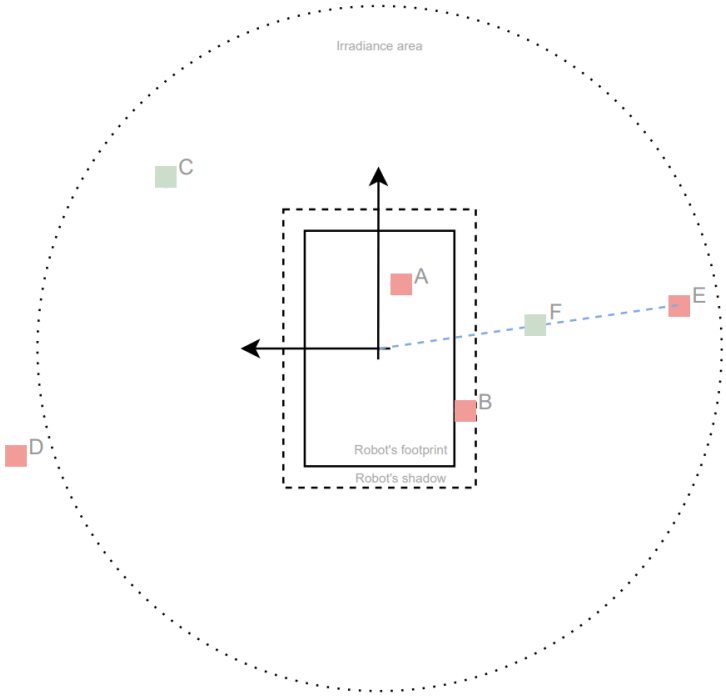


Figure 3.1: 2D example of cells whose UVC dose value can and cannot be updated because of the three conditions presented. Cells in red (A, B, D, E) cannot be updated. Cells in green (C, F) can be updated.

The coordinates of the robot's current pose are (x_r, y_r, z_r) and the centre coordinates of the cell are (x_c, y_c, z_c).

$$radius = \sqrt{(x_r - x_c)^2 + (y_r - y_c)^2 + (z_r - z_c)^2} \quad (m) \quad (3.14)$$

After its estimation, the dose increment is added to the UVC dose value stored in the three *uint*8_*t* variables that are associated with each node of the octree. Firstly, the dose value, D, is reconstructed from the three variables as shown in equation 3.15, then it is recalculated in the same range as the dose increment (which is in the original UVC dose range), then the values are added, and finally, the new dose value is converted to be stored in the three 1 byte variables.

$$D = b + g \cdot 2^8 + r \cdot 2^{16} \quad (J/m^2) \quad (3.15)$$

The UVC dose mapping algorithm is the SSL_SLAM method modified to incorporate an algorithm that associates with each cell of the 3D occupancy grid map a dose value and updates it. The logic and conditions behind updating the dose value have been presented, in the remainder of this chapter the full algorithm and its steps sequence will be presented and clarified. The algorithm is composed of three modules, the first two are the same as the SSL_SLAM method, the third module was modified. The third module begins by verifying if the point cloud and odometry

messages are being received. In this dissertation, the system was physically represented by the sensor used, which means that if no point cloud is received, it is because either the system is not in operation or because an error occurred. In both these cases, no map will be built. After confirming that both the point cloud and the odometry messages are being received, the homogeneous matrix that represents the transformation from the robot frame to the global map frame is calculated. The robot's current position is the origin of the robot frame. Before updating the global map, the UVC dose values are updated. The function that updates these values, iterates through the entire octree that stores the global map using a $leaf\_iterator$ (the $leaf\_iterator$ class is used in the OctoMap algorithm to iterate through all the leafs of a tree). For each node of the octree, its coordinates are extracted and the distance to the robot's current pose is calculated. All of these coordinates are in the global map frame. If the distance calculated is inferior to a maximum distance defined, it means that the cell the node represents is inside the irradiated area by the UV lamps. If the cell is inside the irradiated area, then it is possibly being disinfected, so it is determined if the cell has any obstacles, is bellow the robot or is in the robot's shadow area. The process to determine these conditions has been thoroughly described. If all these necessary conditions are met, then the space represented by the current cell is being disinfected and the UVC dose value has to be updated. After these confirmations, the dose increment is calculated, the dose value is reconstructed and recalculated in the original dose range, the dose increment is added to this dose value, the resulting new dose is recalculated in the range $\{0, 2^{3\cdot 8} - 1\}$, and is saved in the three *uint8_t* variables as previously described. The three $uint8\_t$ variables were originally used to store the RGB values of each cell, and the function used to modify these variables is *setColor()* and receives as parameters r, g and b, which in this algorithm store the most significant byte, the intermediate byte and the least significant byte, respectively. After iterating through all the nodes in the octree, the UVC dose updating portion of the algorithm is complete. The dose updating function is presented in Algorithm 2. Following this, the probability map is updated. By using the received point cloud, a local map is created that is then used to create the global map. This part of the algorithm as also been described previously, in Subsection 3.1.1. Finally, the global map and other necessary information are published. The main development tool used in this dissertation is ROS, so the final portion of the algorithm is dedicated to publishing all visualization information to be used by rviz (a ROS 3D visualization tool), with the exception of the UVC dose values that are published

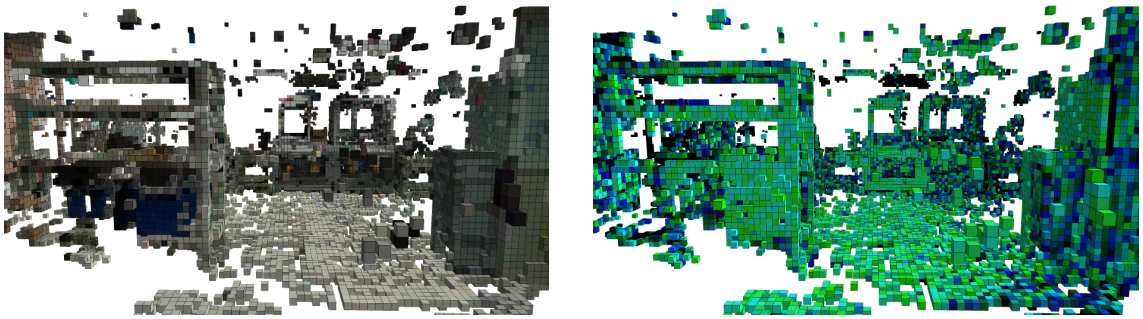
immediately after being updated. The process repeats as long as the system is running.

