UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

Bruno Locks Floriani

# A NOVEL FRAMEWORK FOR MODEL-BASED DISCRETE COVERAGE PATH PLANNING FOR AUVS

Florianópolis
2019

Bruno Locks Floriani

# A NOVEL FRAMEWORK FOR MODEL-BASED DISCRETE COVERAGE PATH PLANNING FOR AUVS

Tese submetida ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina para a obtenção do título de doutor em Engenharia Mecânica.
Orientador: Prof. Henrique Simas, Dr. Eng.
Coorientador: Prof. Lucas Weihmann, Dr. Eng.

Florianópolis
2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Floriani, Bruno Locks
A NOVEL FRAMEWORK FOR MODEL-BASED DISCRETE COVERAGE PATH PLANNING FOR AUVS / Bruno Locks Floriani ; orientador, Henrique Simas, coorientador, Lucas Weihmann, 2019.
92 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Mecânica, Florianópolis, 2019.

Inclui referências.

1. Engenharia Mecânica. 2. Model-Based Coverage Path Planning. 3. View Planning. 4. Autonomous Underwater Vehicles. 5. Inspection. I. Simas, Henrique. II. Weihmann, Lucas. III. Weihmann, Lucas. IV. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. V. Título.

Bruno Locks Floriani

# A NOVEL FRAMEWORK FOR MODEL-BASED DISCRETE COVERAGE PATH PLANNING FOR AUVS

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof.ª Sílvia Silva da Costa Botelho, Dra.
Universidade Federal do Rio Grande

Prof. Daniel Martins, Dr. Eng.
Universidade Federal de Santa Catarina

Prof. Pablo Andretta Jaskowiak, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de doutor em Engenharia Mecânica.

Assinado de forma digital por
Jonny Carlos da Silva:51451506449
Dados: 2019.10.15 09:54:47 -03'00'

Prof. Jonny Carlos da Silva, Dr. Eng.
Coordenador do Programa

Assinado de forma digital por Henrique
Simas:80019463987
DN: cn=Henrique Simas:80019463987, ou=UFSC -
Universidade Federal de Santa Catarina, o=ICPEdu
Dados: 2019.10.14 20:40:22 -03'00'

Prof. Henrique Simas, Dr. Eng.
Orientador

Florianópolis, 19 de março de 2019.

Este trabalho é dedicado a todas as pessoas que seguem a se indignar sem nunca se resignar com as injustiças.

## AGRADECIMENTOS

A Deus por por seu amor e misericórdia Aos meus pais, Claudio e Maria Teresa, aos meus irmãos, Karine e Claudio Neto, que sempre me apoiaram e me incentivaram nos estudos e, sobretudo, são exemplos de vida a serem seguidos. aos meus sobrinhos, Gabriel e Lucas pelas incontáveis horas de distração e relaxamento.

Aos meus orientadores, prof. Henrique Simas e Lucas Weihmann, pela orientação, paciência, conselhos e empenho durante todo o doutorado.

Aos amigos, colegas e bolsistas do Laboratório de Robótica Prof. Raul Guenther e do Laboratório de Simulação Naval. Ao professor Perê Ridão e ao pós-doutorando Narcís Palomeras que me orientaram durante meu estágio de doutorado na Universitat de Girona.

Aos amigos e colegas de CIRS (Centre d’investigació en Robòtica Submarina) e do VICORB (Institut de recerca en visió per computador i robòtica), especialmente a Dina Youakim, Juan David Hernández Vega, Khadidja Himri dentre outros que muito me auxiliaram e me ensinaram durante minha estadia na Espanha.

A CAPES, ao CNPq e ao edital Ciências do Mar II pelo apoio financeiro.

*Experience is not what happens to you;*
*it’s what you do with what happens to you.*
(Aldous Huxley)

# RESUMO

A inspeção subaquática tem um papel fundamental em áreas como a biologia marinha, arqueologia subaquática, exploração de hidrocarbonetos, instalações industriais como represas, pontes ou áreas de turbinas eólicas offshore e, até mesmo, segurança nacional. Geralmente as inspeções são realizadas por mergulhadores especializados ou veículos operados remotamente (ROVs). No entanto o uso de veículos subaquáticos autônomos (AUVs) pode reduzir drasticamente os custos de operação, além de oferecer operações mais seguras em comparação com os demais métodos. Entretanto, os AUVs têm sido empregados, quase que exclusivamente, para batimetria ou outros tipos de investigação 2D, principalmente devido a simplicidade das trajetórias executadas (geralmente limitadas para seguir o perfil do fundo do mar a uma certa altitude). Isto impossibilita o uso de AUVs, com segurança, para inspecionar áreas com alto relevo ou estruturas complexas. As missões de inspeção, normalmente envolvem o planejamento do caminho de cobertura, que consiste em gerar um caminho livre de colisão que garanta a cobertura completa de uma região de interesse. O planejamento do caminho de cobertura (CPP) nas tarefas de inspeção pode ser dividido em duas categorias: discreta e contínua. O CPP discreto busca encontrar o número mínimo de pontos de vista para cobrir completamente uma região de interesse e depois conectar esses pontos. Muitos dos algoritmos de inspeção para estruturas complexas são discretos. Por outro lado, o CPP contínuo realiza uma detecção ininterrupta ao longo da rota a ser seguida. Para executar um CPP, deve-se levar em consideração as restrições do veículo e do sensor, bem como do ambiente. Nesta tese, é proposta uma nova abordagem de CPP discreto baseada em um modelo que divide a tarefa em duas etapas de otimização: encontrar o menor número de pontos de vista para cobrir o objeto e a ordem e o caminho para alcançá-los. Essa abordagem permite obter resultados satisfatórios em pouco tempo para ambientes complexos. O método proposto é avaliado em simulações usando o Gazebo e o Girona 500 AUV. A abordagem empregada considera que não é realista considerar que 100 % de cobertura é sempre possível e, às vezes, simplesmente não é desejável. Além disso, propõem-se um algoritmo para selecionar os pontos de vista com base em um índice de qualidade de dados proposta, buscando minimizar o ruído nos dados coletados. Também estima-se a cobertura levando em consideração o pior caso de incerteza, o que é um problema-chave no planejamento da cobertura e do planejamento de vista. A incerteza de posição é especialmente difícil no ambiente subaquático porque a falta de GPS ou de outros sistemas globais de localização faz com que o erro de posição dos sistemas de sensores cresça durante a inspeção. Por isso, avaliou-se os algoritmos de planejamento de caminhos de cobertura discreta utilizando um modelo. A estimativa de cobertura com ou sem incerteza foi testada e produziu resultado satisfatório, embora estimar a cobertura máxima seja um processo exaustivo. **Palavras-chave**: Planejamento de vista. Model-Based. Planejamento de caminhos para cobertura. AUV. Robótica Subaquática. Inspeção.

# RESUMO EXPANDIDO

## Introdução

Os mares, lagos, rios e oceanos constituem a maior parte da superfície terrestre, tendo um papel fundamental no desenvolvimento e manutenção da vida, sendo fonte de alimentos, energia, rota comercial, além desses corpos d'água desempenharem importante papel na regulação do clima no planeta Terra. Apesar da grande importância dos mares e oceanos e de recentemente o conhecimento sobre estes ambientes ter crescido, pode-se dizer que o lado escuro da Lua é mais conhecido do que o fundo do mar. Além disso, nos últimos anos muitas estruturas industriais têm sido instaladas nos oceanos, com o passar do tempo é necessário inspecionar estas estruturas para avaliar a necessidade de manutenção ou mesmo de reparos. Neste contexto, o uso de veículos subaquáticos não-tripulados (UUVs) tem recebido grandes investimentos, buscando realizar tarefas de inspeção e intervenção mais baratas e seguras do que com mergulhadores ou veículos tripulados. Mas as tarefas de inspeções são ainda, geralmente, realizadas por mergulhadores especializados ou veículos operados remotamente (ROVs), por outro lado, o uso de veículos subaquáticos autônomos (AUVs) pode reduzir drasticamente os custos de operação em comparação com os demais métodos. No entanto, os AUVs têm sido empregados, quase que exclusivamente, para atividades de investigação 2D, essencialmente devido a simplicidade das trajetórias executadas (geralmente limitadas a seguir o perfil do fundo do mar a uma certa altitude), o que impossibilita o uso de AUVs, com segurança, para inspecionar áreas com alto relevo ou estruturas complexas. As missões de inspeção, normalmente envolvem o planejamento do caminho de cobertura, que consiste em gerar um caminho livre de colisão que garante a cobertura completa de uma região de interesse. O planejamento do caminho de cobertura (CPP) nas tarefas de inspeção podem ser divididos em duas categorias: discreta e contínua. O CPP discreto busca encontrar o número mínimo de pontos de vista para cobrir completamente uma região de interesse e depois conectar esses pontos, muitos dos algoritmos de inspeção para estruturas complexas são discretos. Por outro lado, o CPP contínuo realiza uma detecção ininterrupta ao longo da rota a ser seguida.

## Objetivos

A partir das questões levantadas, esta tese tem por objetivo desenvolver uma metodologia de planejamento de caminho para cobertura discreta para a inspeção de regiões com alto relevo ou estruturas complexas utilizando um AUV, levando em consideração as limitações do veículo e dos sensores.

## Metodologia

Inicialmente foi feita uma revisão da literatura, buscando por metodologias e técnicas que pudessem encontrar uma solução para o problema proposto. A partir do levantamento do estado da arte alguns algoritmos para a inspeção foram testados no problema proposto. Nesta etapa foram implementados não apenas algoritmos voltados ao ambiente submerso mas também métodos para veículos aéreos não tripulados.

Tendo em vista que nenhuma das metodologias levantadas atendia aos requisitos de projeto, se desenvolveu um algoritmo para a inspeção subaquática a partir da divisão do problema em duas partes: encontrar o menor número de pontos necessários para inspecionar uma estrutura e o caminho para percorrer as vistas necessárias. Nesta tese partiu-se de uma abordagem *sampling-based* para resolver os dois problemas propostos. Além disso, como muitas vezes não é realista considerar que 100 % de cobertura é sempre possível e, às vezes, simplesmente não é desejável, propõe-se um algoritmo para calcular a cobertura máxima possível para um certo veículo e conjunto de sensores. Também se apresenta um índice de qualidade de vista para auxiliar na seleção das vistas que geram menos ruído nos dados coletados. Por fim, se apresenta uma proposta para considerar o erro de posição no cálculo da cobertura estimada.

**Resultados e Discussão**

A metodologia proposta foi validada em simulação realizada no Gazebo, usando o Girona 500 AUV com uma unidade *multibeam* montada sobre uma unidade *pan and tilt*. Os resultados obtidos foram satisfatórios, mostrando que o uso de pontos de vistas gerados arbitrariamente tem grande aplicação na engenharia, já que selecionando as melhores vistas a partir do algoritmo proposto de um conjunto inicial de 500 pontos de vista gerados aleatoriamente cobre mais de 95% do que seria coberto a partir de um *grid* de 300,000 pontos. Ademais, o uso de *grid* em grandes ambientes é inviável, diferentemente do algoritmo proposto.

**Considerações Finais**

Face a grandeza do problema e os resultados bastante promissores encontrados, entende-se que existe um grande campo de aplicação para algoritmos *sampling-based* em problemas de inspeção, seja no ambiente subaquático, ou no ambiente aéreo. Além disso, vê-se um enorme potencial de aplicação real da metodologia proposta, visto os resultados obtidos em simulação.

**Palavras-chave**: Planejamento de vista. Model-Based. Planejamento de caminhos para cobertura. AUV. Robótica Subaquática. Inspeção.

# ABSTRACT

The inspection of underwater structures plays a key role in areas like geology, marine biology, underwater archaeology, industrial facilities such as oil fields, dams, bridges or offshore wind turbine areas and, even homeland security. These inspections are generally performed by specialized divers or remotely operated vehicles (ROVs) but the use of autonomous underwater vehicles (AUVs) can highly reduce the costs to perform these tasks. Despite the progressive adoption of AUVs to carry out 2D surveys, the simplicity of the trajectories executed (usually limited to follow the profile of the seabed at a certain altitude) makes it impossible to safely use AUVs to inspect areas with a high 3D relief or complex structures. Typically, inspection missions involve coverage path planning, which is the task responsible to generate a collision-free path that ensures the complete coverage of a region of interesting in order to gather highly accurate data to allow the inspection. Coverage path planning (CPP) in inspection tasks and bathymetry mapping can be divided into two categories: discrete and continuous. Discrete CPP looks to find the minimal number of viewpoints to completely cover a region of interest and then connect those points; many of the inspection algorithms to complex structures are discrete. In contrast, continuous CPP perform an uninterrupted sensing along the route to be followed and most of AUV surveys are continuous CPP. To perform a CPP we must take into account the vehicle and sensor constraints, as well as, the environment. In this thesis we proposed a novel model-based discrete CPP approach that split this task in two optimizing steps: the number of viewpoints to cover the object and the order and path to reach them. This approach allows us to obtain satisfactory results in a short time for large and complex environments. The proposed method is evaluated in simulations using Gazebo and the Girona 500 AUV. Our approach takes for that it is unrealistic to consider that 100% coverage is always possible and sometimes it is just not desirable. Even more, we proposed an algorithm to select the viewpoints based on a data quality index proposed by us, looking to minimize the noise on the multibeam data gathered. We, also, estimate the coverage taking into account the worst case of pose uncertainty which is a key problem in coverage and viewpoint path planning. The pose uncertainty it is specially difficult in the underwater environment because the lack of GPS or others global localization systems makes the position error of the sensors systems grows up during the inspection. Because of it we must evaluate on model-based discrete coverage path planning algorithms. The coverage estimation with or without uncertainty have been tested and produced satisfactory result although estimate the maximum coverage is a very time consuming process. **Keywords**: View Planning. Model-Based. Coverage Path Planning. AUV. Underwater Robotics and Inspection.

# LIST OF FIGURES

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| 2.5D | Two-and-a-half-dimensional Space |
| 2D | Two-dimensional Space |
| 3D | Three-dimensional Space |
| AGP | Art Gallery Problem |
| AUV | Autonomous Underwater Vehicle |
| CPP | Coverage Path Planning |
| DSM | Digital Surface Model |
| EST | Expansive Space Trees |
| GPS | Global Positioning System |
| IMR | Inspection, Maintenance and Repair |
| LKH | Lin-Kernighan-Helsgaun Heuristic |
| MAV | Micro Aerial Vehicle |
| NBV | Next Best View |
| NSGA | Non-dominated Sorting Genetic Algorithm |
| OMPL | Open Motion Planning Library |
| PDDL | Planning Domain Definition Language |
| PRM | Probabilistic Roadmap Method |
| PRP | Possible Reach Position |
| pSBL | Parallel Single-query Bi-directional Lazy collision checking planner |
| ROV | Remotely Operated Vehicle |
| RPP | Randomized Path Planner |
| RRT | Rapidly-exploring Random Tree |
| SLAM | Simultaneous Localization and Mapping |
| SPARS | Sparse Roadmap Spanners |
| STOMP | Stochastic Trajectory Optimization for Motion Planning |
| TFD | Temporal Fast Downward |
| TSP | Travel Salesman Problem |
| USD | United States Dollar |

## SUMÁRIO

# 1 INTRODUCTION

Over than seventy percent of the Earth's surface is covered by the oceans, which support a great diversity of life and ecosystems,it is source of food, renewable energy, the main trade route and, play a key role in controlling the Earth's climate. Under the oceans are huge oil and gas reserves (SICILIANO; KHATIB, 2008).

Our scientific knowledge of the underwater environment is expanding rapidly thanks to a variety of technologies (SICILIANO; KHATIB, 2008).

Although the large importance of the oceans, we know less about them than we know about the dark side of the Moon.

Moreover, many industrial facilities such oil fields, dams, bridges, offshore wind turbines were built in the oceans along the years. Such equipaments become old and should be inspected to infer an updated model, or scan for risks and hazards, in order to plan maintenance (SOUZA, 2006; SICILIANO; KHATIB, 2008; ZANONI; BARROS, 2015; JACOBI, 2015; BIRCHER *et al.*, 2015). The underwater inspection even plays a key role in areas like geology, marine biology, underwater archeology and homeland security (SOUZA, 2006). Besides, the underwater Inspection, Maintenance and Repair (IMR) service it is a United States Dollar (USD) 3-billion market (WHITFIELD, 2018).

Usually, inspection missions involve coverage path planning, work space reconstruction and the actual inspection of the structure. Coverage path planning is the task to generate a collision-free path that ensures the complete coverage of a region of interesting in order to gather highly accurate data to allow the model reconstruction. The fusion of the collected data in a map is the aim of the reconstruction task (ALMADHOUN *et al.*, 2016).

In general, underwater inspection used to be performed by specialized divers. Although the divers operation is very restricted because the pressure, temperature, depth limitations among others. More recently much effort has been invested in the development of UUVs as well as automatically and high efficiency inspection systems, which is a task in the active vision problem. Comparing to manned vehicles, the UUVs can reach smaller regions and have a cheaper operation. The UUVs often return superior data at reduced costs (ANTONELLI; FOSSEN; YOERGER, 2008).

The UUVs usually are organized in two main groups according to the autonomy: Remotely Operated Vehicle (ROV) and Autonomous Underwater Vehicle (AUV).