Algorithm 2: UVC dose updating algorithm

```
Input: Current pose
for each node in octree of global map do
    Get node coordinates;
    Calculate distance from node coordinates to current pose, dist;
    if dist < maximum irradiated distance then
        if there are no obstacles then
            if cell is not shadowed by the robot then
                Calculate time period, Δt, since last dose calculation of the current node;
                Calculate irradiated area;
                Calculate dose increment, Δdose;
                Get current dose value from (r, g, b);
                Calculate new dose by adding current dose and Δdose;
                if new dose <= required UVC dose then
                    Save new dose in (r, g, b);
                end
            end
        end
    end
end
```

3.3 Visualization Method

The algorithm developed builds a 3D occupancy grid map that associates and updates a UVC dose value to each cell of the map. A practical and intuitive approach to validate the map built and to visualize the building process and UVC dose increments is to use rviz, the 3D visualization tool for ROS. rviz can subscribe to any topic being published by any running node (topics and nodes have been briefly explained in Subsection 3.2), such as the topic that publishes the ColorOctree. Using rviz, the building of the 3D map of the environment can be visualized, each cell of the map is represented by a cube whose size depends on the resolution chosen. As explained previously, each node in the ColorOctree has three variables associated that would usually represent the RGB color of the cell represented by the node. When visualizing the ColorOctree in rviz, the color of each cube is determined by the values stored in those three variables. However, in this implementation these three variables store the UVC dose value instead of an RBG color. As a result of this modification, visualizing the 3D map using the ColorOctree directly would be confusing and areas that have been disinfected would not be distinguishable from areas that have not. The solution found for intuitively representing the 3D map and the UVC dose values was to use markers. In figure 3.2, the resulting 3D dose map without using markers to visualize it is presented. This map was built using a test file provided by the authors of the SSL_SLAM method. Each cell color is

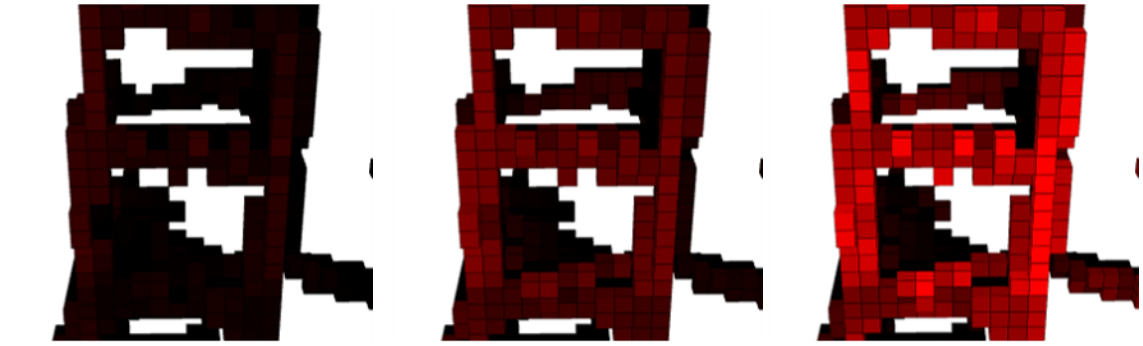


(a) (b)

Figure 3.2: (a) 3D map built with the SSL_SLAM method. (b) 3D dose map built of the same area without using markers.

directly the decomposed value of the UVC dose value stored in the *uint*8_*t* variables. As the figure demonstrates, the map obtained is not informative regarding the UVC dose values of each cell and current state of the disinfection process.

Markers are a rviz display type that allow the display of 3D primitive geometric shapes, such as cubes. These shapes are displayed using *visualization_msgs/MarkerArray* messages. These messages have to be published through a topic, in this implementation this topic is published by the node that executes the program that builds the 3D map. With this approach, while the map is being built, cubes that represent the occupied cells of the map start appearing on rviz. In this application, the RBG color of the cubes range from $(0,0,0)$ to $(255,0,0)$, which means each cell is black or a shade of red. When a new space of the environment is mapped, it is considered that it has not been disinfected yet, so the UVC dose value attributed to the cell that represents that space is 0. As the UVC dose value of a specific cell starts increasing, the color of the cube that represents it becomes brightly red (see figure 3.3). When the dose value reaches its maximum (the required UVC dose value to properly disinfect the surface), the cube has an RGB color of $(255,0,0)$, which is red.



(a) (b) (c)

Figure 3.3: Example of the progression of the UVC dose value of a surface mapped. From (a) to (c) the dose value increases.

After updating the UVC dose values and updating the global map, the visualization solution

is implemented. The algorithm that publishes the *visualization_msgs/MarkerArray*, starts by creating a *visualization_msgs/MarkerArray* class variable and iterating through the entire octree that stores the global map. For each occupied octree node found (the octree saves both occupied and free nodes), the UVC dose value associated is reconstructed using equation 3.15 and recalculated to its original range, that encompasses values from 0 to the maximum dose value defined in the launch file (which is the required UVC dose value to disinfect the environment). The *visualization_msgs/MarkerArray* variable created is basically an array that can store in each position the coordinates and color of the 3D geometric shape it is supposed to represent, which in this implementation is a cube. The coordinates of each cube are the coordinates of each occupied node in the octree. The color of each cube represents the dose value of the node. This color representation is achieved using only the red component of the RGB color. In this case, the red component value can range from 0 to 1 (which represents a range from 0 to 255). The dose value of each node is converted to this range and the resulting value is saved in the red component, while the green and blue components are set to 0. With this representation, as the UVC dose value starts increasing, the color of a specific cube becomes increasingly brightly red and the UVC dose map visualized becomes readable and intuitive. After saving the colors and coordinates of all occupied nodes, other parameters have to be defined such as the type of geometric shape desired. This visualization algorithm only runs if a ROS node subscribes to the marker array, which means that it only runs if in rviz the option to represent the 3D map as described in this section is set.

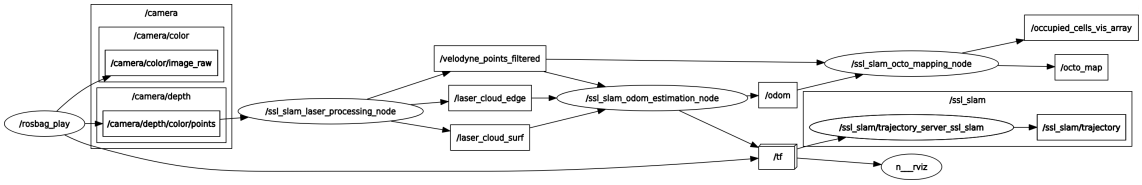


Figure 3.4: Graph with all active ROS nodes and topics during the execution of the UVC dose mapping algorithm

In figure 3.4, a graph with all ROS nodes and topics involved in the UVC dose mapping algorithm and including the visualization algorithm is presented. The ROS nodes are represented by oval shapes, the topics are represented by rectangular shapes, and the arrows represent communications. $/camera/depth/color/points$ is the topic that contains the point cloud, and that is subscribed by the ROS node $/ssl\_slam\_laser\_processing\_node$. The $/ssl\_slam\_octo\_mapping\_node$ is the ROS node that runs the UVC dose mapping algorithm. This node publishes the 3D global map using the topic $/octo\_map$ and the markers that make the map readable using the topic $/occupied\_cells\_vis\_array$.

Chapter 4

Results

On the previous chapters, the background, motivation, and development of the UVC Dose Map have been presented. To validate the algorithm created to build the 3D dose map several tests were performed. These tests were performed using the L515 sensor and not using the entire desired system, because the sensor is the subsystem required to test the algorithm and is sufficient. As explained in Subsection 3.1.2, the L515 sensor has a different performance when used under different conditions, especially lighting conditions. For this reason, the method developed was tested in different locations with different characteristics, such as different layout, lighting, existence of windows. During the development of the UVC dose map building algorithm, several tests were also performed to ensure that different modules of the algorithm were functioning as expected, however these will not be presented in this chapter. Each group of tests that were performed to evaluate the algorithm's performance when in certain conditions are presented in Section 4.1, Section 4.2, Section 4.3, Section 4.4 and Section 4.5. The results obtained are discussed in Section 4.6.

To validate the UVC Dose Mapping algorithm developed, tests have to be performed and their results analysed and discussed. The sensor used to capture the environment to be mapped performs best when there is minimal sunlight present in the environment, when there are reflective surfaces, when no other LiDAR sensors are present, and is optimized for indoor applications. The desired disinfection system is meant to be used in indoor applications, so all the tests performed are indoors. However, to properly validate the work developed, the tests should be performed in environments with varying conditions that affect the sensor's performance. For this reason, the tests were executed in environments with different sunlight exposure and different layout with surfaces of different reflectivity. Other conditions that were modified between tests were the presence of more or less obstacles in the environment and varying moving and rotating speed of the sensor. For the tests only the L515 sensor was used and a computer with a Intel® Core™ i3-4005U CPU @ 1.70GHz. Firstly, the results obtained with the test file provided by the authors of the SSL_SLAM method will be presented. Secondly, the results obtained with the sensor placed in a remote controlled wheelchair and obtained with the L515 sensor hand-held will be presented and compared. Finally, the results of the tests with the varying conditions described will be demonstrated.

The following images presented are the results of tests performed under the same parameters but different environmental conditions. The map resolution was set to 0.1 m, which means the resulting maps have enough details to recognize shapes while not having a number of cells that overloads the processor of the computer used. The maximum distance the sensor can detect points was set to 9.0 m, which corresponds to the maximum distance the L515 can detect in ideal conditions, and the minimum distance was set to 0.2 m. The radius of the irradiated area was set to 5.0 m, instead of its default value of 3.0 m. For the purpose of shortening the tests duration, the required UVC dose value was set to 50 J/m^2 and the lamp power to 100 W and its efficiency to 1.0, which means the dose increments will be higher. The footprint of the robot was left with the default values which represent a width of 20 cm and length of 20 cm. Finally, the minimum displacement for the map to be updated was set to 0.5 m. The ROS visualization tool, rviz, runs at 30fps. In most of the figures bellow, the global map frame and the sensor frame (which would be the robot frame, but in these tests only the L515 sensor was used) are represented. The global map frame is the starting position of the sensor. The green axis represents the y axis, the x axis is red and the z axis is blue.

4.1 Experiment 1

The first result is presented in figure 4.2 and was obtained by using a test file provided by the authors of the SSL_SLAM method. The environment in which this test file was created is called environment 1 for the remainder of the chapter. This environment is a wide space with different obstacles and is presented in figure 4.1. Figure 4.1a presents the area seen by the sensor in its starting position, figure 4.1b is the area to the left of the machines seen in figure 4.1a, figure 4.1c is the area seen when the same machines are behind the sensor, and figure 4.1d is the area to the right of the area in figure 4.1c.

This first test was performed in ideal conditions, with no sunlight interference and at a low moving and rotating speeds. The four images in figure 4.2 demonstrate the evolution of the UVC dose mapping algorithm, starting in 4.2a and ending in 4.2d. As expected, the number of cells increases as the sensor is moved around the environment, and because there was no sunlight entering the room the map obtained properly represents the real world environment with some detail (the level of detail is also conditioned to the map resolution chosen). As the sensor is moved, the cells nearer to its position become a brighter shade of red faster (see equation 3.13). As expected, the areas shadowed by obstacles are not disinfected and the cells that represent them remain black, which is representative of an applied UVC dose value of 0. A top view of the 3D map built in this test is presented in figure 4.3.



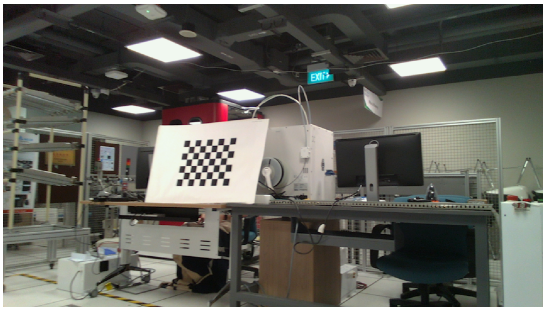
(a) View from the starting position of the sensor.



(b) Left view.



(c) View when the sensor is moving toward the starting position.



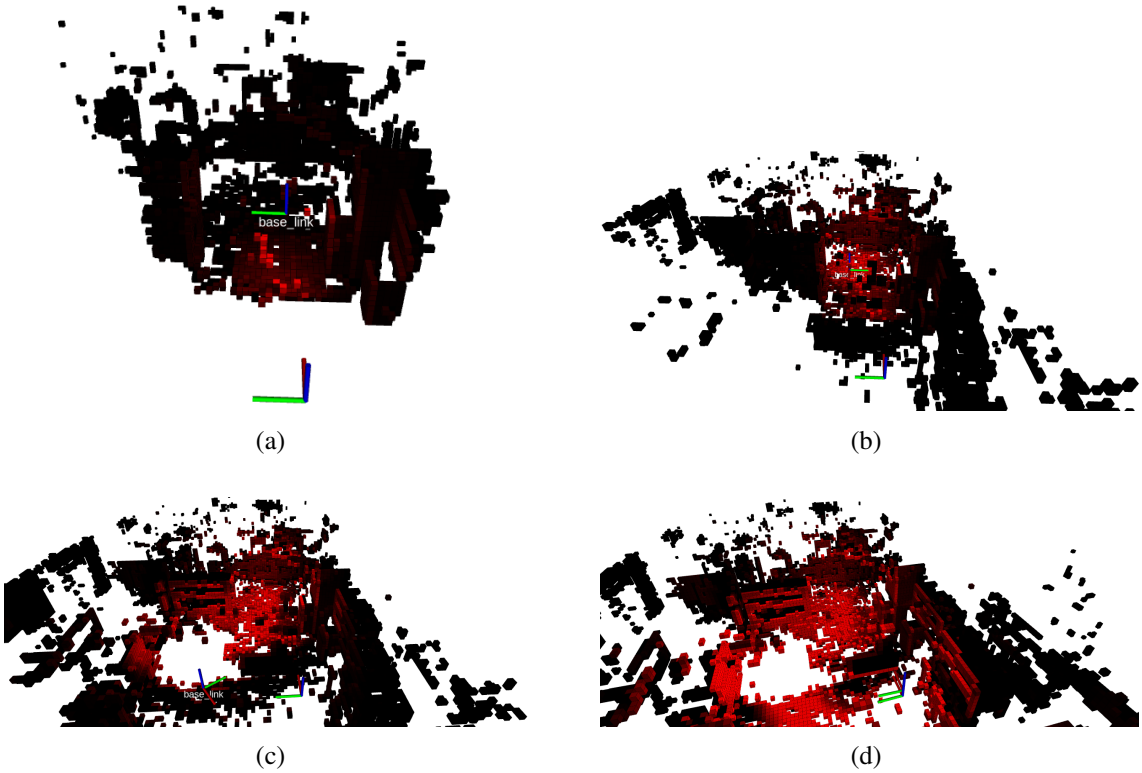
(d) Right view when the sensor is moving toward the starting position.