The ROVs are unmanned underwater remotely operated vehicles characterized by a physical connection with the base, through this connection energy is sent to the vehicle and data is exchanged. The first ROV was built in 1953, in the 1960s and 1970s were used to military applications and in the 1980s to commercial and scientific purposes (SICILIANO; KHATIB, 2008). Fig. 1 shows the Subastian ROV being launched at sea.

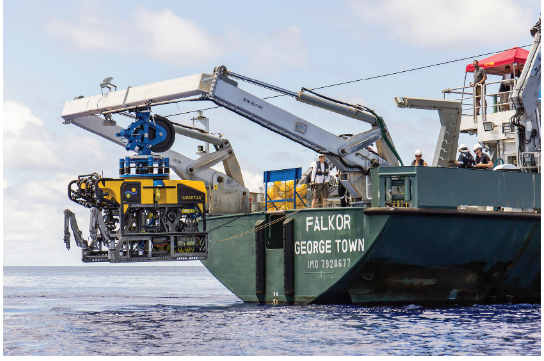


Figura 1 – Subastian ROV being launched for a research mission. Image Credit: Schmidt Ocean Institute

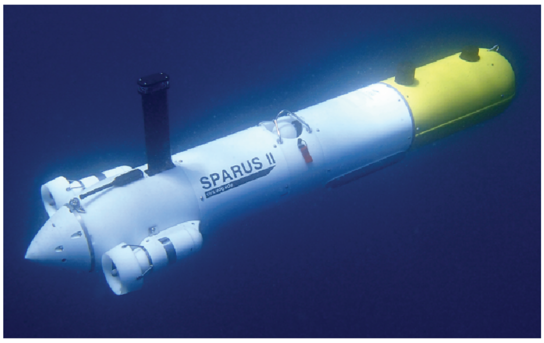


Figura 2 – Sparus II at sea. Image Credit: Computer Vision and Robotics Research Group (VICOROB)

ROVs enable exploring deep-sea regions far beyond reach of human divers. However the dynamic effects produced by the tether[1], the requirement of a qualified crew in constant contact with the vehicle among others, restrict the operation of the ROV mainly to the offshore oil industry (ANTONELLI; FOSSEN; YOERGER, 2008; GALCERAN, 2014).

To handle with the ROV limitations, such as the high operational cost and the limitations caused by the tether, some efforts have been invested to provide some level of autonomy to the ROVs (GALCERAN, 2014). This was achieved by providing to the vehicle it's own source of energy and capability to plan tasks and take decisions face to unexpected events. These novel kind of unmanned vehicle is called AUV. Figure 2 shows an AUV in operation.

More recently AUVs became largely used to map the ocean floor using optical and/or acoustic sensor for applications such as dam inspection, marine geology, underwater archaeology and Mine CounterMeasures (MCM) just to mention a few, which just become possible thanks the important advances in the technology in the last decades. A key application of AUVs is the bathymetric mapping for charting the ocean floor, which is essential to support safe navigation for example. In addiction, AUVs usually perform near-bottom surveys producing a high resolution map (GALCERAN, 2014).

[1] Umbilical cable that connects the ROV with the base.

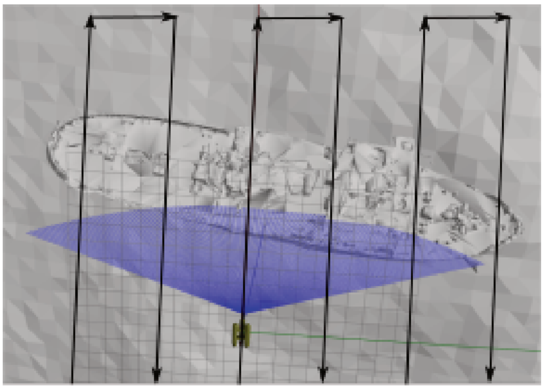


Figura 3 – Typical *mowing the lawn* AUV survey over a shipwreck named Boreas (Pálamos, Spain) using Girona 500 AUV.

The operations with AUVs are limited to Two-dimensional Space (2D) and Two-and-a-half-dimensional Space (2.5D)[2] surveys, following a predefined pattern trajectory (that use to be follow the profile of the seabed at a certain altitude), which makes it impossible to safely use them to inspect areas with a high Three-dimensional Space (3D) relief, non-convexities, pronounced curves or geometrically complex structures (ANTONELLI; FOSSEN; YOERGER, 2008). Figure 3 shows a typical AUV survey mission, in which the path is composed by a sequence of straight lines connected in a mowing the lawn manner. This approach is used in most of the commercial AUVs.

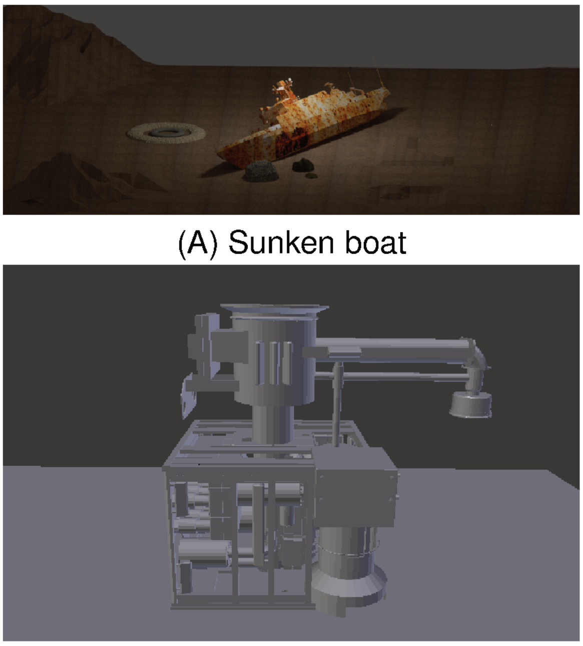
In the last years, some efforts have been invested to inspect complex 3D underwater structures such as oil and gas industrial facilities, dams, bridges, offshore wind turbines, sunken boats as well as ship hulls because most of those tasks still being performed by divers and ROVs. Fig. 4[A] shows a sunken boat similar to a scenario in Boreas (Spain) and 4[B] shows a subsea tree structure which is part of a larger oil industrial facility.

Besides all the obvious technical challenges in the underwater environment such as, the lack of a ubiquitous absolute positioning (like a Global Positioning System (GPS)), sensors limitation, the reduced visibility, currents, navigation drift and all sorts of unexpected disturbances, creating a plan to structural inspection is essential.

A motion plan to structural inspection must cover all the interesting parts or the hole structure with the AUV sensor's minimizing the energy consuming since the AUV has no external power (ELLEFSEN; LEPIKSON; ALBIEZ, 2016).

Coverage Path Planning (CPP) to inspection tasks as well as to bathymetry mapping can be divided into two categories: discrete and continuous. Discrete CPP looks to find the minimal number of viewpoints to complete cover a region of interesting and then connect those points. In contrast, continuous CPP perform a uninterrupted sensing along the route to be followed. Most of the typical AUV surveys previous mentioned are continuous CPP. On the other hand, many of the inspection algorihtms to inspect

[2] A surface is said 2.5D if for all pair $(x, y)$ there exist only one $z$ corresponding.



(A) Sunken boat

(B) Subsea tree structure

Figura 4 – Original environment representation. Extracted from Floriani *et al.* (2017)

complex structure are discrete, such as Schmid *et al.* (2012), Cashmore *et al.* (2013), Ososinski e Labrosse (2014), Bircher *et al.* (2015) e Alexis *et al.* (2015).

To automate the inspection we need to take into account not only the vehicle and the structure of interest but the sensors. A lack of sensors, methods and technologies could be used as visual methods, sonars/echo sounders, magneto-metric sensors and laser scanning (MAI *et al.*, 2016).

The goal of this thesis is to propose a model-based discrete CPP approach to inspect complex structure, Fig. 5 shows the Girona 500 AUV equipped with a multibeam mounted over a pan-and-tilt unit inspecting a tree structure. Several discrete and continuous coverage path planners for inspection applications have being proposed using Unmanned Aerial Vehicles (UAVs) (CASHMORE *et al.*, 2013; ALEXIS *et al.*, 2015), as well as methods to continuous coverage path planning for arable fields (JIN; TANG, 2011) and, even, to subsea coverage (GALCERAN, 2014). For some specific applications on the underwater inspection of complex structures have being proposed, as ship hull inspection (ENGLOT, 2012), oil and gas facilities (ELLEFSEN; LEPIKSON; ALBIEZ, 2016).

This thesis addresses the *model-based discrete coverage path planning problem* in the aforementioned structural underwater inspection problem, which is the task to define the viewpoints to cover the region of interest and, connecting them to minimizing the travelled distance. This task is fundamental in many robotic and industrial applications, such as search, recovery, inspection, reconstruction. As we will see in Chapter 2 a large effort has been invested to address the 2D and 2.5D continuous CPP on aerial,

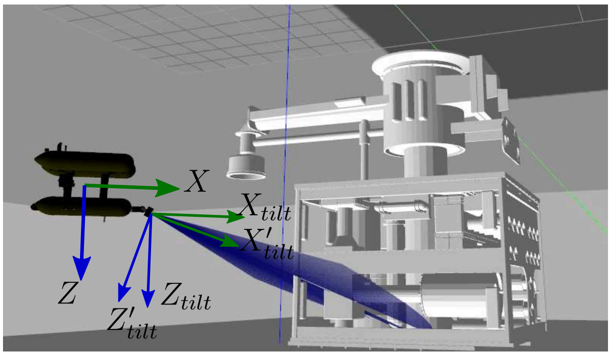


Figura 5 – Girona 500 AUV with its coordinate frame ($X$, $Z$), tilt frame ($X_{tilt}$, $Z_{tilt}$) and multibeam frame ($X'_{tilt}$, $Z'_{tilt}$) inspceting a tree structure.

terrestrial and underwater environments, as well as, to the discrete CPP on aerial and terrestrial environments. Although, discrete CPP for high 3D relief regions have been addressed for few researchers and those approaches are not able to deal with different sensors and were not tested in various scenarios. In addition, some of those algorithms are highly time consuming and hard to implement.

## 1.1 OBJECTIVE OF THE THESIS

The main objective of this thesis is stated as follows:

*To develop a discrete coverage path planning technique for autonomous underwater vehicles to inspect complex structures, taking into account the sensor limitations (range).*

### 1.1.1 Specific Objectives of the Thesis

The main objective of the thesis can be separated into the following specific objectives:

- **REVIEW OF THE VIEW PLANNING LITERATURE.** To carry out a review of the most relevant works in the coverage path planning literature and to identify requirements and limitations of the proposed methods to application in the underwater domain;
- **3D DISCRETE CPP FOR AUVS.** To propose a model-based discrete coverage path planning method for inspection underwater complex structures or environments with a strong 3D relief;
- **ESTIMATE THE MAXIMUM COVERAGE.** To propose an algorithm to estimate the maximum coverage reachable taking into account the vehicle and sensor constraints;

- **INDEX A DATA QUALITY OF THE SAMPLES.** To propose a index data quality of the samples that provides a metric to select the best samples to be used in the discrete CPP;
- **DISCRETE CPP UNDER POSITION UNCERTAINTY OF THE VEHICLE.** To propose a model-based discrete CPP algorithm that estimate the minimum coverage considering the worst case of the position uncertainty.

## 1.2 OVERVIEW AND CONTRIBUTIONS OF THE THESIS

This manuscript is organized presenting the incremental and progressive development of this thesis. In chapter 2 we present a review of the most recent and relevant methods, techniques, and applications of model-based coverage path planning presented in the literature focus on the discrete methods. This review points out the limitations of the methods and show directions for further research.

In chapter 3 we introduce a novel model-based discrete coverage path planning method based on octrees and random viewpoint distribution for 3D complex structures, some tests, simulations and some discussion about the results. A key innovation of this method is the use of simple approaches on an algorithm to deal with a variety of complex environments in a short time and the easily adaptation to different vehicles.

A coverage estimation method is presented in Chapter 4. We also introduce and discuss some questions about coverage estimation. This is another contribution to the field of discrete CPP presented in this thesis.

A model-based discrete CPP planning under the worst case of uncertainty in the vehicle pose is proposed in the Chapter 5.

We conclude by summarizing the work, reviewing the contributions of this thesis and, finally, identifying compelling areas for future work, in chapter 6.

# 2 STATE OF THE ART

This chapter contains a survey of previous work in coverage path planning focusing on underwater environment and related topics. Initially we introduce the path planning in general, including the classical approach and sampling-based planning. Then we discuss the coverage problem in 2D and subjects about view planning. We close this survey reviewing the latest view planning algorithms and their contributions. Finally we summarize the presented view planning methods and discuss the opening challenges in the area.

## 2.1 PATH PLANNING

A fundamental component of autonomy in mobile robots, as well as, in manipulators is the ability to use an *a priori* model or sense the environment to plan and execute tasks that require physical motion. These task could require a re-planning capability based on the real-time sensing. In any case, the decision-make procedure is driven by a path planning algorithm.

Path planning can be divided in two main categories according to their application: *start-to-goal* and *coverage path planning*. Paths use to be planned to minimize or maximize a diversity of functions, normally path planning algorithms try to minimize the traveling distance (ZELINSKY *et al.*, 1993; ENGLOT, 2012; HERNÁNDEZ VEGA *et al.*, 2017).

## 2.2 START-TO-GOAL PATH PLANNING

Start-to-goal path planning consists in a task of determining a collision free path from a start configuration to a goal configuration. At first it would seem a simple task, however involves changes in the robot's configuration, may contain different topological spaces. The first work in path planning dates back to the late 60's (NILSSON, 1969). Although, this field of study just become active after the introduction of the concept of the *configuration space* by Lozano-Perez (1983), which arise to deal with problems like arrange or move objects without collisions.

Any set of parameters which fully specify the position of all points on an object is called the configuration (q) of a moving object. Then, the configuration space (C-space) denoted by $\mathcal{C}$, is the set of all possible configurations that a moving object can adopt in the workspace ($\mathcal{W}$) (LOZANO-PEREZ, 1987; HERNÁNDEZ VEGA *et al.*, 2017). As not all positions are reachable, the C-space ($\mathcal{C}$) is subdivided into *free space* $\mathcal{C}_{free}$ and *obstacle region* $\mathcal{C}_{obs}$ (LOZANO-PEREZ, 1987).

The start-to-goal path planning task consists in finding a continuous path $p : [0, 1] \rightarrow \mathcal{C}_{free}$ from $p(0) = q_{start}$ to a $p(1) = q_{goal}$ (HERNÁNDEZ VEGA *et al.*, 2017).

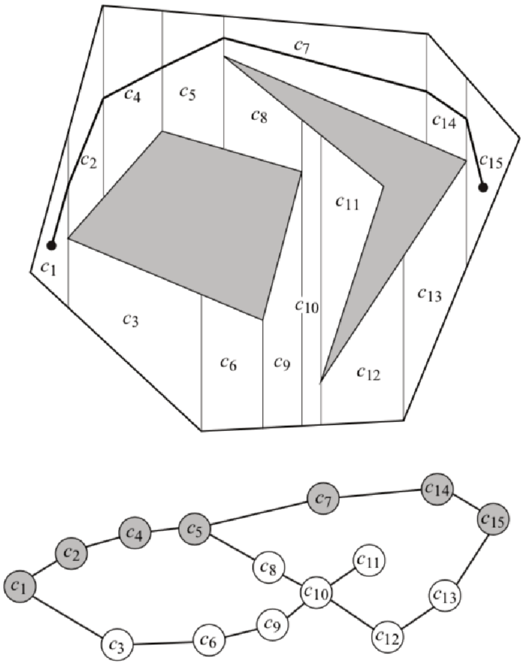


Figura 6 – Trapezoidal cell decomposition and adjacency graph, showing a planned path. Extracted from Choset (2007)

## 2.3 THE CONFIGURATION SPACE

One of the earliest and most studied approaches is the cell decomposition. It consists of decomposing the $\mathcal{C}_{free}$ into a finite collection of cells (HERNÁNDEZ VEGA *et al.*, 2017). The following step to build a graph in C-Space whose nodes represent the collision-free configuration $\mathcal{C}_{free}$ and the edges the connectivity between these cells (BARRAQUAND, Jerome; LATOMBE, 1991; ENGLOT, 2012; HERNÁNDEZ VEGA *et al.*, 2017).

There are many ways to decompose the free C-Space, such as grid-based (MORAVEC; ELFES, 1985), trapezoidal (LATOMBE, 2012), boustrophedon decomposition (CHOSET; PIGNON, 1997, 1998). Figure 6 shows a trapezoidal decomposition of the C-Space and the path planned from the adjacency graph.

Specially in grid-based algorithms the size of the cell is an important parameter, because coarse cells will allow fast solutions although will not be able to deal with narrow passages. On the other hand, the problem become intractable for finner grids (BARRAQUAND, Jerome; LATOMBE, 1991; HERNÁNDEZ VEGA *et al.*, 2017).

There are several methods to calculate an optimal path in the adjacency graph, like Dijkstra's (DIJKSTRA, 1959), A* (HART; NILSSON; RAPHAEL, 1968), Dynamic A (STENTZ, 1994), just to name some.

To minimize and sometimes avoid drawbacks of the cell decomposition approaches in the *start-to-goal* problem other methods have been proposed (HERNÁNDEZ VEGA *et al.*, 2017).

The use of potential functions is one of those alternative approaches. *Potential field* methods consists in an attractive component pose in the goal position pulling the robot form the goal and repulsive components pushing the robot away from the obstacles (KHATIB, 1986).

After define the potential function and calculate the gradient of the potential field, the start-to-goal path can be obtained by the gradient descent (KHATIB, 1986). Although it is not possible to guarantee to reach the goal position using potential field algorithms, because could exist a local minimum that not correspond to the goal (BARRAQUAND, Jérôme; LATOMBE, 1990).

To escape of the local minima of a potential field, Jérôme Barraquand e Latombe (1990) proposed the RPP (Randomized Path Planner) that uses random motion, which are Brownian motions implemented as a discrete random walk.

## 2.4 SAMPLING-BASED

An alternative philosophy for addressing the motion planning problem to the configuration space is the sampling-based motion planning. The main idea of sampling-based methods is to avoid the explicit construction of the $C_{free}$ and $C_{obs}$ which is expensive for high-dimensional (LAVALLE, 2006; ENGLOT, 2012).

To achieve this goal, it is takes advantage of fast and efficient collision detection algorithms. Initially samples are generated and tested for collision, then the samples are interconnected producing different routes. Although the large number of advantages the sampling-based methods weaker guarantee that a solution will be find (LAVALLE, 2006; HERNÁNDEZ VEGA *et al.*, 2017).

However, given that many of sampling-based methods adopt random sampling strategy usually using an uniform distribution, which is dense with probability one, the probability to find a solution if exist, converges to one as the number of samples grows. Moreover, random sampling algorithms are the easiest sampling approach to apply to C-space (LAVALLE, 2006).

Despite the widely use of randomized algorithms, sampling-based can also make use of deterministic sampling schemes. A recently in-detail survey of deterministic sampling-based methods is presented by Youakim e Ridao (2018).

Sampling-based algorithms can be classified based on their capability to solve *single* or *multi-query*.

#### 2.4.0.1 Single-query

In single query, there is just one *start-to-goal* query to the algorithm. Usually those methods are based on a incremental search, expanding a graph tree of random samples configurations from the start configuration to the goal configuration, these

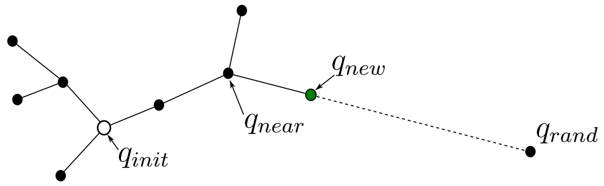


Figura 7 – RRT - extend procedure.

approach are called *unidirectional*. In *bidirectional* approaches one tree is expanded from the start configuration and another from the goal configuration until the trees meet each other at a common sample.

The earlier mentioned Randomized Path Planner (RPP) (BARRAQUAND, Jérôme; LATOMBE, 1990), commonly acknowledged as the first randomized algorithm, was a single-query algorithm. Later, many other approaches have been presented, as the Expansive Space Trees (EST) (HSU; LATOMBE; MOTWANI, 1997) and Parallel Single-query Bi-directional Lazy collision checking planner (pSBL) (Parallel Single-query Bi-directional Lazy) (SÁNCHEZ; LATOMBE, 2003), Rapidly-exploring Random Tree (RRT) (Rapidly-exploring Random Trees) (LAVALLE, 1998) and many RRT variants, like RRT-connect (KUFFNER; LAVALLE, 2000) and RRT* (KARAMAN; FRAZZOLI, 2011), to mention few of them.

RRT is commonly called as the state-of-the-art on single-query sampling-based algorithms and offers a fast and efficient approach. These algorithm consists in two main procedures: *sample* and *extend*. In the *sample* procedure a tree is incrementally built until a stop criteria is achieved and in each iteration the algorithm attempts to extend the tree onto a randomly sampled configuration ($q_{rand}$). To perform the extend procedure the nearest configuration ($q_{near}$) to $q_{rand}$ is find. Then, a path of length $\delta$ is calculated from $q_{near}$ to $q_{rand}$ using a local planner; if the path is collision-free the new configuration ($q_{new}$) is added to the tree together with the path, this procedure is shown in Fig. 7

### 2.4.0.2 Multi-query

Another group of sampling-based algorithms is focused on *multi-query*, there are multiple initial-goal queries to the algorithm keeping the robot model and the environment fixed. To solve *multi-query* it make sense perform a preprocessing phase to construct a roadmap (topological graph) that is a subset of the C-space containing the samples and the interconnection of the nearest samples checked for collision, a roadmap construction is shown in Figure 8. Then, answering path planning queries become quick. Probabilistic roadmap (PRM) (KAVRAKI *et al.*, 1996) is one of the earliest, widely used sampling-based methods (LAVALLE, 2006; ENGLOT, 2012).

Some variants of the PRM and other *multi-query* methods have been presented, like lazyPRM (BOHLIN; KAVRAKI, 2000), PRM* (KARAMAN; FRAZZOLI, 2011), Gaus-

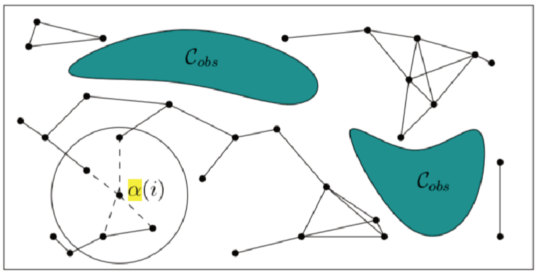


Figura 8 – A sampling-based roadmap under construction, seeking to connect each new sample, where $\alpha(i)$ denotes the $i$th sample. Extracted from LaValle (2006)

sian distribution (BOOR; OVERMARS; VAN DER STAPPEN, 1999), Sparse Roadmap Spanners (SPARS) (DOBSON; KRONTIRIS; BEKRIS, 2013) and SPARS2 (DOBSON; BEKRIS, 2013), just to mention some.

## 2.5 VIEW PLANNING

View planning is the task aiming to determine a suitable set of viewpoints associated to imaging parameters that supply a reconstruction or inspection task taking into account the sensor range. Additionally, the viewpoint is mainly responsible to generate the quality of the resulting image (TARABANIS; ALLEN; TSAI, 1995; SCOTT; ROTH; RIVEST, 2003). Reconstruction tasks didn't have any *a priori* model of the environment and require that each part have been seen at least $k$[1] times, this group of algorithm are called as exploratory view planning (or non-model-based view planning). Although, for inspection tasks there is a preexisting model at some level of fidelity which allows offline and optimized planning, these class is called as model-based view planning (SCOTT; ROTH; RIVEST, 2003).

Early works in computer vision focus on the object placement considering that the viewpoint was given and suitable for the task (TARABANIS; ALLEN; TSAI, 1995). Even more, view planning placed an emphasis only in the target object shape but in the last 15 years have been growing the number of application worried with the surface visual texture or reflectance properties as well as the shape (SCOTT; ROTH; RIVEST, 2003).

A well chose viewpoint and illumination condition ensures good image quality and minimize the measurement error (CHEN; LI; KWOK, 2011).

Determine the viewpoint is associate to determine the sensor and illumination parameter value, which are related to the type of the sensor and illumination. Sensor parameters could be *geometric* (($X$,$Y$,$Z$,$\phi$,$\theta$,$\psi$) and *optical* (the back principalpoint, the entrance pupil diameter and the focal length $f$ of the lens). There are two types of illumination parameters: *geometric* ($X$,$Y$,$Z$,$\phi$, $\theta$,$\psi$ and the shape of the illumination

[1] $k$ is the number of times that is required that each part must be seen

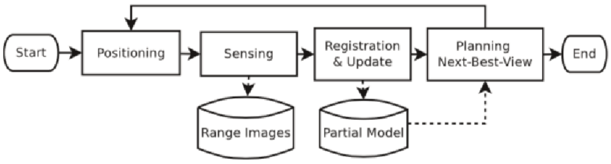


Figura 9 – Exploratory view planning flowchart. Extracted from Vasquez-Gomez, Sucar, Murrieta-Cid e Lopez-Damian (2014)

beam) and *radiometric* (the radiant intensity, the spatial distribution of intensity, the spectral distribution of intensity and the polarization of the illumination) (TARABANIS; ALLEN; TSAI, 1995).

### 2.5.1 Exploratory view planning

In exploratory view planning there is non *a priori* model, then the reconstruction process must be performed concurrently with the view planning (VASQUEZ-GOMEZ; SUCAR; MURRIETA-CID; LOPEZ-DAMIAN, 2014). In other words, the main idea of the non-model-based view planning algorithms is to increase the knowledge of the region of interesting taking into account the system constraints (CHEN; LI; KWOK, 2011).

*Next Best View (NBV)* is one of the most successful strategies in exploratory problems. Although determine the *next-best-view* is a complex problem which must satisfy the following criteria: (a) add new information, (b) the sensor's position must be reachable, (c) the surface to be seen must be in the sensor range and (d) there are overlap (VASQUEZ-GOMEZ; SUCAR; MURRIETA-CID; LOPEZ-DAMIAN, 2014).Fig. 9 shows the flowchart of an exploratory view planning.

Non-model-based algorithms can be divided in tree categories: (a) surface-based, (b) volumetric, or (c) global. Nonetheless most of the non-model-based fall into volumetric or surface-based categories and some of them combine more than one technique (SCOTT; ROTH; RIVEST, 2003).

Surface-based algorithms could be divided in tree subcategories: occlusion edge methods, contour following and parametric surface representations (SCOTT; ROTH; RIVEST, 2003). Occlusion edge method is the most used approach in exploratory view planning and is based on choose the next view using the internal occlusion edges and jumps in the edges boundary representation (MAVER; BAJCSY, 1993). Contour following methods follow the idea of “painting” the object, after locate the object follow the contour surface putting the sensor in a close proximity to the object (PUDNEY, 1994). A parametric surface method is presented by Whaite e Ferrie (1997), in that case the obtained data is segmented and infer the super ellipsoid models for each segmented part. The next best view is chosen where the ability to predict the segmented part is worst.

Different from the surface-based methods that looks to the next best view by reasoning the geometry of the structure surface, volumetric methods reasoning by volume occupied by the structure. Volumetric methods select the NBV as the viewpoint offering the largest prospective reduction in uncertainty about region of interesting.

The volumetric methods can be categorized in four groups: voxel occupancy, octree, space carving and solid geometry methods (SCOTT; ROTH; RIVEST, 2003).

Voxel occupancy models the working space and the object using a occupancy grid, where the voxel can be *occupied* or *unoccupied* (BANTA *et al.*, 1995; MASSIOS; FISHER *et al.*, 1998). Octree methods encode the voxels in a tree structure which is more efficient (CONNOLLY, 1985). The basic idea of the space carving methods is sweep the imaging workspace following the contour and avoiding the obstacles (PAPADOPOULOS-ORFANOS; SCHMITT, 1997). In solid geometry methods for each scan a surface mesh is produced and extruded along the vector of the rangefinder's sensing axis until reach the workspace boundary generating a solid representing the imaged object and the occluded volumes, then the solids are merged (REED; ALLEN, 2000).

Global view planning methods do not derive the view planning from local characteristics of the geometric data (SCOTT; ROTH; RIVEST, 2003).

### 2.5.2 Model-Based

Model-based methods consists in determine and optimize the sensor positions before the task operations, basing the view planning task on an *a priori* model of the region of interesting at some level of fidelity (SCOTT, 2009; CHEN; LI; KWOK, 2011). Many of these approaches takes into account shadow effects, object shapes and material properties and image overlapping properties for future integration (ALMADHOUN *et al.*, 2016). Although many approaches are design for a particular application becoming difficult to be applied in other tasks (CHEN; LI; KWOK, 2011).

Model-basel algorithms can be divided into (a) set theory (visibility matrix), (b) graph theory (aspect graphs) and (c) computational geometry (art gallery problem) (SCOTT; ROTH; RIVEST, 2003).

The set theory is based on the visibility and measurability matrices, visibility matrix catalogs the visibility of a discrete object surface from each viewpoint, and measurability matrix encode the structure surface points that can be measured from selected viewpoint (ENGLOT, 2012). A point in the structure surface is said viewable from a particular viewpoint if there is no solid object between the viewpoint and the surface point, and if reach the surface normal criteria. A point is said to be measurable if is viewable from the camera and the light source (TARBOX; GOTTSCHLICH, 1995).

*Aspect graphs* are a popular class of the viewpoint planning, where the nodes represent every aspect of that object and arcs connect all adjacent aspects on an object (TARBOX; GOTTSCHLICH, 1995; SCOTT; ROTH; RIVEST, 2003). An aspect is loosely defined as the set of viewpoints of the object such that set of views must be able to see the same aspect (vertices, edges, and faces) (TARBOX; GOTTSCHLICH, 1995).

At the heart of the computational geometry methods is the *(Art Gallery Problem (AGP))*, which consists on determine the upper bound on the number of point *guards* ($g$) to cover a polygon of n-vertices. This problem was first proposed by Victor Klee to Chvátal in 1973 (O'ROURKE, 1987).

There are different types of the AGP based on the guard type, such as edge guards, vertex guards, point guards and mobile guards.

Chvatal (1975) proposed that [n/3] guards were occasionally necessary and always sufficient to coverage a n-vertices polygon. To achieve this upper bound value $[g(n) >= 3/n]$[2] is used the idea of triangulation, because a triangle always can be seen by one guard. Although this approach is not able to place the guards (SAFAK, 2009). Although, usually, AGP solutions assume that a guard has an unlimited field of view that is obstructed by obstacles. Even more AGP is NP-hard[3] (LEE; LIN, 1986).

In the 80's many researchers works in the Chvátal AGP and some variations. Some of those researchers spent effort to find a better and linear time solution for the Art Gallery Problem, as Tarjan e Van Wyk (1986) and Guibas *et al.* (1987).

In 1986, O'rourke (1987) compile the most relevant works about the art gallery problem, presenting theorems and algorithms. This book starts with the Chvatal (1975) approach, then presents the problem to orthogonal polygons, extending the problem to 'mobile' guards, polygons with holes and other variations of the art gallery problem. Those approaches modeled the environment as a polygon in a plane although many real problems cannot be modeled in a plane requiring a spatial representation.

The spatial environment representation could be done in many ways, the most commonly is using vertices, that can be stored as a point cloud or joined together using edges to connect them creating a mesh, another way to represent the environment is volumetric representation, storing a list of voxels[4] (OSOSINSKI; LABROSSE, 2014; HORNUNG *et al.*, 2013).

## 2.6 COVERAGE PATH PLANNING

Coverage Path Planning (CPP) is the task of determining a collision-free path that passes over all points of an area or volume of interest avoiding the obstacles while try to maximize or minimize some parameters, such as maximize the coverage and minimize the route length (CHOSET, 2001; GALCERAN, 2014). Since CPP is NP-

[2] $g(n)$ is the number of guards in function of the number of vertices $n$.

[3] NP-hard (*non-deterministic polynomial-time hardness*) is a class of problems that are at least as hard as the hardest problems in NP and there is no known polynomial-time algorithm to deal with this problem (CORMEN *et al.*, 2009).

[4] A *voxel* is a cubic volume in a grid of cubic volumes of equal size of a discretized area.

hard, the computational time required to solve the problem grows drastically when the dimension of the problem increases (GALCERAN, 2014).

CPP is related to many robotics applications such as: floor cleaning (YASUTOMI; YAMADA; TSUKAMOTO, 1988), lawn mowing (HUANG; CAO; HALL, 1986), mine hunting (LAND; CHOSET, 1998), harvesting (OLLIS; STENTZ, 1996), painting (ATKAR; CHOSET *et al.*, 2001), window cleaners (FARSI *et al.*, 1994) and inspection of complex structures (ENGLOT; HOVER, 2017; FLORIANI *et al.*, 2017), just to cite a few.

Cao, Huang e Hall (1988) proposed a list of requirements to CPP methods, as follows:

- 1. Robot must move through all the points in the target area covering it completely;
- 2. Robot must fill the region without overlapping paths;
- 3. Continuous and sequential operation without any repetition of paths is required;
- 4. Robot must avoid all obstacles;
- 5. Simple motion trajectories (e.g., straight lines or circles) should be used (for simplicity in control);
- 6. An “optimal” path is desired under available conditions.

Although, it is not always possible to suit all criteria (GALCERAN, 2014).

CPP algorithm involves the definition of the exploration method (model or non-model-based), generating viewpoints and planning an optimized path, and quantifying the coverage completeness (ALMADHOUN *et al.*, 2016).

The CPP problem is related to the *covering salesman problem* (CSP), a variant of the *Travel Salesman Problem (TSP)* where instead of visiting each city, an agent must to satisfy all the client's demand by visiting or covering them. In other words, each customer have covering radius, then, if a customer is visited, all the other customers inside the radius are said visited. The TSP main idea is try to find the shortest cycle in a network such that all the nodes are visited and the minimum total distance is traveled (GOLDEN *et al.*, 1980).

We divide coverage path planning algorithms into two categories, continuous and discrete. Continuous CPP perform a continuous sensing along the route to be followed. Differently, the discrete CPP attempts to find the minimal number of viewpoints covering the work-space or structure (AGP), followed by finding the shortest route passing through all viewpoint set (TSP) (CHOSET, 2001; ENGLOT, 2012; ALMADHOUN *et al.*, 2016).