Figure 4.1: Environment 1

4.2 Experiment 2

After validating the algorithm developed with the test performed under ideal conditions, the algorithm was tested under non-ideal conditions and using created test files (instead of test files provided by the authors of SSL_SLAM). Firstly, the sensor was attached to a remote controlled wheelchair that was moved around a wide environment, which will be referred to as environment 2 for the remainder of this chapter, without sunlight interference and with surfaces with different reflectivity (see figure 4.5). Secondly, the sensor was hand-held and moved around the same environment (see figure 4.6), making the same path as the one made when the sensor was attached to the wheelchair. Environment 2 is composed of a wide area with multiple doors that has attached a long corridor (see figure 4.4).

The height at which the sensor was attached to the wheelchair is inferior to the height the sensor was hand-held, which caused the ceiling of the room to be mapped in the first situation but not completely in the second. The path taken in both these tests starts at the connection between the corridor and the wide area, then it is a circle around the wide area and ends by entering the corridor. The map obtained with the hand-held test represents the reality better than the one obtained with the test where the sensor was attached to the wheelchair. For example, figure 4.7 demonstrates that the corners of the room are more accurately mapped when the sensor was hand-held. This is likely due to small vibrations from the chair that negatively impact the odometry estimation



(a) (b)

(c) (d)

Figure 4.2: Evolution of the 3D dose map built using a test file provided by the authors of the SSL_SLAM method (environment 1). The progress is represented from (a) to (d).



Figure 4.3: Top view of the final 3D dose map built using a test file provided by the authors of the SSL_SLAM method (environment 1).



(a) View from the starting position of the sensor.



(b) Right view of the environment.



(c) Left view of the environment.

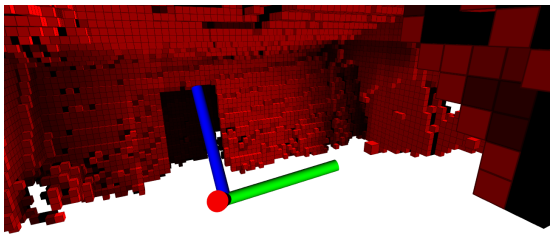


(d) Final view of the environment (corridor).

Figure 4.4: Environment 2



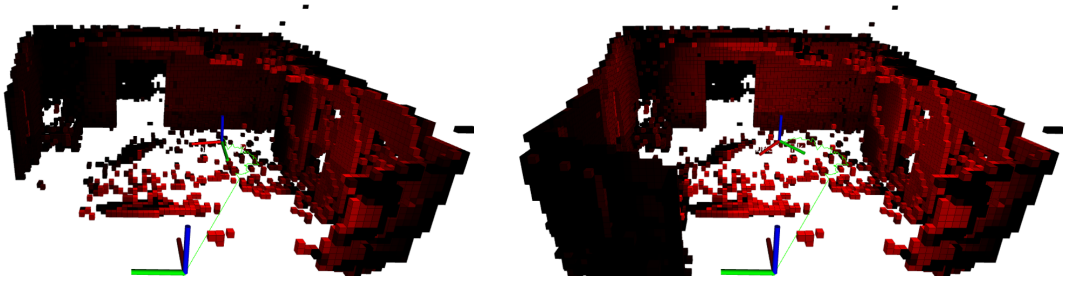
(a)



(b)

Figure 4.5: UVC dose mapping with the L515 sensor attached to a remote controlled wheelchair in environment 2.

and consequently the overall mapping of the points captured. Tests in different environments were performed with the sensor hand-held and with the wheelchair (with the sensor attached at different heights), and the results with the sensor hand-held were consistently better. For this reason, the following results were all obtained with the L515 hand-held.



(a) (b)

Figure 4.6: UVC dose mapping with the L515 sensor hand-held in environment 2.



(a) (b)

Figure 4.7: Top view of the 3D maps built with the sensor attached to a remote controlled wheelchair, (a), and hand-held, (b), in environment 2.

4.3 Experiment 3

The objective of this experiment was to evaluate the impact of the speed of the sensor (both moving and rotating) in the final result. Figure 4.8 demonstrates another 3D map of environment 2. This map was obtained by moving the sensor at a considerably higher speed (moving and rotating) than in the previous tests performed in Section 4.2. When inspecting figure 4.8, the corridor connected to the wider area is not properly mapped, this is due to the fast rotation speed of the sensor.

Figure 4.10 demonstrates the maps built of environment 3, which is a narrower area than the wide area in environment 2. Environment 3 is composed of a main corridor with several objects, and a second corridor exists to the left of the first one (see figure 4.9). This environment has a wall completely made out of glass that lets sunlight penetrate the room, however both the tests in figure 4.10 were performed at night when there is no sunlight interference and at different moving speeds. Figure 4.10b was obtained by moving the sensor at a higher speed than in figure 4.10a, except when rotating the sensor. In both these tests, the sensor was moved forward the first corridor and then rotated to face the second corridor and moved forward again. The map in figure 4.10b has less details than 4.10a and was not able to map the floor as accurately, however the final map still properly represents the environment and the objects are still distinguishable. In figure 4.10b, the connecting corridors were properly mapped because the rotation speed was not high. The results

from figures 4.8 and 4.10 demonstrate that the rotation speed influences the odometry estimation, and should be low for a better result.

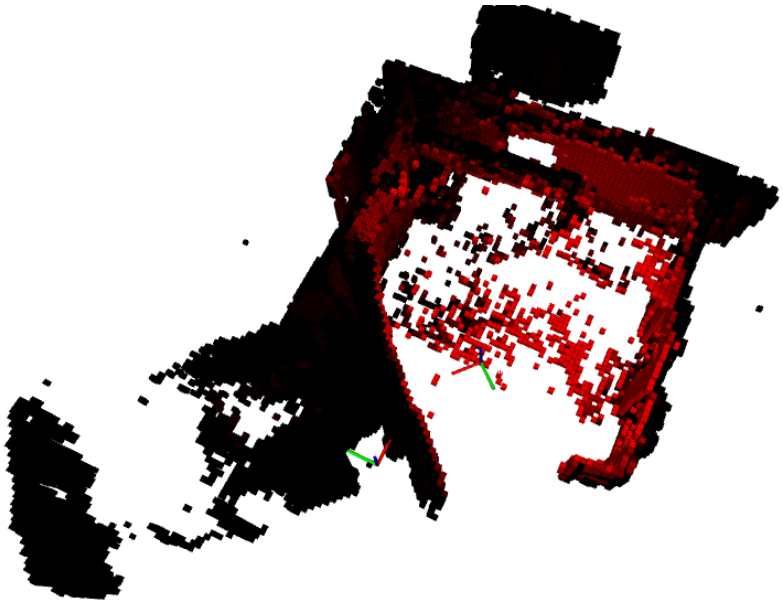


Figure 4.8: Top view of the map of environment 2 with a high sensor moving speed.