### 2.6.1 Continuous CPP

Continuous sensing the environment or deposing by the end effector while follow a trajectory is a characteristic of most of the classical coverage path planning (ENGLOT, 2012; ALMADHOUN *et al.*, 2016). Those methods resemble more the classical path planning than the model-based view planning (ENGLOT, 2012).

Continuous CPP have some similarity to the classical *watchman route problem*. The watchman route problem consists in, given a polygon which the inner area must be observed by an infinite-range *guard*, find the shortest continuous cyclical route that does not intersect the exterior bound of the polygon or any hole, and all points in the polygon are visible along the route (CHIN; NTAFOS, 1988). The problem can be solved to optimally in polynomial time if the starting location is specified and to simple polygons without holes (CHIN; NTAFOS, 1988; ENGLOT, 2012). If there are holes or if the problem is posed on 3D the problem becomes NP-hard (CHIN; NTAFOS, 1988).

Concepts such as cell decomposition, the generalized Voronoi diagram (GVD), and grid-based planning have been widely used in continuous CPP problems (ENGLOT, 2012; ALMADHOUN *et al.*, 2016). Although Englot (2012) and Almadhoun *et al.* (2016) differentiate the Voronoi and grid-based from cell decomposition, Choset (2001) classifies CPP in four categories: (a) heuristic and randomized, approximate (ZELINSKY *et al.*, 1993), semi-approximate (HERT; TIWARI; LUMELSKY, 1996) and exact (CHOSET; PIGNON, 1998) cellular decomposition.

Heuristic approaches are some of the early works in CPP, which consists in some simple rules of thumb that may work well (CHOSET, 2001). In the randomized approach is not necessary a foreknowledge of the environment. An example of heuristic behaviors for multi-robot coverage is repulsion from the others robots. For a single robot an example is avoid obstacle. But those methods do not guarantee the coverage successful (CHOSET, 2001).

Approximate cellular decomposition is based on a fine-grid-based representation of the environment.In this case, all cells has the same size and shape. The union of all cells only approximate the original region. Besides that, normally the cell size is similar of the mapping sensor area and if the vehicle entered into a cell would be assumed that the cell is covered (CHOSET, 2001). As other authors, we will categorized the approximate cell decomposition as grid-based methods.

In semi-approximate cell decomposition approaches the target region is divided into cells with fixed width, where top and bottom can have any shape (CHOSET, 2001).

In exact cellular decomposition the free space is divided in a set of non-intersecting regions whose union fills the target region. To achieve this decomposition many algorithms as the ones presented in Section 2.3 could be used. During the cell decomposition process a adjacent graph should be generated. Two cells are adjacent if they share a boundary. Moreover, usually the shared boundary of two cells have

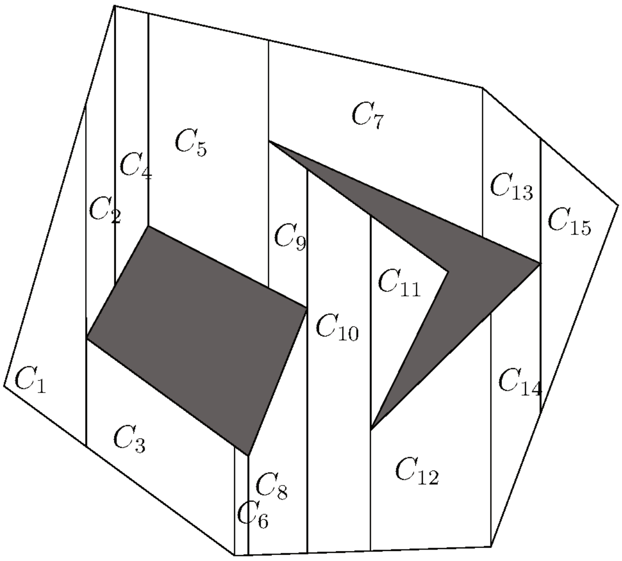


Figura 10 – Trapezoidal decomposition of the $C_{free}$.

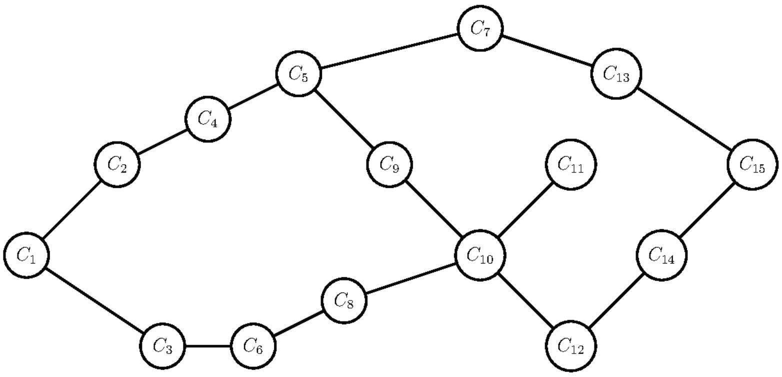


Figura 11 – Adjacent graph of the trapezoidal decomposition of the $C_{free}$.

a physical meaning as changing in the closest obstacles or a change in line sight to surrounding obstacles. In this work, when we mention cell decomposition we are dealing with the exact cell decomposition from Choset (CHOSET, 2001) classification.

Since each cell has a simple structure, each cell could be covered using back-and-forth motions and the coverage problem is reduced to connect and calculate the minimal cost to sequentially visit all the cells.

One popular cell decomposition technique is the trapezoidal decomposition (LATOMBE, 2012). This approach depends on the polygonal representation of the configuration space and comprises two-dimensional cells shaped like trapezoids, though some cells could be shaped like triangles that are degenerate trapezoids (CHOSET, H. M. *et al.*, 2005). Fig. 10 shows a trapezoidal decomposition and fig. 11 the adjacent graph.

Looking to Fig. 10 we intuitively observe that some cells could be merged to form bigger cells. *Boustrophedon decomposition* (CHOSET; PIGNON, 1997; CHOSET, H. *et al.*, 2000) looks to reduce the number of cells without loosing the capability to complete coverage by simple motion, as shown in Fig. 12. Similar in trapezoidal decomposition

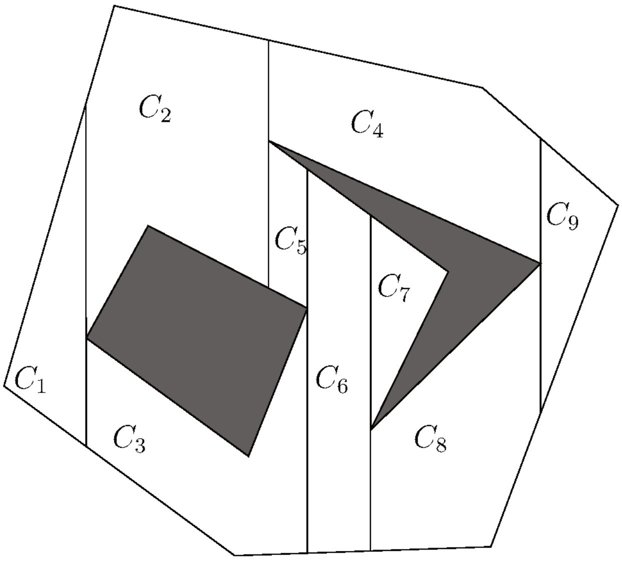


Figura 12 – Boustrophedon decomposition of the $C_{free}$.

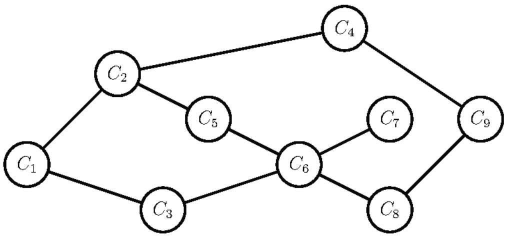


Figura 13 – Adjacent graph of the boustrophedon decomposition of the $C_{free}$.

a line segment, called a slice, is swept through the environment but it only consider a critical vertex and extend a vertical segment if is possible to extend above and below the vertex.

Later Acar *et al.* (2002), presented a novel cellular decomposition which is a generalization of the *boustrophedon decomposition* called *Morse decomposition* based on the critical points of Morse functions. This method is able to handle even with non-polygonal obstacles (ACAR *et al.*, 2002; CHOSET, H. M. *et al.*, 2005; ENGLOT, 2012; GALCERAN, 2014).

Several other continuous coverage algorithms have been developed. Huang (2000, 2001) proposed the optimal line-sweep-based decomposition for coverage algorithm, which explore the reduction on the time on path length based on different sweep orientation in different cells. A spanning tree grid-based method was proposed by Gabriely e Rimon (2001), where initially the $C_{free}$ is discretize in equal square cells and a graph is generated representing the connectivity of the grid. A minimum spanning tree is computed using PRIM (PRIM, 1957) and the coverage path consists in circumnavigates along the spanning tree.

More recently some continuous CPP in 3D workspace approach have been proposed (JIN; TANG, 2011; ENGLOT; HOVER, 2017; PAPADOPOULOS; KURNIAWATI;

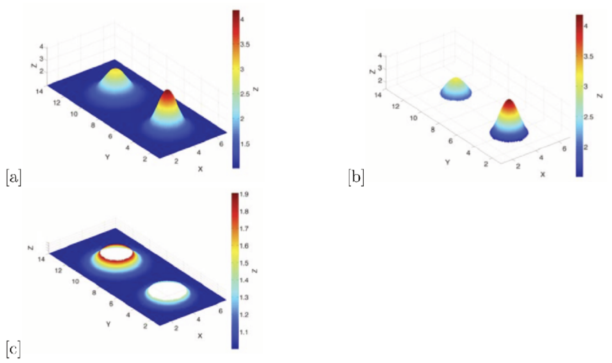


Figura 14 – Terrain classification into high-slope and planar regions. The bathymetric surface in [a], the high-slope in [b] and the planar region in [c].

PATRIKALAKIS, 2013; GALCERAN, 2014; LIM; LA; SHENG, 2014; OLIVIERI *et al.*, 2014). Atkar, Choset *et al.* (2001) presented an online CPP addressed to covering closed orientable surface for spray-painting tasks extending the ideas of the morse decomposition. In their application the end effector did not cover the target surface but an off set surface. Their algorithm was tested in simulation, although didn't consider the robots kinematic. In later work, Atkar, Greenfield *et al.* (2005a,b) proposed an offline approach to spray-painting tasks, focus on uniform coverage problem resulting in a uniform paint deposition. To achieve uniform coverage, they decompose the *a priori* model (CAD model) into simple cells of similar curvature. The method was validate in real experiments.

Jin e Tang (2011) proposed an algorithm to coverage 2.5D arable fields taking into account that most of the farms have rolling terrains. They decompose the arable field in subregions based on its terrain features and look for a recommended *seed curve* to each region based on a cost function, then parallel coverage paths are generate until the whole region is completely covered. Their approach was validated on real-world elevation maps.

Galceran e Carreras (2013) proposed a CPP method to inspect 3D natural structures on the seafloor based on a 2.5D bathymetric map ($\mathcal{B}(x, y)$). This task is important for missions such microbathymetry mapping or image photo-mosaicing. Those missions require a different approach than the usual lawnmover in a constant depth or distance from the seafloor because suddenly depth changes causing a very costly motion and after suddenly changes takes time to adjust the distance between the vehicle and the target surface resulting in deteriorating data quality.

This approach is a tree-step process. Initially, high-slope regions are detected, as shown in Fig.14. The slope map $S(x, y)$ is calculated by the Eq. 1, where $\mathcal{B}(x, y)$ is the *a priori* map function which returns the elevation at point $(x, y)$ on the map and $||\nabla\mathcal{B}||$ is the norm of the gradient of $\mathcal{B}$.

$$S(x, y) = \|\nabla \mathcal{B}\| = \|\frac{\partial \mathcal{B}}{\partial x}\hat{\imath} + \frac{\partial \mathcal{B}}{\partial y}\hat{\jmath}\| \tag{1}$$

A user-defined slope threshold, $\delta_s$ define the high-slope regions

$$T(x, y) = \begin{cases} 1, \text{if } S(x, y) \geq \delta_s \\ 0, \text{if } S(x, y) < \delta_s \end{cases} \tag{2}$$

Then constant-depth horizontal-pattern coverage path are generated using a slicing algorithm. Finally a Morse boustrophedon cellular decomposition in the remained area is performed, since the high-slope regions already been are treated as obstacles. This stratagem allow to mapping high relief area. An extension of this method with realtime replanning, presented in Galceran, Campos *et al.* (2014)

Most of the previous methods looking for 100% coverage, although 100% coverage sometimes is not possible or desirable. Could be not desirable if the difference cost between a well-done coverage and a complete coverage is too large when compared with 100% coverage, or it can be not possible if the structure to be inspected has occluded or hidden parts due to limitations of the vehicles size or movement capability.

Taking into account that complete coverage is a objective and not a constraint, Ellefsen, Lepikson e Albiez (2016) presented an algorithm which optimize the set of viewpoints and the order to visit them together.

At the start a set of candidate viewpoints is generated ordered as a grid around the region of interesting. And some possible initial *circling path* are generated, as Galceran, Campos *et al.* (2014), which are *seeds* solutions. The *seeds* allow them to use expertness of a fine solution to guide optimization. Later the *non-dominated sorting genetic algorithm (NSGA-II)* utilizing the *distributed evolutionary algorithm in Python (DEAP)* was used to optimize the path looking for: (a) maximizing coverage and (b) minimizing energy usage. After some iterations the optimization terminates, and the final path is returned.

A model-based coverage path planning algorithm to surveillance of sensitive areas for a team of UAVs, called PARCov (Planner for Autonomous Risk-sensitive Coverage) was proposed by Wallar, Plaku e Sofge (2014). That approach aim to maximize the coverage area by each UAV with a high sensor data quality and minimizing detection risk. Initially PARCov uses a dynamic grid in the $XY$ bounding box to evaluate the parts that already have been covered and the time that they were last visited.

PARCov achieve a high coverage (i.e., greater than 90%) after around 20 iteration[5] is able to reduce the risk when the number of iterations grows.

In-detail surveys of continuous CPP are presented by Scott, Roth e Rivest (2003) e Galceran e Carreras (2013).

[5] One iteration corresponds to one move (a novel position and orientation) for each quadcopter.

### 2.6.2 Discrete CPP

Discrete CPP consists in two main main problems, find a discrete set of viewpoints to cover the work-space or structure (AGP), followed by finding the shortest route passing through all viewpoint set (TSP) (ENGLOT, 2012; ALMADHOUN *et al.*, 2016). This class of CPP also bears some similarity to the Generalized Watchman Route Problem (WANG; KRISHNAMURTI; GUPTA, 2010), where the views are collected at discrete locations along the route.

On this tasks usually the robot follows a sequence of waypoints and each time a viewpoint is achieved the system must stabilize and perceive the environment (collect the data) in order to avoid missing information (PAPADOPOULOS; KURNIAWATI; PATRIKALAKIS, 2013; ALMADHOUN *et al.*, 2016).

Discrete approaches have good performance even handling with obstacles and occluding elements. Although the dependencies between the viewpoint planning and path planning goal's tasks limit the optimization of the two step methods when performed separate (ENGLOT; HOVER, 2017; PAPADOPOULOS; KURNIAWATI; PATRIKALAKIS, 2013)

One of the earliest approaches to deal with sensor placement in a discrete way is called *view sphere*, which constraint the visibility space on the surface of a sphere surrounding the object, then, sample viewpoints should be uniformly distributed. While does not exist a known uniform tessellation of a sphere (SCOTT, 2009). A close approximation is centering a regular polyhedron inside the sphere in a such way that all the polyhedron vertices touch the sphere surface. A widely used regular polyhedron is the icosahedron and to allow a better approximation between the sphere and the discretization is commonly subdivided the triangular faces of the icosahedron projecting the new vertices in the sphere surface (TARBOX; GOTTSCHLICH, 1995; TARABANIS; ALLEN; TSAI, 1995; TRUCCO *et al.*, 1997). This stratagem reduces the problem to find the direction of view in the sphere.

Although, as all the viewpoints are in the sphere surface and the object is enclosed in the sphere those approaches did not address collision-avoidance.

The following step is solve the TSP, which is a NP-hard well studied problem where many fast algorithms were proposed and approximately solve are known, as (DANTZIG; FULKERSON; JOHNSON, 1954; LIN; KERNIGHAN, 1973; ROSENKRANTZ; STEARNS; LEWIS, 1974; CHRISTOFIDES, 1976; GOLDEN *et al.*, 1980; LAWLER *et al.*, 1985; REINELT, 1994; DORIGO; GAMBARDELLA, 1997; APPLEGATE *et al.*, 2006)

More recently, Englot e Hover (2017) introduces a novel model-based, sampling-based, discrete or continuous complete CPP to inspect complex 3D structures. The problem to be solved is an autonomous ship hull inspection which is a challenge structure because the object to be inspected is large and contiguous, and the robot's sensor footprint is small relative to the size of the structure, and the close proximity of the parts to be inspected (shafts, propellers, and rudders) and to the hull.