(a) View from the starting point (first corridor).

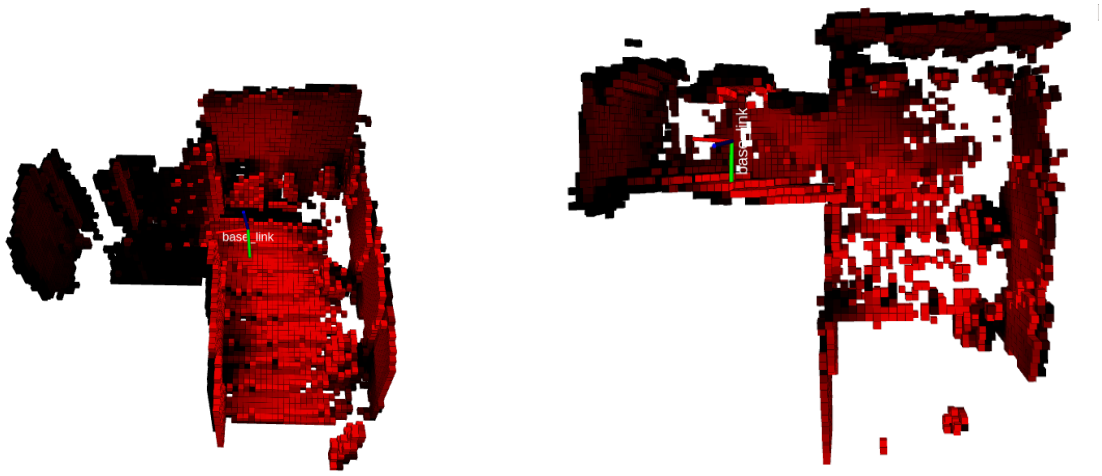


(b) Second corridor (to the left of the first one).

Figure 4.9: Environment 3

4.4 Experiment 4

As previously mentioned, the performance of the L515 depth camera is affected by the presence of sunlight, which influences the quality of the 3D map built. The objective of experiment 4 is to analyse the results of the algorithm when sunlight is present. In this section, two tests are presented in environment 3 (see figure 4.9) and environment 4 (see figure 4.11). Figure 4.12 was obtained by mapping environment 4, which has glass doors that allow sunlight to penetrate, however this



(a) (b)

Figure 4.10: Top view of the maps of environment 3, when moving the sensor at a low speed, (a), and at a higher speed, (b), except when rotating.

light is partly blocked by coffee machines that are placed a couple of meters in front of the doors, which means that there is some sunlight that reaches the areas presented in the figure. In 4.12a the sunlight interference is minimal because of the coffee machines (that are to the left). In 4.12b the L515 was near the coffee machines and facing the glass doors. As the sensor approaches the doors, more sunlight starts to interfere with it, and the algorithm is no longer able to correctly map the environment nor estimate the current pose of the sensor. With the increase of sunlight interference, the odometry estimation error increases, especially when rotating the sensor. Figure 4.13 presents another map built of environment 3, however the points used to build this map were captured during a sunny day. As mentioned previously, in environment 3 there is a glass wall, which means that during a sunny day an extreme amount of sunlight penetrates the area. Trying to map this environment in these conditions proved to be ineffective, as demonstrated by figure 4.13. The objects are not distinguishable and many points were mapped with the incorrect coordinates and appear as noise.



(a) View from the starting point (coffee machines to the left).

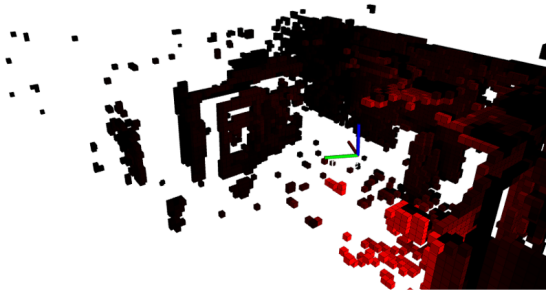


(b) Coffee machines blocking the light that enters through the doors.

Figure 4.11: Environment 4



(a)



(b)

Figure 4.12: Results from the mapping algorithm of environment 4 in the presence of sunlight.

4.5 Experiment 5

A final test was performed in environment 2, which was the environment in which the best results were obtained, but with the inclusion of an obstacle. The path taken was the same as in experiment 2 (Section 4.2), and the sensor was moved and rotated at a low speed. The purpose of this test was to validate the UVC dose update algorithm. When cells are being shadowed by an obstacle, their UVC dose value should not be updated, and their color should remain the same. The result of this test is presented in figure 4.14. In 4.14b it is shown that the cells behind the obstacle, that are prevented from being irradiated by the UVC light, have a black color, which means that their UVC dose value has not been updated. This is the desired behaviour. Figure 4.15 demonstrates the same behaviour, in the same environment (environment 2) but with a different obstacle, a motor placed on the ground.

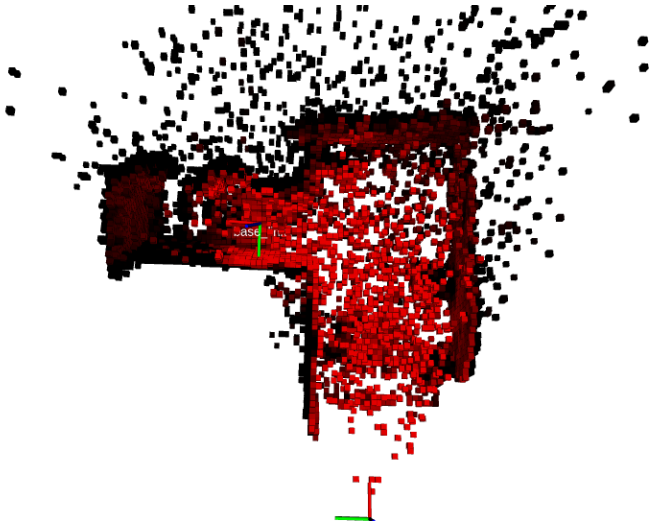
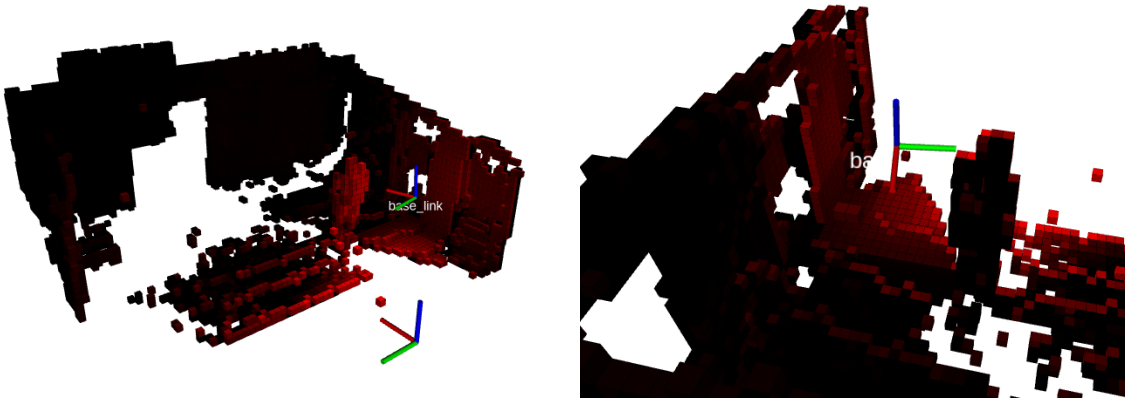


Figure 4.13: Results from the mapping algorithm of environment 3 in extreme lighting conditions.



(a) (b)

Figure 4.14: Mapping of environment 2 with an obstacle.

4.6 Discussion

The tests performed on the method developed reveal that the method works as desired when the conditions are close to ideal. With minimal sunlight exposure and a lower moving and rotation speed, the resulting 3D map accurately represents the surrounding environment and the UVC dose value is updated as expected. The environmental conditions do not affect specifically the UVC dose update algorithm, but because they can highly impact the mapping algorithm (because they affect the sensor's performance), they interfere with the overall outcome of the UVC dose map. For a better resulting map, the L515 depth camera should be properly secured to the robot or other device, in order to minimize vibrations and odometry errors. The odometry estimation errors impact the mapping of the points captured and may lead to incorrect mapping of certain areas of the environment. For example, these errors may lead to a planar surface be represented in the 3D map as a group of cells that do not have the same height, in other words, a planar surface may

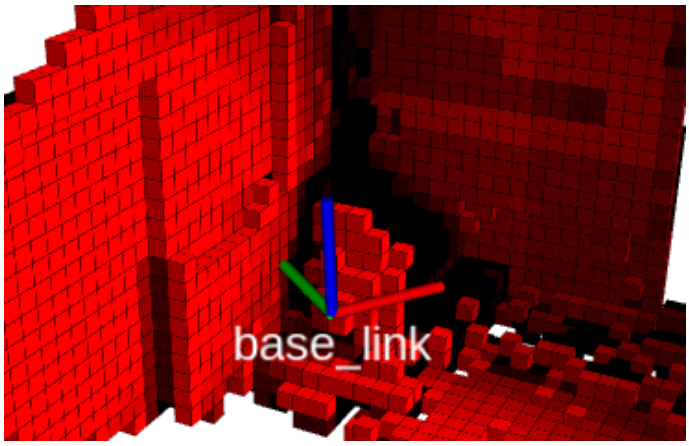


Figure 4.15: Mapping of environment 2 with another obstacle on the ground.

not be represented as a planar surface. This impacts the UVC dose update algorithm because the dose value of cells of a planar surface that is being irradiated by UVC lamps may not be updated because when checking for obstacles, the algorithm will interpret that some of these cells are being shadowed by other cells from the same surface (because they were mapped with different heights). It was demonstrated that when the environment was reached by sunlight it also lead to errors in the odometry estimation. Besides this negative effect, the maximum detected distance by the sensor is also heavily conditioned by the amount of sunlight present in the environment, and the field of view of the sensor is also considerably affected. The desired disinfection system is meant to be used indoors, however in rooms where windows exist sunlight can always be a factor. All the tests presented previously were performed in indoor facilities, however when large windows existed, the resulting map was not satisfactory. This limitation of the L515 sensor limits greatly the overall system, and perhaps another sensor should be used for this system instead. Another factor evaluated was the moving speed of the sensor. For higher speeds, the resulting map is not as detailed, but this problem is overcome by having the sensor repeat its path or capturing the same areas more than once. However, the rotation speed of the sensor cannot be high, otherwise the algorithm is not able to properly estimate its current pose which then affects every point mapped. This is a limitation that cannot be surpassed. When there is sunlight interference, the rotation speed has to be lower than when there is no sunlight. Another limitation related to the sensor used is that its performance depends on the reflectivity of the surrounding surfaces. On narrow spaces that have no distinguishable details, such as a narrow corridor with simple walls and nothing else, the algorithm as difficulties recognizing when and how the sensor is moving, and if it is only moving in the same direction. To summarize, the system is limited by the L515 sensor and has a better performance when there is minimal sunlight, when the sensor is properly secured, and when the rotation speed is low. The UVC dose update algorithm performs as desired, but the odometry estimation module of the algorithm could be improved.

Chapter 5

Conclusions and Future Work

The main objective of this dissertation was to develop a system that is capable of building a 3D map of the surrounding environment while associating and updating a UVC dose value to each cell that integrates the map, in order to apply the minimal UVC dose required to every surface in the environment to properly disinfect it. Firstly, the context and motivation for this dissertation were given. The described system was motivated mainly by the Covid-19 world pandemic and the necessity for a chemical and human intervention free disinfection method. Secondly, basic UVC disinfection concepts were presented as well as the mathematical model for a UVC disinfection process. The mathematical model presented was later applied when developing the algorithm that builds the UVC dose map. An overview of existing mapping methods was also given, including an introduction to OctoMap, which was the basis for the mapping algorithm of the system. Thirdly, the technologies used in the development of this algorithm were presented, as well as the algorithm itself, and its characteristics, functionalities, and logic. The technologies used include the SSL_SLAM method, the Intel RealSense L515 depth camera, and ROS. Finally, it was presented the results of some of the tests performed on the overall algorithm developed with varying conditions that revealed vulnerabilities and limitations of the technologies used and of the algorithm.

5.1 Conclusions

The desired UVC dose mapping algorithm was tested in different environments with varying conditions in order to demonstrate the vulnerabilities and limitations of the system and points of improvement. A detailed view of the results obtained from these tests can be found in Chapter 4. From the tests performed, the algorithm was validated. As desired, a 3D occupancy grid map of the environment was built where each cell of the map had a UVC dose value associated that was updated according to the mathematical model described in Subsection 2.1.2. It was verified that only the cells that were inside the maximum defined distance and that were not being shadowed by the robot or by an obstacle had their UVC dose value updated. The resulting map had a higher level of detail if the map resolution was a lower value, however with a lower resolution value the number of cells increases and the computational effort and algorithm running time increase

as well. A compromise between these factors had to be made and the chosen map resolution for the tests presented was 0.5. The tests highlighted the limitations of the chosen sensor, the L515. The algorithm developed when used with the L515 sensor has a better performance when used in a wide environment with minimal sunlight, with lower rotating speed, and when the sensor is properly secured and vibrations are minimized.

The main objective described was fulfilled, however many improvements can and should be made for this algorithm to be as versatile as possible and to always have a good performance in different environments. The algorithm should not be as limited to environment conditions because of the incorporation of the L515 sensor. Another objective was the tracking and representation of the current state of the disinfection process. The algorithm developed tracks the UVC dose of every cell, that represents a surface area, and updates its value until it reaches the required UVC dose, and it also provides visual means to track the UVC dose progress and the current state of the process. Because in this first implementation there is no module that decides the trajectory, it cannot be guaranteed that every surface will be completely disinfected by the end of the trajectory, however the UVC dose values are tracked and areas that have not been disinfected are easily identified.