The approach divided the problem into two phases, as many others discrete or continuous CPP algorithms, although Englot e Hover (2017) divided the method based on the two sampling-based subroutines used. The first phase is comprised of solving a coverage sampling, then a roadmap is constructed which is, samples feasible robot configurations and list their sensor observations, generating a discrete state space that together guarantees 100% coverage of the structure boundary, they phase is called *coverage sampling problem (CSP)*. This phase differs from the classical *AGP* widely used, as requires just a set of a feasible covering configurations and not the minimum set.

The second phase requires solving a variant of the TSP called *multi-goal planning problem (MPP)*, to perform this task they initially use a lazy point-to-point planner to build the adjacency matrix, then a Christofides (CHRISTOFIDES, 1976) algorithm to compute a *minimum spanning tree (MST)*. A RRT post-optimization smoothing implementation is applied to find a collision-free paths and avoid the violate of the triangle inequality. Then a Lin-Kernighan algorithm (LIN; KERNIGHAN, 1973) iteratively improves a TSP solution.

Another sampling-based path planning algorithm to inspection tasks called *random inspection tree algorithm (RITA)* very similar to the one presented by Englot e Hover (2017) was proposed by Papadopoulos, Kurniawati e Patrikalakis (2013). One of the motivations to *RITA* development is the difficult or impossibility to reach some samples because the robot's kinematic and dynamic constraints weren't take into account in the sampling and selection phases. To avoid this limitation of the previous approaches, they simultaneously compute the samples and the trajectory to achieve then.

Some improvements on *RITA* like realistic sensing constraint of the forward looking camera and performance of the the roadmap expansion were presented in Kafka, Faigl e Váňa (2016). Moreover, this algorithm evaluate the distance and the angle to guarantee gathered data with sufficient details.

A two step optimization, model-based, sampling-based view planning for 3D inspection or reconstruction tasks with unmanned aerial vehicle was proposed by Jing *et al.* (2016). Their approach consists in the following four steps: (a) preprocesing the input model, (b) viewpoint generation, (c) combinatorial optimization and (d) iteration of the two-step process. Initially, in the pre-processing step a Bubble mesh method (SHIMADA; GOSSARD, 1995) is used to generate a novel triangularization with well-shaped and uniformity triangles. Then a binary voxel dilatation (HARALICK; STERNBERG; ZHUANG, 1987) is performed two times, using the camera's maximum and minimum FOD (field of depth) to generate the sampling space. Random sampling are generated in the sampling space to define the candidate viewpoint position. To generate the view-

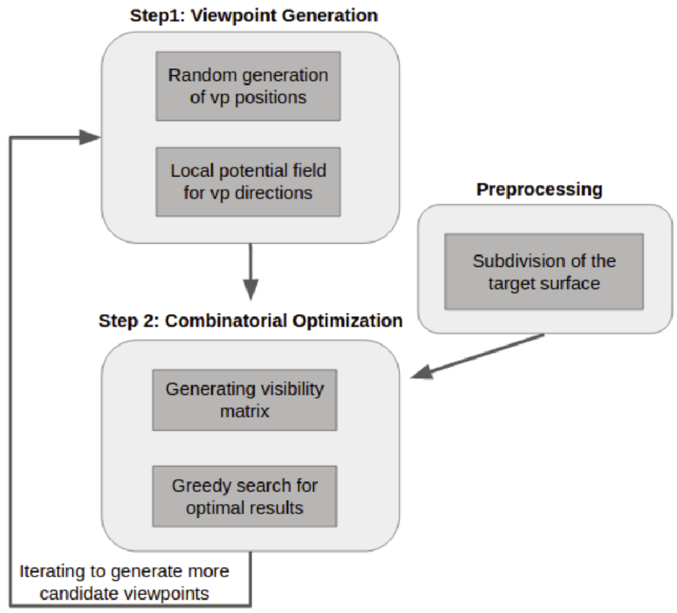

Step1: Viewpoint Generation
Random generation of vp positions
Local potential field for vp directions
Preprocessing
Subdivision of the target surface
Step 2: Combinatorial Optimization
Generating visibility matrix
Greedy search for optimal results
Iterating to generate more candidate viewpoints

Figura 15 – Jing(JING *et al.*, 2016) sampling-based view planning scheme.

point direction a novel probabilistic potential-field method is used and the direction is calculated based on the nearby attraction forces.

The next step is the combinatorial optimization which includes the visibility matrix generation and solving the *set covering problem (SCP)*. Jing *et al.* (2016) uses a modified greedy search algorithm to solve the SCP, other approximate methods such as genetic code and dynamic programming could be used to deal with the SCP although the greedy algorithm have a low computational cost and produce acceptable results.

The last part is the iterative step which starts with the detection of the existence of unseen patches. If there are unseen patches additional viewpoints are generated, those viewpoints are added to the candidate viewpoints list and the combinatorial optimization is performed again, this process is repeated a number of times to improve the solution, as shown in Fig. 15.

A model-based coverage method to surveillance missions for a micro aerial vehicle (MAV) was proposed by Janoušek e Faigl (2013) using triangular meshes to model the regions of interesting. The proposed algorithm is an extension of the *watchman route problem (WRP)* derived on self-organizing neural network (FAIGL, 2010) based on a cover set. This approach different from many of the view planning algorithms which first determine the viewpoint positions and later perform a multi-goal path planning, execute these two steps simultaneously.

The start input is a triangular mesh with *Objects of Interest (OoIs)* and the visibility dependencies. A tetrahedral mesh of the working environment (e.g. using Si e TetGen (2006) is obtained), then a roadmap is generated (e.g. using PRM (KAYRAKI *et al.*, 1996). The next step is the construction of the covering spaces (Sensing locations).

The covering space is a set of tetrahedrons in the free space of the robot working space from which the whole *OoIs* can be seen. Then the inspection path is determined based on the *self-organizing map (SOM)* approach for the *WRP* variant for a roadmap graph representation.

Janoušek e Faigl (2013) approaches is very similar to Englot e Hover (2017) method, the principal distinction is the possible sensing location, Englot e Hover (2017) consider determine the sensing locations prior to finding a path to connect the set of needed samples. On the other hand, Janoušek e Faigl (2013) define a bounding box where the viewpoints should be locate and during the path planning define the viewpoint position, similar to the approach proposed by Bircher *et al.* (2015)

Bircher *et al.* (2015) presented an offline structural inspection path planning algorithm for complex 3D structures using aerial robots. The approach is designed to be able to path planning for rotorcraft and fixed-wing unmanned aerial systems.

The method requires a triangular mesh representation of the region of interesting (structure). The algorithm employs an alternating two-step optimization paradigm to find viewpoints that provide full coverage with a low cost path.

First for each triangle in the mesh a sampling space where the viewpoint can be set, is generated using the camera range, the incidence angle and the normal of the edges of this triangle. A sample viewpoint in each sampling space is selected, then a cost matrix is calculated and the TSP solved in order to determine the initial tour.

Next the viewpoints are resample inside the sampling space, the cost matrix and the TSP solved again in order to find an optimum path. The sampling space guarantee that any point in this volume can view the whole triangle Moreover, as for each triangle in the mesh they select a viewpoint the complete coverage is guaranteed.

A novel uniform coverage focus not only in complete coverage in the shortest path but also uniform observation of each detail was proposed by Alexis *et al.* (2015).

Initially the input uniform triangular mesh is remeshed producing a coarse mesh and a viewpoint per mesh face to complete coverage is computed based on pure geometric calculations. Then a combination of a *Boundary Value Solver (BVS)* and *RRT** are performed to connect the set of viewpoints, finally they use an implementation of the *Lin-Kernighan-Helsgaun Heuristic (LKH)* heuristic, as Bircher *et al.* (2015) to solve the *TSP*. Those steps produce the first solution, subsequently more detailed meshes are used and new viewpoints are added generating novel inspection routes.

On experimental studies the iterative step of load higher fidelity mesh to improve the uniformly coverage generates a few increase in the number of used viewpoints as well as in the time consuming.

A method to plan inspection tasks in underwater installations, like a collection of structures on the oilfield is proposed by Cashmore *et al.* (2013).

They consider that a precise model of the structure is not available, then they represent the installations using a simple shape, as a cube, enclosed by free space.

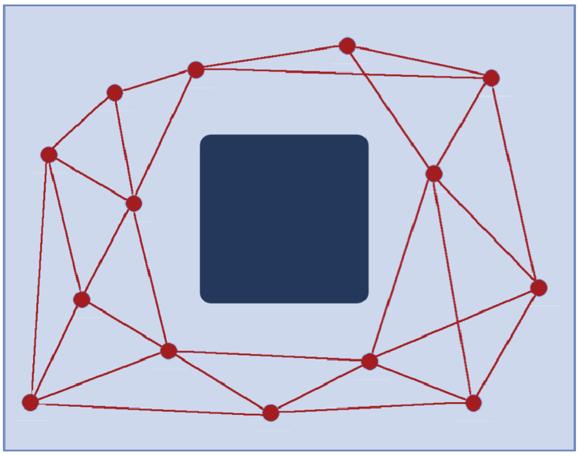


Figura 16 – The 2D slice of the coarse structure representation (dark blue) and a preliminary roadmap (red), extracted from Cashmore *et al.* (2013).

An initial map surrounding the region of interesting is generated using a *Probabilistic Roadmap Method (PRM) (KAYRAKI* et al.*, 1996)* utilizing the Open Motion Planning Library (OMPL) (SUCAN; MOLL; KAVRAKI, 2012), as shown in Fig. 16

To guarantee the visibility of the *inspection points*, that could be achieved by making the *PRM* more dense, the authors define and use *strategic points*. *Strategic points* are acquire by projecting rays from the *inspection points* and sampling along that ray. The distance between the samples and the structure must be larger than the safe distance and smaller than the maximum sensor range. These samples are connected with the roadmap.

Later the inspection task is modelled in *Planning Domain Definition Language (PDDL)* and *FPOP* (COLES *et al.*, 2010) algorithm is used to solve the planning problem. Although FPOP didn't take into account the orientation, the re-orientation between adjacent inspection acts and the necessity of multiple visits to the same viewpoint. Cause of this, a post-processing is performed.

This strategy use a sample-based approach to deal with an inspection problem, producing fast results although just for very simple environments.

A similar problem of the CPP is the coverage search, although for the coverage search methods, the main objective is search objects or survivors for example. Coverage search is also related with the NBV algorithm although they usually didn't consider the costs to travel to the next viewpoint which is fundamental in the search problem.

Dornhege, Kleiner e Kolling (2013) presents a model-based coverage search algorithm using a mobile robot platform. The method representation of the environment is required in the form of a *OctoMap* (HORNUNG *et al.*, 2013).

Initially they sample generated *poses* $(x, y, z)$ where the sensor could be placed, i.e. reachable sensor states, and obtain a set of poses from which large parts of search space are visible. Then they sort the *poses* vector using the *utility* $(U)$, sample one orientation to each *pose* and re-calculate the *utility*, if the utility sensor is larger than a set value the pose is add to the sensor state set (x̃), after N = | x̃ | for a given N, they stop to add samples. The stop criteria is used because in many applications is unrealistic expected the complete coverage, then the user must choose the number of view according to the intended coverage.

To reduce the number of views (sensor state) they partitioned the search space (environment/structure) based on the detection for a sensor state set (x̃).

Finally they test several coverage search approaches, like (a) greedy solutions (next best view, cost-based next best view, greedy cost) to compute the coverage sequence by incrementally selecting x $\in$ x̃ until all the possible visible parts are seen, (b) complete planning and (c) TSP.

Dornhege, Kleiner e Kolling (2013) compared different variants of their algorithm in several scenarios showing that the set cover approach using a specific TSP solver produces a better result.

A method to estimate the visibility of a complex 3D structure and the position of the required viewpoints posed in a 2D plan was proposed by Ososinski e Labrosse (2014). They use an octomap representation of the environment and pose the set of candidate viewpoints in the vertex of a grid in a plane.

The initial task is the visibility estimation which is performed into two steps. First is evaluate the voxel perception, in other words, analyze if a voxel face is full visible, partial visible or no visible, which is the perception class constant. Then estimate how well the face is view (visibility value) based on the distance and the incidence angle between the voxel center and the viewpoint.

The normalized distance ($D(V_p, f)$) is calculated by Eq.( 3), where D is the normalized distance between the viewpoint and the cell face, $R_{min}$ is the minimum sensor range and $R_{max}$ is the maximum sensor range.

$$D(V_p, f) = \frac{d(V_p, f) - R_{min}}{R_{max}} \tag{3}$$

The normalized incidence angle ($A(V_p, f)$) is calculated by Eq. 4, where $\theta$ is the angle of incidence from the viewpoint onto the face $f$, expressed in radians.

$$A(V_p, f) = \frac{\theta(V_p, f) - \frac{\pi}{2}}{\frac{\pi}{2}} \tag{4}$$

The visibility $V(V_p, f)$ of the face from a viewpoint $V_p$ is given by the Eq.5

$$V(V_p, f) = 0.5^{6} + 0.3xD(V_p, f) + 0.2xA(V_p, f) \tag{5}$$

The global visibility $G_v(S, F)$ is calculated as the sum of the maximum visibility $V(V_p, f)$ of a face $f$ from any $V_p$ of a set of all selected viewpoints $S$ normalized by the number of faces $|F|$.

[6] If the face is fully visible should use 0.5, if the face is partially visible ( $< 4$ corners visible) use 0.3

Although the global visibility estimation the selection of viewpoints to be added in the $S$ is performed as Floriani *et al.* (2017).

$$\bigwedge_{V_p \in S} G_v(S, F) = \sum_{f \in F} \frac{max(V(V_p, f))}{|F|} \tag{6}$$

The global visibility estimation is an important parameter, mainly to non complete coverage problems, nevertheless is a very time consuming process and is not used to select the viewpoints.

## 2.7 SUMMARY

In this chapter, we have seen a comprehensive review of relevant prior work in path planning, view planning and coverage path planning. To model-based coverage tasks groups emerge from our review: *continuous CPP* where the environment is continuous sensing while the robot follow a trajectory, and *discrete CPP* where a discrete set of configurations is used to coverage the environment.

Continuous CPP has been addressed using many different approaches. The trapezoidal decomposition have been widely used on polygonal planar space problems. The Boustrophedon decomposition represented an improvement to the trapezoidal decomposition generating shorter coverage paths. Later, the Morse-based cellular decomposition is able to deal with non-polygonal obstacles, since the obstacles boundary were differentiable. Although morse-decomposition cannot handle with rectilinear environments.

Some perspectives emerge in our review of discrete coverage to deal view the viewpoint generation: icosahedron discretization, grid-based, surface normal's, random sampling and sampling in a limited bounding box to each subregion.

Icosahedron discretization is one of the earliest view planning approaches to generate candidate viewpoints, although has many drawbacks, as do not avoid collision as already presented, and for the best of the authors known, there are no utility for large areas. One of the reasons to this, is because the radius to enclose the whole volume should be larger than the sensor range, making it impossible to inspect, or produce high quality data since the sensor must be set very far away from the object to be inspected.

Grid-based approaches have been used mainly in indoor applications because grid maps suffer from exponential growth of memory usage and became impracticable on large areas.

The use of the surface's normal to determine the viewpoint or to generate a region where the viewpoint can be located have been used in some recently methods, nevertheless sometimes is not applicable to complex environments or structures because the normal could point to an obstacle then there is no viewpoint to observe that region. Moreover, if we are using triangular mesh to represent the environment, if the number of triangles in the mesh to inspect is relatively elevated, the time required to generate the path can be prohibitive or, the generated path taking very long time what is not acceptable for an AUV.To solve this problem it is possible to combine several views in a single one, resizing the mesh. However, this operation can change the orientation of some normals, which affects the use of this method. Moreover, it is possible to lose some important environment information (like, normals) when resizing the mesh.

Random sampling algorithms are broadly and successfully used on path planning problems of high dimension or complex geometry for more than 20 years and recently have been used on discrete CPP methods. Thanks to the *probabilistic completeness* the probability to find a path if one exists tends to one as the number of samples tends to infinity. In a practical perspective this means that a near-optimal solutions can be obtained using a finite number of samples.

Besides that, recently Englot (2012) define the upper bound value to guarantee the complete coverage using random samples.

The definition of a bounding box where the viewpoint must be locate to inspect a subregion could be a very interesting approach, specially to coverage problem where goals are sensing locations. Although, as the case of normal's, if there is to much subregions the problem could be intractable.

The second part of the discrete coverage path planning is related to solve the TSP, which have many fast solver's algorithms.

Table 1 shows a summary of the most relevant methods on model-based continuous and discrete CPP. From those methods, just Englot (2012) and Ellefsen, Lepikson e Albiez (2016) are focused on underwater 3D complex structures, while Galceran e Carreras (2013) center on regions with high relief. Englot (2012) approach solves the hull ship inspection problem although the approach was not tested on different scenarios, even more to sample in the local neighborhood of a geometric primitive, a random configuration is constructed in a spherical coordinate system centered at the primitive, which means that, for each primitive a sample space is constructed. On the other hand, Ellefsen, Lepikson e Albiez (2016) use an evolutionary multi-objective optimization to find the best way to continuous inspect the structure. The method proposed by Galceran e Carreras (2013) deal with continuous CPP tasks in high relief regions but is not able to deal with 3D complex structures.

The goal of this thesis is to plan an inspection route to coverage complex structures. As earlier mentioned, many inspection approaches have being proposed using UAVs, as well as methods to coverage path planning for arable fields and, even, to subsea coverage, although can we notice a lack for discrete CPP to inspect several kind of 3D complex structures.

Survey of Model-Based Continuous Coverage Path Planning Summary

| Reference | Approach<br>Viewpoint generation | Coverage path<br>planning | Remarks | Application |
|---|---|---|---|---|
| (ATKAR; GREENFIELD *et al.*, 2005a) | Cellular decomposition based on the curvature | Optimal paint-deposition CPP in each cell | Uniform paint decomposition | Spray-painting of automotive parts |
| (GALCERAN; CARRERAS, 2013) | High-slope regions coverage +<br>Boustrophedon decomposition | Lawnmower path +<br>Exhaustive walks | Lawnmower path to 3D environments | High-relief underwater environment |
| (WALLAR; PLAKU; SOFGE, 2014) | Dynamic grid +<br>Random sampling orientation | Move the UAVs to regions that have not been covered in a long time | Coverage completeness + high senosr data quality | Surveillance using a team of UAVs |
| (ELLEFSEN; LEPIKSON; ALBIEZ, 2016) | Circling path | Non-dominated Sorting Genetic Algorithm (NSGA)-II (DEB *et al.*, 2002) | 100% coverage is not a constraint | Inspection of subsea oilfield installation |

Survey of Model-Based Discrete Coverage Path Planning Summary

| Reference | Approach<br>Viewpoint generation | Coverage path<br>planning | Remarks | Application |
|---|---|---|---|---|
| (TRUCCO *et al.*, 1997) | Icosahedron discretization vertices | Extension of CAOO<br>(LAWLER *et al.*, 1985) | Reduce the viewpoint space to a discretize sphere | Small objects |
| (ENGLOT, 2012) | Random sampling-based +<br>Roadmap | Lazy point-to-point planner +<br>minimum spanning tree (MST)+<br>RRT + Lin-Kernighan algorithm | Sampling-based complete coverage | In-water ship hull |
| (SCHMID *et al.*, 2012) | Normal of the smoothed digital surface model | Abstraction of the environment +<br>Replanning | The algorithm time complexity is linear | Reconstruction using Autonomous Multicopter |
| (PAPADOPOULOS; KURNIAWATI; PATRIKALAKIS, 2013) | Randomized Kino dynamics sampling-based motion planne | RITA algorithm | computes viewpoint and the trajectory simultaneously | 2D structures |
| (CASHMORE *et al.*, 2013) | PRM + *Strategic points* from the *inspection points* | PDDL + FPOP | | Structural Inspection using a Micro Aerial Vehicle (MAV) |
| (JANOUŠEK; FAIGL, 2013) | Tetrahedralization of the working environment | PRM + SOM | Calculate a bounding box where the viewpoints should be locate | Coverage problem where goals are sensing locations |
| (DORNHEGE; KLEINER; KOLLING, 2013) | Random Sampling +<br>*Utility* function + sensor state set | Set coverage approach +<br>Temporal Fast Downward (TFD) (DORNHEGE; EYERICH *et al.*, 2009) | Compared several variants of the algorithm to evaluate the trade-off between schedule computation and execution time. | Coverage search |
| (OSOSINSKI; LABROSSE, 2014) | Grid-based distribution | - | Estimate the visibility | 3D structures |
| (BIRCHER *et al.*, 2015) | Sampling space to each triangle + Viewpoint Sampling | LKH (HELSGAUN, 2000) | Resampling viewpoints to minimize the tour lenght | Structural Inspection |
| (ALEXIS *et al.*, 2015) | Geometric calculations to determine the viewpoint to each triangle | BVS + RRT* + LKH<br>(HELSGAUN, 2000) | Uniform coverage | Structural Inspection using a MAV |
| (JING *et al.*, 2016) | Random sampling-based +<br>Probabilistic potential-field | - | Use a potential-field to determine the view orientation | 3D structures |

Tabela 2 – Survey Summary

# 3 MODEL-BASED DISCRETE COVERAGE PATH PLANNING APPROACH

In this chapter we introduce a new 3D model-based discrete coverage path planning algorithm for 3D complex structure in the underwater environment, as shown in Fig. 17. The underwater environment have some particular challenges, like a poor *a priori* model, navigation drift, very limited perception range of optical sensor *etc*. The key advantage of our approach is provide a simple and fast algorithm to deal with complex structures. In Section 3.1 we introduce the problem, the method is detailed in Section 3.2, Section 3.3 presents results obtained in simulation experiments with real structures and a discussion is presented in Section 3.4.

## 3.1 INTRODUCTION

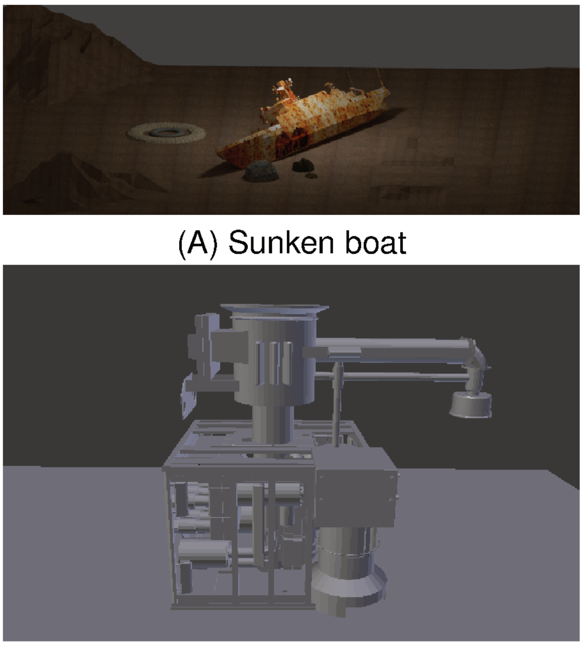
As mentioned earlier, many underwater structures require to be inspected to infer an updated model, or scan for risks and hazards, to plan the maintenance. Even more, to install new structures is necessary mapping the environment. Also to install new structures is required to know the environment.

AUVs have been used to coverage (mapping) those regions, although most of the existing CPP approaches operates in a constant depth using a typical overhead [1] survey, back-and-forth motion, which are insufficient to map 3D high relief environments; if the sensor is pointing perpendicular to the sea bottom it will produce a 2.5D model, if the sensor's ray have a non perpendicular angle to the sea bottom and if overlapped scans are gathered from different directions, a partial 3D map can be generated. Regarding these environments, Figure 18 shows a typical bathymetry performed using the back-and-forth laps and Fig. 19 shows the octree model from the gathered data at 0.3 $m^3$ resolution. In other words, those approaches are not adapted to 3D mapping of submerged industrial facilities, as shown in Figure 17 or complex 3D regions; usually the distance between the back-and-forth laps is determined by the robot's sensor footprint.

More recently Galceran (2014) presented a coverage method for 3D complex structure, using the back-and-forth motion. To deal with the depth variation they slice the environment in regions with similar depth which appear as an interesting solution. Although this algorithm operate mostly in 2.5D scenarios.

Then, to inspect 3D complex environments, such as oil and gas industrial facilities, dams, bridges, offshore wind turbines, sunken boats as well as ship hulls, a novel kind of algorithms is required.

[1] Overhead point of view is produced by a flight at a safe height above the scene pointing the camera straight down (ROBERTS *et al.*, 2017)



(A) Sunken boat

(B) Subsea tree structure

Figura 17 – Original environment representation. Extracted from Floriani *et al.* (2017)

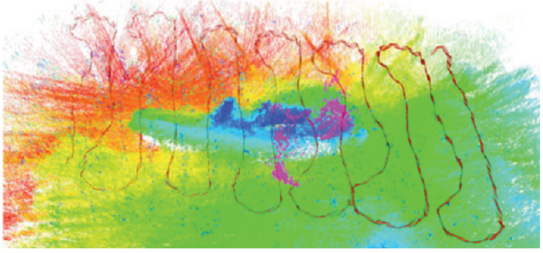


Figura 18 – Raw data from a Imagenex DeltaT multibeam survey over a shipwreck named Boreas (Pálamos, Spain) using Girona 500 AUV. Extracted from Palomeras *et al.* (2018)

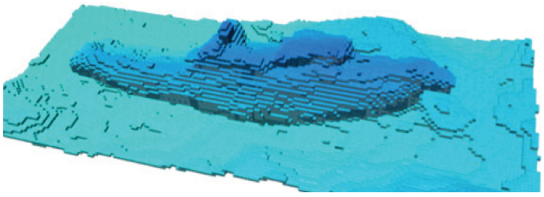


Figura 19 – Octree model from Boreas (Pálamos, Spain). Extracted from Palomeras *et al.* (2018)

## 3.2 3D MODEL-BASED DISCRETE COVERAGE PATH PLANNING FOR AUVS

The method we propose generates an efficient path to coverage (inspect) a 3D complex structure of interest by an AUV. Despite the method can be adapted to other acquisition sensor and vehicles, like the Sparus II (CARRERAS *et al.*, 2013) shown in Fig. 20, at this point the method is limited to vehicles with 4 DoFs at least. The Girona 500 AUV (RIBAS *et al.*, 2012)(4 DoFs) equipped with a multibeam sonar (240

Figura 20 – Sparus II. Image Credit: Computer Vision and Robotics Research Group (VICOROB).

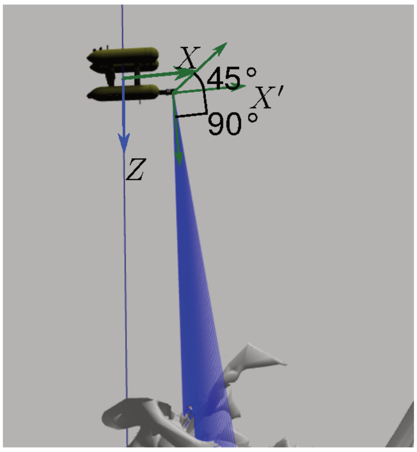


Figura 21 – Girona 500 AUV on Boreas (Pálamos, Spain).

beams distributed in a 120°) mounted over a pan and tilt unit was been used in the tests, Fig. 21 shows the Girona 500 AUV, the multibeam sonar, the scenario, their respectively coordinate system and the tilt limitation from -90°to 45°.

On this proposition the initial inputs are composed by an *a priori* model of the region of interest, a bounding box delimiting the region of interest, the safety distance (the minimum distance between the vehicle and the closest occupied cell), the sensor range and field of view and the number of samples (viewpoints) to be generated. Many viewpoints are generated and we select the best viewpoints based on the number of occupied cells that are visible from that viewpoint. Later the shortest path is calculated using the nearest neighbor TSP solver.

If no *a priori* Digital Surface Model (DSM) is available, a initial survey with back-and-forth rectangular sweep pattern at a constant depth could be performed to build a 2.5D model allowing a detailed inspection.

Next, our approach is described in-detail. In Subsection 3.2.1 we present the *a priori* model representation. Subsection 3.2.2 introduces the viewpoint generation (lines from 1 to 6 of the algorithm 1) and selection (lines 7 to 17 of the algorithm 1). The sequence of viewpoints (lines 18 to 21 of the algorithm 1) is shown in Subsection 3.2.3 and we describe the final (line 22 of the algorithm 1) path computation in Subsection 3.2.4

**Algorithm 1** - Viewpoint

```
Input Octomap, Working Space (S), Number of Random Viewpoints (X) and Stop
    Criteria (C);
    Initialization :
 1: for $int\ i = 0$ to $X - 1$ do
 2:    Viewpoint Candidate = random();
 3:    if Viewpoint Candidate is place in a free node then
 4:       Vector of Viewpoints $\leftarrow$ Viewpoint Candidate;
 5:    end if
 6: end for
 7: Select Viewpoints(Vector of Viewpoints)
 8: {
 9: sort.NumVisibleCells.Vector of Viewpoints()
10: int $V$ = Vector of Viewpoints.size()
11: Viewpoints Sortted $\leftarrow$ Vector of Viewpoints;
12: }
13: for $int\ i = 0$ to (V -1) do
14:    if NumVisibleCells.Vector of Viewpoints[(V-1)-i] > C then
15:       Best Viewpoints $\leftarrow$ Vector of Viewpoints[(V-1)-i];
16:    end if
17: end for
18: for $N = 0$ to Best Viewpoints.size() do
19:    Connect all Viewpoints;
20: end for
21: Solve the TSP;
22: Find a path
23: return  Final Path
```

### 3.2.1 Environment Representation

In Chapter 2 we show different environment representation. Our approach uses a volumetric representation (octree) of the environment. An octree is a set of nodes (occupied spaces), each one representing a cubic volume (*voxel* or cell) that is recursively subdivided until reach a defined size or a number of pre-defined subdivisions, as shown in Fig. 22.

We use the Octomap (HORNUNG *et al.*, 2013) implementation, an open-source framework based on octrees and probabilistic occupancy estimation for mapping and environment representation. Differently from the original octree representation, octomap explicitly represents the free and unknown cells [2]. Moreover, octomap keeps a compact size, includes a fast ray casting computation algorithm. Besides, octomap is

[2] Occupied cells are obtained by the end point of a distance sensor such as a multibeam, free cells are the cells between the sensor and the end point in the sensor direction and a cell is unknown if we do not have any information about it.)

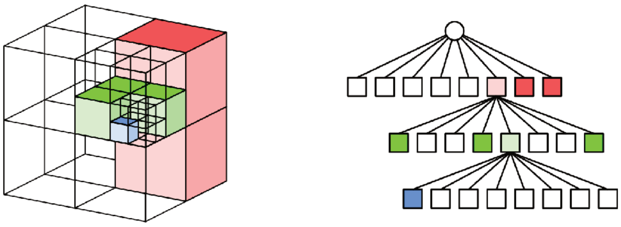
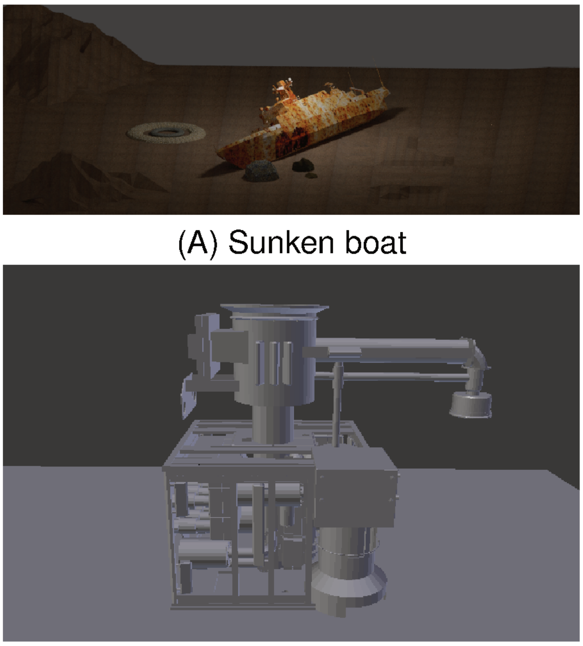


Figura 22 – An octree model, where the blue cell is occupied. In the left, the volume representation and in the right the corresponding tree. Figure extracted from Hornung *et al.* (2013).



(A) Sunken boat

(B) Subsea tree structure

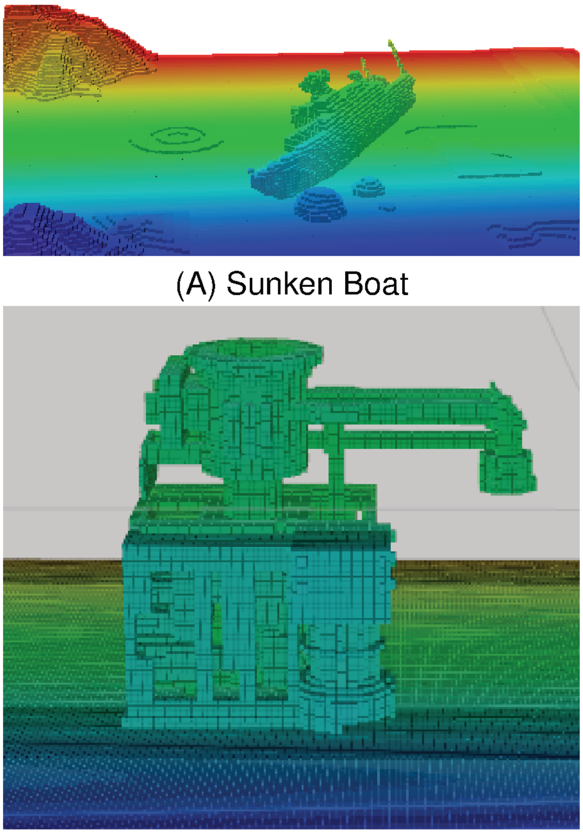
Figura 23 – Original environment representation. Extracted from Floriani *et al.* (2017)

a well documented and supported library and have been used in several NBV approaches (VASQUEZ-GOMEZ; SUCAR; MURRIETA-CID, 2013; QUIN *et al.*, 2013), offline coverage and view planning (OSOSINSKI; LABROSSE, 2014) and many other robotic applications.

Figure 23 shows two test cases, (A) a sunken boat and (b) a subsea tree structure. Both scenes are enclosed in a cubic volume of 20x20x20 $m$ and each cell of the octomap representation has a side length of approximately 0.078 $m$, the octomap representation is presented in Figure 24.