5.2 Future Work

As stated previously, to have a robust system that can be used in multiple situations and environments some improvements have to made to the one developed. As future work, some modifications that lead to the needed system improvements are required, and the system should also be expanded in order to achieve a complete autonomous disinfection system.

One modification that should be made is regarding the LiDAR sensor used in the system. The sensor chosen was the L515 depth camera, however as the tests performed demonstrated this sensor is too sensitive to lighting conditions and limits the overall system. This limitation is not compensated by the advantages of using the sensor, such as its compact size, because the system cannot be robust if this sensor is used. Another LiDAR sensor should be incorporated in the system. The odometry estimation module should also be improved, because it conditions the system to be used at low speed and is easily disrupted by small vibrations. In this implementation and in the tests performed, the robot frame was considered the L515 sensor frame and the centre for calculating the irradiated area was considered to be the sensor's current pose. When using the complete desired disinfection system, the centre of the irradiated area should be related to the UV lamps and not the LiDAR, and the robot's current pose will not be equal to the sensor's current pose (which impacts the verification if a cell is being shadowed by the robot) so a transform between the two frames should be provided.

In this dissertation, the mapping module of a disinfection system was developed. To achieve a complete disinfection system, in this case an autonomous mobile robot, the work developed has to be integrated with other modules, such as a navigation module. In the tests performed, the trajectory was defined by the user, but a more interesting approach would be if the trajectory was

defined autonomously by the robot and it was dependent on the UVC dose value of the cells of the map and on the geometry of the surfaces of the environment. With this approach, the robot would autonomously move to areas that have not reached the required UVC dose, which means that are not yet disinfected. However, even with this approach, some shadowed areas could still not be disinfected.

References

[1] Wladyslaw Kowalski. *Ultraviolet Germicidal Irradiation Handbook*. 2009.

[2] ICNIRP. Guidelines on limits of exposure to ultraviolet radiation of wavelengths between 180 nm and 400 nm (incoherent optical radiation). *Health Phys*, 2004.

[3] Armin Hornung, Kai Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34, 04 2013.

[4] A. Vyshnavi, A. Naga Sri Manasa, Ch. Sesha Sai Hamsika, and P. Shalini. Uv disinfection robot with automatic switching on human detection. *EAI Endorsed Transactions on Internet of Things*, 6(23), 09 2020.

[5] Tomoaki Fukui, Takahiro Niikura, Takahiro Oda, Yohei Kumabe, Hiroyuki Ohashi, Masahiro Sasaki, Tatsushi Igarashi, Makoto Kunisada, Nozomi Yamano, Keisuke Oe, Takahiko Matsushita, Shinya Hayashi, Chikako Nishigori, and Ryosuke Kuroda. Exploratory clinical trial on the safety and bactericidal effect of 222-nm ultraviolet c irradiation in healthy humans. *PLOS ONE*, 15:e0235948, 08 2020.

[6] Nozomi Yamano, Makoto Kunisada, Sachiko Kaidzu, Kazunobu Sugihara, Aiko Nishiaki-Sawada, Hiroyuki Ohashi, Ai Yoshioka, Tatsushi Igarashi, Akihiro Ohira, Masaki Tanito, and Chikako Nishigori. Long-term effects of 222-nm ultraviolet radiation c sterilizing lamps on mice susceptible to ultraviolet radiation. *Photochemistry and Photobiology*, 96, 03 2020.

[7] Vijay Kalyani, Prachi Mathur, Nupur Makwana, and Nidhi Singhal. Study on coronavirus (covid-19) and how uvc light helps to destroy it and its applications. 7:2394–8124, 06 2020.

[8] Wenjun Sun Ernest R. Blatchley, Brian Petri. Sars-cov-2 uv dose-response behavior. Accessed on January, 2021. URL: https://iuva.org/resources/covid-19/SARS%20CoV2%20Dose%20Response%20White%20Paper.pdf.

[9] James R. Bolton and Christine A. Cotton. *Ultraviolet Disinfection Handbook (1st Edition)*. American Water Works Association (AWWA), 2008.

[10] N. Lindow, D. Baum, M. Leborgne, and H. Hege. Interactive visualization of rna and dna structures. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):967–976, 2019.

[11] IUVA International Ultraviolet Association. Iuva fact sheet on uv disinfection for covid-19. Accessed on January, 2021. URL: https://iuva.org/IUVA-Fact-Sheet-on-UV-Disinfection-for-COVID-19.

[12] William Rutala and David Weber. Hicpac guideline for disinfection and sterilization in healthcare facilities. *Facilities*, 01 2008.

[13] Milad Raeiszadeh and Babak Adeli. A critical review on ultraviolet disinfection systems against covid-19 outbreak: Applicability, validation, and safety considerations. *ACS Photonics*, 7, 10 2020.

[14] Brahmaiah Pendyala, Ankit Patras, and Doris D'Souza. Genomic modeling as an approach to identify surrogates for use in experimental validation of sars-cov-2 and hunovs inactivation by uv-c treatment. 06 2020.

[15] Martin Hessling, Katharina Hönes, Petra Vatter, and Christian Lingenfelder. Ultraviolet irradiation doses for coronavirus inactivation -review and analysis of coronavirus photoinactivation studies ultraviolette bestrahlungsdosen für die inaktivierung von coronaviren -review und analyse von coronavirusinaktivierungsstudien. *GMS Hygiene and Infection Control*, 15, 05 2020.

[16] Markus Eickmann, Ute Gravemann, Wiebke Handke, Frank Tolksdorf, Stefan Reichenberg, Thomas Müller, and Axel Seltsam. Inactivation of ebola virus and middle east respiratory syndrome coronavirus in platelet concentrates and plasma by ultraviolet c light and methylene blue plus visible light, respectively: Ebov and mers-cov inactivation. *Transfusion*, 58, 05 2018.

[17] Markus Eickmann, Ute Gravemann, Wiebke Handke, Frank Tolksdorf, Stefan Reichenberg, Thomas Müller, and Axel Seltsam. Inactivation of three emerging viruses – severe acute respiratory syndrome coronavirus, crimean–congo haemorrhagic fever virus and nipah virus – in platelet concentrates by ultraviolet c light and in plasma by methylene blue plus visible light. *Vox Sanguinis*, 115, 01 2020.

[18] Hiroaki Kariwa, Nobuhiro Fujii, and Ikuo Takashima. Inactivation of sars coronavirus by means of povidone-iodine, physical conditions and chemical reagents. *Dermatology (Basel, Switzerland)*, 212 Suppl 1:119–23, 02 2006.