### 3.2.2 Viewpoint Generation and Selection

Viewpoints start to be randomly generated inside the bounding box (Algorithm 2, lines 3 to 5). For each viewpoint generated some criteria are evaluated: (a) the viewpoint must be inside a free cell (Algorithm 2, lines 6 and 7), (b) at least one occupied cell must be visible from the viewpoint under sensors constraints (Algorithm 2, lines 8 to



(A) Sunken Boat

(B) subsea tree structure

Figura 24 – Octomap representation of the original environment.

11), and (c) the viewpoint is in a safest distance from the obstacles (Algorithm 2, line 10). If this three criteria are accomplished the viewpoint is added in a list of candidate viewpoint (Algorithm 2, line 12).

For each candidate viewpoint are stored its position and an octomap that contains all the cells from the structure/environment to be inspected that are visible from this position, at this phase each octomap have the same resolution of the *a priori* model, since is the only data we have from the environment, i.e. if our *a priori* model have a resolution of $0.3m^3$ the visible cells have a $0.3m^3$ size. Nevertheless after the real inspection the resolution is related to the sensor's resolution e.g. for a sensor with a range from 2 to 5 meters with rays separated by 0.5°the maximum distance between a data and it's neighbour should be 0.0437 $m$, which means, that the new octomap have a resolution of 0.04 $m$, in the worst case.

The number of candidate viewpoints to be added is an input defined by the user, which depends of the experience of the user, as well as, the complexity of the environment and the equipment possibility. However, as the algorithm is fast is recommended that the user perform some tests with different number of random points to select the ones that fits better to his application as we will see on the Section 3.3. Nevertheless a theoretical upper boundary is the number of cells on the *a priori* model (ENGLOT, 2012).

Based on the probabilistic theory, the number of trials to obtain a success is given by the Eq.7 :

$$E[X] = E[\sum_{k=j}^{\infty} kq^{k-j}p], \tag{7}$$

where, $p$ is the probability of reaching a safe point, with a minimum distance from the obstacle, $q$ is the probability of failure, $k$ is the number of success of the event $X$, $j$ is the number of failures of the event $X$ and $n$ is the number of trials.

From Eq.7 we obtain:

$$E[X] = \frac{1}{p} \tag{8}$$

Then, if we want to obtain $m$ safe viewpoints, we expect be fast:

$$mE[x] = \frac{m}{p} \tag{9}$$

where $p$ is the probability of reaching a safe point, in other words, if $q$ is the percentage of the total volume which is occupied or in a not safe distance from the obstacle, $p = (1 - q)$.

After reaching the number of candidate viewpoints required by the user, the next step consists in to select the viewpoints to be used in the inspection task. The candidate viewpoint that “sees” more cells is selected and added to a list of viewpoints to be used, then, the cells which are visible from this candidate viewpoint are removed from the octomap of all other candidate viewpoints. This process is repeated until there are no more candidates viewpoints, or the number of cells in the octomap for any remaining candidate viewpoint, is below a threshold defined by the user. As the number of samples, the stop criteria depends of the user experience and sensitivity, but the *a priori* octomap can support this decision, e.g. if my octomap size is 3 x 3 x 3 $m$ with a $0.3m^3$ resolution and stops if any remaining viewpoint sees more than 5 cells I could be missing information, otherwise if the resolution is $0.03m^3$ could I miss much less information.

**Algorithm 2** - Viewpoint

```
Input Octomap, Working Space (S) and Number of Random Viewpoints (X)
Ensure: Selected Viewpoints
    Initialization :
N - Vector of Viewpoints
for $N = 0$ to $X$ do
    $vp.x(i) =$ rand();
    $vp.y(i) =$ rand();
    $vp.z(i) =$ rand();
    if ($vp_occupancy < 0.5$) then
        $vp$ is placed in a free node
        Raycasting from $vp$
        $ds = sqrt(vp^2 - raycast^2)$
        if $ds > min_{range}$ && $ds < max_{range}$ && $ds > Safedistance$ then
            $vp$ is acceptable viewpoint
            $N \leftarrow N + vp$
        end if
    end if
end for
return $N$
```

### 3.2.3 Viewpoint Sequence

This step is similar to the second-phase of many other discrete CPP and view planning algorithms (TRUCCO *et al.*, 1997; CHEN; LI, 2004; CASHMORE *et al.*, 2013; BIRCHER *et al.*, 2015) (order the shortest tour to connect the viewpoints), in other words, solve the travelling salesman problem (TSP). As already aforementioned in Chapter 2 the TSP is a well-studied problem with many fast solvers (ROSENKRANTZ; STEARNS; LEWIS, 1974; GOLDEN *et al.*, 1980; REINELT, 1994; DANTZIG; FULKERSON; JOHNSON, 1954; LIN; KERNIGHAN, 1973; HELSGAUN, 2000).

To solve the TSP problem is required a cost matrix with the cost (distance) between the viewpoints. The cost matrix can be obtained calculating the euclidean distance between the viewpoints, which don't guaranteed a collision free path or using a collision free motion planning approach, in this case, RRT*. RRT* (KARAMAN; FRAZZOLI, 2011) is a asymptotic optimal variant of the Rapidly-exploring Random Tree (RRT) (LAVALLE; KUFFNER JR, 2001), which is a sampling-base algorithm which builds a tree of collision-free vehicle configurations to find a collision-free path.

In our case we decided to use the euclidean distance, despite the cost matrix obtained using RRT* is more precise, the time of running RRT* is not negligible.

Then we must solve the TSP problem, since we are dealing with a small number of viewpoints (i.e., below 100), we use the classic nearest neighbour approach which is an O$(n^2)$ algorithm (ROSENKRANTZ; STEARNS; LEWIS, 1974). We also test the Lin-Kernighan-Helsgaun Heuristic (LKH) TSP solver (HELSGAUN, 2000) and the results were very similar nevertheless the nearest neighbour approach is simpler to implement and easy to modify by the user.

### 3.2.4 Final Path

The final path is obtained connecting the viewpoints in the sequence calculated by the nearest neighbor approach using the RRT* algorithm (KARAMAN; FRAZZOLI, 2011). We also use the *a priori* model to avoid collisions. The RRT* has already been used in other discrete CPP and view planning methods as (BIRCHER *et al.*, 2015) and underwater applications (HERNÁNDEZ *et al.*, 2016). We have used the open motion planning library (OMPL) (SUCAN; MOLL; KAVRAKI, 2012) which has many state-of-the-art planning algorithms implemented.

The output of the proposed discrete CPP algorithm is a list of viewpoints $x, y, z, min\_yaw, max\_yaw$ and the paths that connects them, where $min\_yaw$ and $max\_yaw$ define the arc, with respect to the north, in which the vehicle can see a part of the object to inspect once in the viewpoint $x, y, z$. The main objective of the $min\_yaw$ and $max\_yaw$ is saving energy because our vehicle can rotate in $Z$ axis covering $360°$ although not for all viewpoints is necessary covering all the angles since for some *yaw* angles the sensor can't see any part of the environment which means we don't need to cover those angles.

The position $(x, y, z)$ is the location where the sensor must be placed, requiring a transformation from the sensor frame to the vehicle frame, as can be observed on Fig.21. For Girona 500 AUV equipped with a laser scan mounted over a pan-and-tilt unit we have those transformations:

- From robot to pan-and-tilt base: xyz = [0.4, 0, 0.5] rpy[3] = [-$\frac{\Pi}{2}$, $\frac{\Pi}{4}$ -$\frac{\Pi}{2}$];
- From the pan-and-til base to pan: xyz = [0, 0, 0.185] rpy = [$\frac{\Pi}{2}$, 0, $\Pi$];
- From the pan to tilt: xyz = [0, 0, 0] rpy =[$\frac{\Pi}{2}$,0,$\Pi$];
- From tilt to the multibeam: xyz = [0, 0, 0.06] rpy =[0, 0, 0];

Figure 25 shows in a 2D top view a simplified representation of Girona 500 AUV (i.e., the red dot), the $min\_yaw, max\_yaw$ angles, and in yellow the effective range of view from the sensor.

## 3.3 RESULTS

In order to evaluate and validate our discrete CPP method, we evaluate the algorithm in two realistic and challenging scenarios (Fig. 17), we choose those scenarios

[3] On this thesis we are using the roll-pitch-yaw (rpy) orientation notation (SCIAVICCO; SICILIANO, 2012)

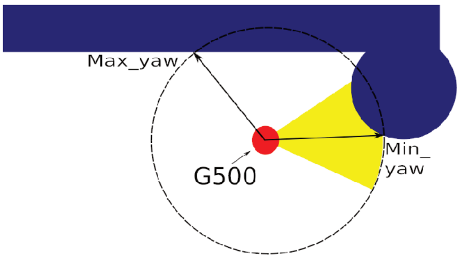
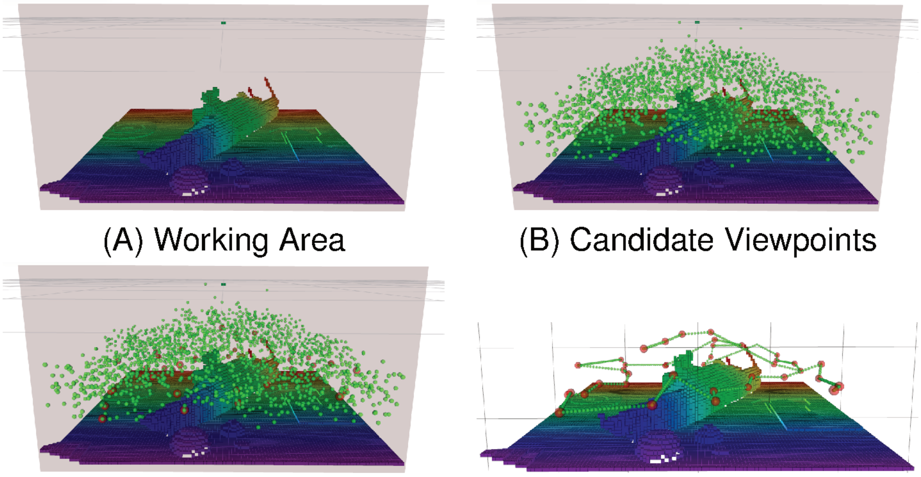


Figura 25 – $Min\_yaw$ and $max\_yaw$ and sensor view field extracted from (FLORIANI *et al.*, 2017).



(A) Working Area

(B) Candidate Viewpoints

(C) Selected Viewpoints

(D) Final Path

Figura 26 – Model-based discrete CPP steps for a sunken boat environment.

because there are no benchmark available for this problem. For the tests we consider Girona 500 AUV (RIBAS *et al.*, 2012) within 4 DoFs (i.e., Surge, Sway, Heave, and Yaw) equipped with a multibeam sensor with a range from 2 to 5 $m$ mounted over a pan and tilt unit in order to scan the environment. Even more we set a safety distance of 2 $m$ from the vehicle to any obstacle.

- Sunken Boat

  The first environment is a sunken boat laying on the seafloor (Fig. 17A). Figure 26 shows the steps of our method for the first scenario, Fig. 26A shows the working area set by the user, Fig. 26B presents all the candidate viewpoints generated. Fig. 26C shows the selected viewpoints in red and Fig. 26D presents the final path.

  We evaluate the algorithm comparing the number of selected viewpoints, the cost to perform the generated path and the estimate coverage varying the number of candidate viewpoints as following: 100, 200, 350, 500, 750, 1000, 2000, 2750, 3500 and 5000. The stop criteria is that no remaining candidate viewpoint can see more than 5 cells.

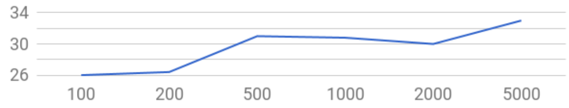


Figura 27 – Number of used viewpoints per number of candidate viewpoints.

To evaluate and compare the coverage obtained using our algorithm for different numbers of random candidate viewpoints, we define as the maximum possible coverage as the one achieved using a fine grid (0.25×0.25×0.25 $m$) distribution to pose the viewpoints. As previous mentioned use grid distribution is unrealistic to large scenarios, even to the test scenarios a powerful computer[4] was used because the computer used to run our algorithm didn't have enough memory. Although we use it as a comparing criteria.

This compare criteria is important because some areas are not visible taking into account the sensor limitations and safety conditions and would not be realistic considering that all the cells could be visible. Nevertheless, if we use as input a bathymetry map we can consider that all cells can be seen.

For a relatively small environment, like the one proposed here (20 x 20 x 20 $m$), this test generates 295,245 candidate viewpoints. However, for a larger environment the number of generated candidate viewpoints become intractable. The number of visible cells obtained in this test (grid-based) is considered the maximum number of cells (i.e., 100% coverage) that can be seen in this scenario.

Figure 27 presents the number of viewpoints selected (used) per number of candidate viewpoints. Figure 28 shows the minimum cost path in meters per number of candidate viewpoints. Those curves were built using the mean value obtained for each number of samples defined by the user after a large number of compilations. The variation of these curves could be caused by the use of a random base; when we perform a test with 2000 random candidate viewpoints we generate 2000 new random viewpoints instead of re-use the 1000 of the previous test.

Therefore, Fig. 29 shows the percentage of visible cells per number of candidate viewpoints. It is worth noting looking the three graphics that using 500 samples we are able to see 96.11%+-0.01% of what we can see using a grid distribution with almost 300,000 candidate viewpoints with a path 13.84% shorter and just consume around 2.5 minutes to compile our algorithm. If we compare the results of the grid distribution to the ones using 5000 samples, which consume around 17 minutes, the number of visible cells based on the grid distribution is 0.35%+-0.06% larger with a path 14.69% longer path.

[4] A desktop using a Intel ®Pentium ®II Xeon ®Processor 450 MHz, 2M Cache, 100 MHz and 16 GB of RAM

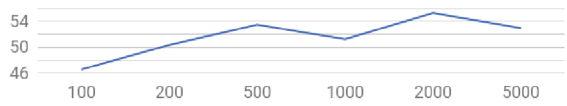


Figura 28 – Minimum path cost in $m$ per number of candidate viewpoints.

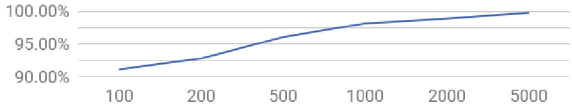


Figura 29 – Percentage of visible cells per number of candidate viewpoints.

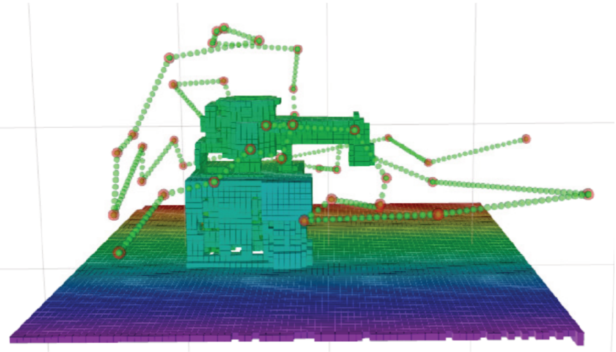


Figura 30 – Final Path for the subsea tree structure.

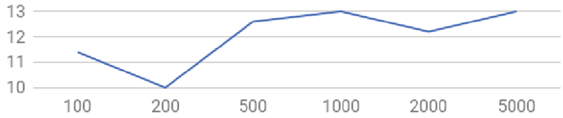


Figura 31 – Number of used viewpoints per number of candidate viewpoints.

- Tree Control Structure

  Our second test scenario is a subsea tree structure commonly used in the oil and gas industry posed in the sea-bottom (see the tree structure in Fig. 17A). The same tests varying the number of samples and keeping the same stop criteria, no remaining candidate viewpoint could see more than 5 cells were used. Figure 30 shows the final path and the viewpoints.

  The number of viewpoint selected per number of candidate viewpoints is presented in Fig. 31, the minimum cost path in meters per number of candidate viewpoints is shown in Fig. 32, and the percentage of visible cells per number of candidate viewpoints appears in Fig. 33. In the first two graphics we can observe a variation as well as in the first test scenario, caused too for the random base. It's worth to observe that for the tree control structure using 500 samples the coverage reaches 95.03%+-1.57% although the curve have some variations differently to the sunken boat.

All execution have been obtained with an Virtual Machine (Ubuntu 16.04) instal-

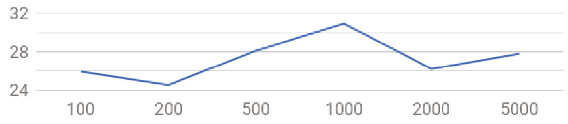


Figura 32 – Minimum path cost in meters per number of candidate viewpoints.

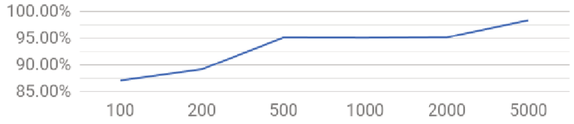


Figura 33 – Percentage of visible cells per number of candidate viewpoints.