[19] Shu-Ming Duan, Xin-Sheng Zhao, Rui-Fu Wen, Jing-Jing Huang, Guo-Hua Pi, Su-Xiang Zhang, Jun Han, Sheng-Li Bi, Li Ruan, and Xiao-Ping Dong. Stability of sars coronavirus in human specimens and environment and its sensitivity to heating and uv irradiation. *Biomedical and environmental sciences : BES*, 16:246–55, 10 2003.

[20] Shashank Patil, Shirahatti Prithvi, and Ramith Ramu. The pathogenicity of mers-cov, sars-cov and sars-cov-2: A comparative overview. *Research Journal of Biotechnology*, 16:182–192, 01 2021.

[21] Edward Patterson, Tessa Prince, Enyia Anderson, Aitor Casas-Sánchez, Shirley Bonner, Cintia Cansado-Utrilla, Tom Solomon, Michael Griffiths, Alvaro Acosta-Serrano, Lance Turtle, and Grant Hughes. Methods of inactivation of sars-cov-2 for downstream biological assays. *The Journal of infectious diseases*, 222, 08 2020.

[22] Andrea Bianco, Mara Biasin, Giovanni Pareschi, Adalberto Cavalleri, C. Cavatorta, Claudio Fenizia, Paola Galli, Luigi Lessio, Manuela Lualdi, Edoardo Redaelli, Irma Saulle, Daria Trabattoni, Alessio Zanutta, and Mario Clerici. Uv-c irradiation is highly effective in inactivating and inhibiting sars-cov-2 replication. 06 2020.

[23] Hiroko Inagaki, Akatsuki Saito, Hironobu Sugiyama, Tamaki Okabayashi, and Shouichi Fujimoto. Rapid inactivation of sars-cov-2 with deep-uv led irradiation. *Emerging Microbes & Infections*, 9:1–8, 07 2020.

[24] Christiane Heilingloh, Ulrich Aufderhorst, Leonie Schipper, Ulf Dittmer, Oliver Witzke, Dongliang Yang, Xin Zheng, Kathrin Sutter, Mirko Trilling, Mira Alt, Eike Steinmann, and Adalbert Krawczyk. Susceptibility of sars-cov-2 to uv irradiation. *American Journal of Infection Control*, 48, 08 2020.

[25] Nadia Storm, Lindsay McKay, Sierra Downs, Rebecca Johnson, Dagnachew Birru, Marc de samber, Walter Willaert, Giovanni Cennini, and Anthony Griffiths. Rapid and complete inactivation of sars-cov-2 by ultraviolet-c irradiation. *Scientific Reports*, 10, 12 2020.

[26] Hiroki Kitagawa, Toshihito Nomura, Tanuza Nazmul, Omori Keitaro, Norifumi Shigemoto, Takemasa Sakaguchi, and Hiroki Ohge. Effectiveness of 222-nm ultraviolet light on disinfecting sars-cov-2 surface contamination. *American journal of infection control*, 08 2020.

[27] J. A. Guerrero-Beltr·n and G. V. Barbosa-C·novas. Advantages and limitations on processing foods by uv light. *Food Science and Technology International*, 10(3):137–147, 2004.

[28] Guettari Moez, Ines Gharbi, and Samir Hamza. Uvc disinfection robot. *Environmental Science and Pollution Research*, 98:434–441, 10 2020.

[29] ISO. Iso 15858:2016 uv-c devices — safety information — permissible human exposure. Accessed on January, 2021. URL: https://www.iso.org/obp/ui/#iso:std:iso:15858:ed-1:v1:en.

[30] John Boyce, Nancy Havill, and Brent Moore. Terminal decontamination of patient rooms using an automated mobile uv light unit. *Infection control and hospital epidemiology : the official journal of the Society of Hospital Epidemiologists of America*, 32:737–42, 08 2011.

[31] Jesse Cooper, Elizabeth Bryce, George Astrakianakis, Aleksandra Stefanovic, and Karen Bartlett. Efficacy of an automated ultraviolet c device in a shared hospital bathroom. *American journal of infection control*, 44, 08 2016.

[32] Marcel Bentancor and Sabina Vidal. Programmable and low-cost ultraviolet room disinfection device. *HardwareX*, 4:e00046, 11 2018.

[33] Maria Guirguis. Robot search and rescue a comparison of 3d mapping techniques. 02 2011.

[34] Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. The MIT Press, 2nd edition, 2011.

[35] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. volume 1, pages 321–328, 01 2000.

[36] Nikos Mitsou and Costas Tzafestas. *An Introduction to the Problem of Mapping in Dynamic Environments*, pages 295–314. 08 2008.

[37] Jianhao Du, Meiqin Liu, and Weihua Sheng. Efficient exploration for real-time robot indoor 3d mapping. 07 2015.

[38] Julian Ryde and Huosheng Hu. Mobile robot 3d perception and mapping with multi-resolution occupancy lists. 01 2007.

[39] Levente Tamas and Lucian Goron. 3d map building with mobile robots. pages 134–139, 07 2012.

[40] David Portugal, André Araújo, and Micael S. Couceiro. *A Guide for 3D Mapping with Low-Cost Sensors Using ROS*, pages 3–23. Springer International Publishing, Cham, 2020.

[41] Timo Korthals, Julian Exner, Thomas Schöpping, and Marc Hesse. Semantical occupancy grid mapping framework. 09 2017.

[42] David Silva. Grid-based representations in mobile robotics using multisensory data. Master's thesis, Universidade de Coimbra, 2016.

[43] Amiraj Dhawan, Parag Oak, Rahul Mishra, and George Puthanpurackal. Path based mapping technique for robots. *International Journal of Advanced Research in Artificial Intelligence*, 2, 12 2013.

[44] Vassilis Varveropoulos. Robot localization and map construction using sonar data. Accessed on January, 2021. URL: http://rossum.sourceforge.net/papers/Localization/PosPosterv4.pdf.

[45] Jinwoo Choi, Minyong Choi, Sang Nam, and Wan Chung. Autonomous topological modeling of a home environment and topological localization using a sonar grid map. *Auton. Robots*, 30:351–368, 05 2011.

[46] Jon Howell. A robot map-creation algorithm. Accessed on January, 2021. URL: http://www.jonh.net/~jonh/robots/mapping/submitted-paper.html.

[47] Xiaoxiao Zhu, Qixin Cao, Hiroshi Yokoi, and Yinlai Jiang. Large scale indoor 3d mapping using rgb-d sensor. volume 9834, pages 313–321, 08 2016.

[48] Sining Yang, Shaowu Yang, and Xiaodong Yi. Real-time globally consistent 3d grid mapping. 12 2017.

[49] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. pages 2276–2282, 10 2006.

[50] Nicolò Valigi. Benchmarking occupancy mapping libraries. Accessed on January, 2021. URL: https://nicolovaligi.com/occupancy-mapping-benchmarks.html.

[51] H. Wang, C. Wang, and L. Xie. Lightweight 3-d localization and mapping for solid-state lidar. *IEEE Robotics and Automation Letters*, 6(2):1801–1807, 2021.

[52] Optimizing the intel realsense lidar camera l515 range. Accessed on June, 2021. URL: https://www.intelrealsense.com/optimizing-the-lidar-camera-l515-range/.