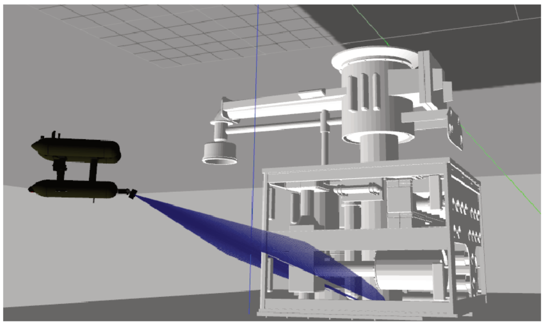


Figura 34 – Simulation with Gazebo.

led in a Intel i5 (2.7GHz) laptop with 8 GB of RAM with 4 GB of RAM dedicate to the Virtual Machine.

### 3.3.1 Simulation

We simulate the path planned with Gazebo (KOENIG; HOWARD, s.d.) using the Girona 500 AUV (fig. 34) equipped with a multibeam sensor mounted over a pan and tilt unit in order to scan the environment. To reduce the noise in the multibeam sensor, all planned trajectories scan the object of interest at a very short distance (i.e., between 2 and 5 meters).

To follow the path planned a high level controller is required to command the vehicle from a start point to the first viewpoint, following a collision-free path defined by the RRT*. When the viewpoint is reached the vehicle have to keep its position as static as possible, while commanding the pan and tilt unit to acquire a scan of the environment. Then if the difference between the $max\_yaw$ and $min\_yaw$ for the viewpoint is greater than 60 degrees, the vehicle has to change its heading to guarantee the coverage and perform a new scan. This process is repeated until all required orientations are covered. After this process the vehicle goes to the next viewpoint. As the positions are defined on the multibeam frame our controller to follow the path transform the positions from the multibeam frame to the vehicle frame.

The main result of our simulation is a point cloud of the inspected structure.

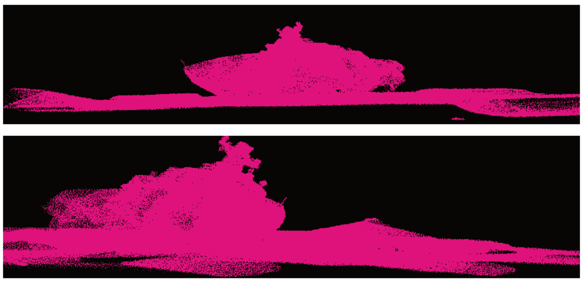


Figura 35 – Sunken boat reconstruction in point cloud.

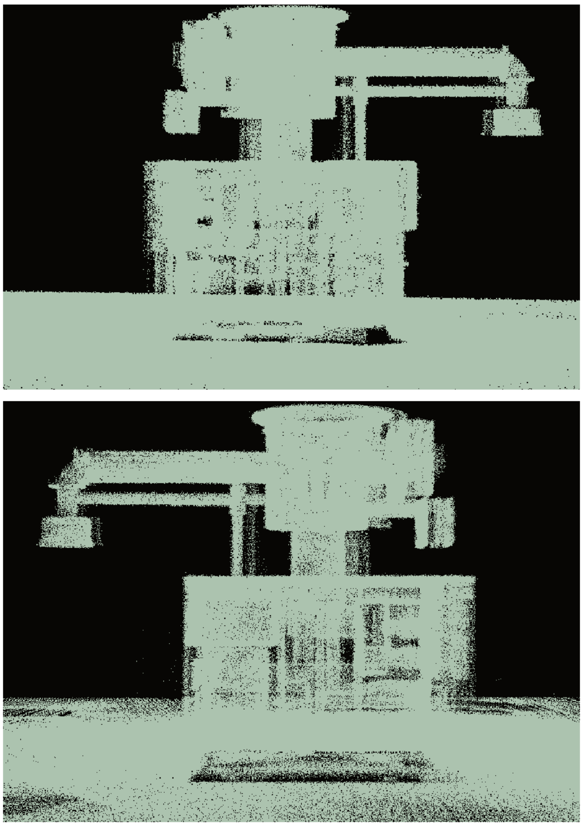


Figura 36 – Control tree structure reconstruction in point cloud.

Figure. 35 and Figure 36 show the raw point cloud from different points of view of the sunken boat and the tree structure setting 2000 candidate viewpoints, respectively. This value must be processed to remove or minimize the noise for example, although it is not the scope of this thesis.

## 3.4 DISCUSSION

This chapter presented a new model-based viewpoint planning for 3D complex structures which is able to produce in a short time (around a minute) high coverage results. Considering the coverage obtained using a grid distribution with almost 300,000 candidate viewpoints as complete coverage (100%), using our approach with at least 500 samples we can produce a shortest path, in less time, covering more than 95% in the examples we test. Using 5000 samples for both examples we almost coverage 100% with a shortest path and our algorithm can be easily implemented and ran in a personal computer, differently from the grid distribution. In other words, for 3D environments inspection tasks, random sample distribution can produce almost the same coverage as grid distribution with a shortest path in a shortest time.

CPP that is intractable in large areas using a grid distribution, mainly because the large number of viewpoints and operation required, can have a suitable solution using a few hundreds or thousands random samples.

Although some improvements to deal with limitations of our approach were required to extend the possibilities of the method.

First, it's necessary to have a novel approach to estimate the maximum cover that could be obtained using a specific vehicle and sensor because the problem became intractable for many scenarios if we have to use a fine grid. Then, we present a maximum coverage estimation approach and a data quality estimation to select the *best* viewpoints in the Chapter 4.

Another problem is the vehicle position estimation, usually in underwater environment the estimate position could have a large position error. It is important that the approach take into account this uncertainty of the position. In the Chapter 5 we present a method to evaluate the position uncertainty in the model-based viewpoint planning.

# 4 COVERAGE AND DATA QUALITY ESTIMATION

A new coverage estimation taking into account the vehicle and sensor constraints and a novel viewpoint selection criteria are presented in this chapter. In Section 4.1 we introduce the problem of coverage estimation, as well as, the data quality estimation as viewpoint selection criteria and some discussion about it. Our coverage estimation method is presented in Section 4.2 and the results in Section 4.3, a variation of the proposed algorithm is presented in Section 4.4 and a discussion in Section 4.5 The novel viewpoint selection based on the a proposed data quality index is presented in Section 4.6. Section 4.7 shows the results of the proposed criteria. Section 4.8 contains some discussion about this chapter.

## 4.1 INTRODUCTION

Most of the coverage (2D and 2.5D) path planning approaches are designed assuming that a complete coverage of the environment is achievable. Although for 3D complex structures sometimes is not feasible the complete coverage because parts could be occluded or hidden, or the vehicle can't achieve some positions for inspection. Moreover, the complete coverage could be not desirable as we presented in Section 2.6.1. Estimate the coverage is a computationally very expensive process.

Ellefsen, Lepikson e Albiez (2016) presents in their paper a coverage estimation approach based on the primitivies and the triangles mesh area. A sample uniformly distribution of viewpoints along the edges of the plans around the structure is used to determine the coverage estimation. The distance between these samples affects the precision and the processing time of the result. Then, the sum of the areas of the principles seen is compared with the total area of the environment. These approach is similar to the method we used in Section 3.3.

The coverage (visibility) estimation using octrees could be calculated based on the number and how well the faces of a voxel are seen from a position, as presented in the Chapter 2. Although the approach proposed by ososinski2014multi didn't consider that many faces are not visible from any position because are occluded or hidden.

Ellefsen, Lepikson e Albiez (2016) and Ososinski e Labrosse (2014) approaches starts from a viewpoint position to estimate the coverage. On the other hand, Englot e Hover (2017) starts from the structure primitivies to determine the minimum number of random viewpoints to guarantee the complete coverage.

## 4.2 COVERAGE ESTIMATION - INITIAL APPROACH

The novel approach that we propose calculate the maximum coverage and estimate the percentage of coverage for a set number of candidate viewpoints taking into account the vehicle and sensors constraints.

Initially we calculate the maximum coverage, which is a very time consuming process, to do it the initial input are similar with the inputs in Algorithm 1 in Chapter 3, an *a priori* octree model of the region of interesting and the vehicle and sensor constraints. Once again, consider an *a priori* model is an strong consideration, because is not always available an CAD or point cloud model, however if the *a priori* model was obtained performing a typical 2D or 2.5D bathymetry, as discussed on Section 3.2, isnot necessary to estimate the maximum coverage because all the cells can be visible on a discrete CPP, since the typical bathymetry will not represent occluded regions, as we can see on Fig. 19.

To calculate the maximum coverage, as well as Englot e Hover (2017), we look for an available viewpoint from each occupied voxel that sees the original voxel, as shown in Algorithm 3.

First for each occupied voxel (Algorithm 3, line 2) we do many ray casting in the inverse directions of the sensor field of view (Algorithm 3, lines 1 to 5). This step generates many viewpoints for each occupied voxel. The viewpoints are evaluated if they sees the original occupied voxel, if they are in a safe distance from the structures, obstacles and if they are reachable (Algorithm 3, line 7). Which means, if the vehicle can achieve this position, because is possible to have a position inside a structure that is in a safe distance but is not reachable. If all this criteria are filled we can count that occupied voxel as visible.

**Algorithm 3** - Coverage Estimation

```
Input Octomap
     Initialization :
 1: X - Number of voxels of the Octomap
 2: D - Vector of vectors (directions);
 3: for int $i = 0$ to $X$ do
 4:    P(i) = [x(i), Y(i), Z(i)]
 5:    if $P(i)$ is a occupied voxel then
 6:       for int $j = 0$ to D.size() do
 7:          Viewpoint = P(i) * D(j);
 8:          if (Viewpoint is not in a occupied) then
 9:             if (Viewpoint sees the node && Viewpoint is in a safe distance && View-
                point is reachable ) then
10:                The Occupied Node (voxel) is visible
11:             end if
12:          end if
13:       end for
14:    end if
15: end for
16: return Number of Visible Nodes
```

The Algorithm 3 output is the maximum number of visible nodes.

Englot e Hover (2017) proves that for a certain minimum number of random viewpoints is guaranteed the complete coverage, nevertheless that prove didn't consider the vehicle constraints and a safe distance. Although we can determine the maximum cover for specific vehicle, sensors if not complete coverage is possible.

## 4.3 COVERAGE ESTIMATION - INITIAL APPROACH - RESULTS

To validate our coverage estimate approach we performed some tests. We tested it, on the test case scenarios from Chapter 3 (sunken boat and the tree structure), the result obtained shows that using our discrete CPP, presented on Chapter 3 we could see more cells than our coverage estimation algorithm calculated as the maximum number of visible nodes. These happens because, if a ray originate on a Viewpoint ($VP_1$) hit any part of the cell, will return the center of the cell which means that cell is considered visible. However, it doesn't mean that the center of the cell. In other words, if the ray is originate on the center of the cell the $VP_1$ will not be visible.

If we test our approach in a scenario composed by tree parallels square planes (2.7 $m$ of diagonal) spaced of 0.4 $m$ as shown in Figure 38, and perform a cut, as shown in Fig. 38 we obtained a 2D representation, as shown in Figure 39, where each square represents a voxel, the blue nodes are visible based on the maximum coverage estimation, the pink and white nodes are visible from the Viewpoint (VP), and the yellow triangle represents the area where the VP must be pose to see the pink node based on the maximum coverage estimation algorithm. Nevertheless, is not possible to find a VP on the yellow area, since the planes are spaced by 0.4 $m$, which is smaller than the vehicle safety distance specification, i.e. any VP on the yellow area is closer to an obstacle than the safety distance.

This limitation is caused because to estimate the maximum coverage we use the center of the node and to our discrete CPP we use the whole node.

Then, this maximum visibility estimation is limited to environments with non close occluded regions, which do not fit on our complex structures inspection problem. Our goal is present an approach able to deal with occluded regions.

## 4.4 COVERAGE ESTIMATION - MODIFIED APPROACH

Our modified approach, based on our original (Section 4.2) maximum coverage estimation algorithm, initially for each cell we do many ray casting in the inverse directions of the sensor field of view (Algorithm 3, lines 1 to 5), starting in the cell's ($C_1$) center, as in Section 4.2. If there is no VP that see the cell ($C_1$), we change the origin of our rays to the center of the cell's ($C_1$) faces. If non VP that reaches the vehicle and sensor's requirements and sees the cell ($C_1$), we move the origin of the rays to the

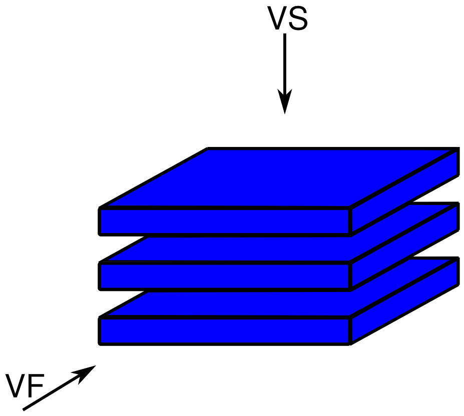


Figura 37 – Parallels square planes representation.

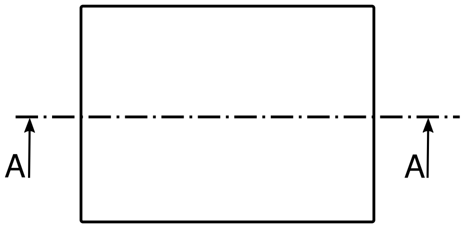


Figura 38 – Cut AA on the top View (VS).

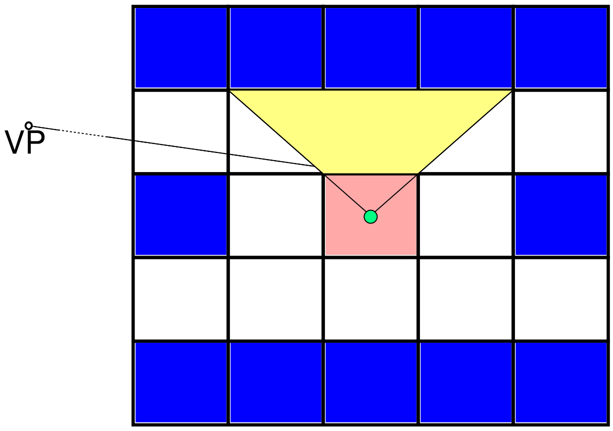


Figura 39 – Maximum coverage estimation and and accumulated views.

corners of the cell's ($C_1$) faces, then do many rays to look for a VP that sees the cell ($C_1$). This stratagem to test first the cell's center, then if necessary the faces and later the corners reduce the time consuming because the cell's center is enough to find a VP to most of the cells ($C_i$) and avoid excessive and unnecessary compilation.

We apply our method in a Tree Control Structure (Fig. 40), commonly used in the oil and gas industry posed in the sea-bottom measuring around $20$ x $20$ x $20$ m. We consider the Girona 500 AUV (RIBAS *et al.*, 2012) equipped with a laserscan mounted over a pan-and-tilt unit, sensor range from 2 to 5 m, and set 2 m as safety distance.

The maximum coverage percentage for the tree structure using the Girona 500 AUV with the set up constraints and limitations is 92.37%.

Figure 41 shows the curves of coverage per samples, where the blue circles

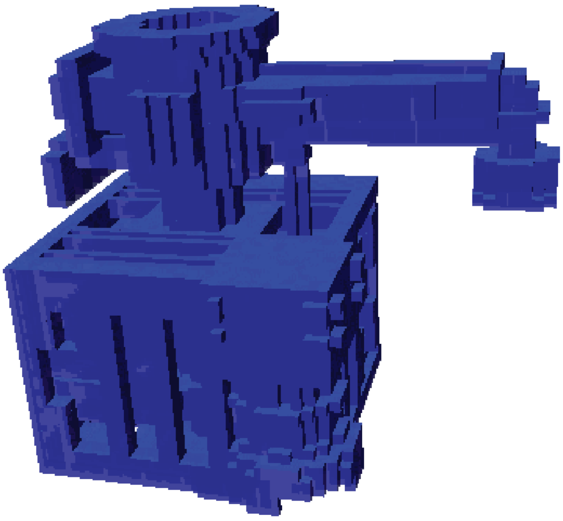


Figura 40 – Simplified Tree Control Structure

are the coverage behavior (%) per samples obtained using as stop criteria when no remaining viewpoint can see more than 5 nodes and the solid line is the fitted spline; The red square are the coverage behavior (%) per samples obtained using as stop criteria when no remaining viewpoint can see more than 1 node and the respective dashed line is the fitted spline. The 100% is the total number of occupied cells and 92.37% is the limit for our system (vehicle and sensor).

Fig. 42 shows the curves of path cost $(m)$ per samples, where the blue circles are the path cost (distance traveled) per number of samples obtained using as stop criteria that no remaining viewpoint can see more than 5 nodes and the solid line is the fitted spline; The red square are the path cost per number of samples obtained using as stop criteria that no remaining viewpoint can see more than 1 node and dashed line is the fitted spline. Those results were obtained selecting the best views as the viewpoints that “see” more cells.

Those graphics shows that the percentage of coverage grows when the number of samples increases, which is expected as, more samples means more options of viewpoints which sees more cells. Another interesting information obtained from Fig. 41 is that the stop criteria affects more when more samples are generated, once again, these can be explained because as much samples we have more options we have to chose the viewpoints to be used.

Looking to the Fig. 42, the path using as stop criteria when no remaining viewpoint can see more than 5 nodes decreases even producing more larger coverage when the number of samples grows, again, because more samples means more options to select the VPs, although when the stop criteria is that no remaining viewpoint can see more than 1 node variety but showing a small increasing. These increment is caused because for many samples there is a lot of viewpoints that sees more than 1 node and

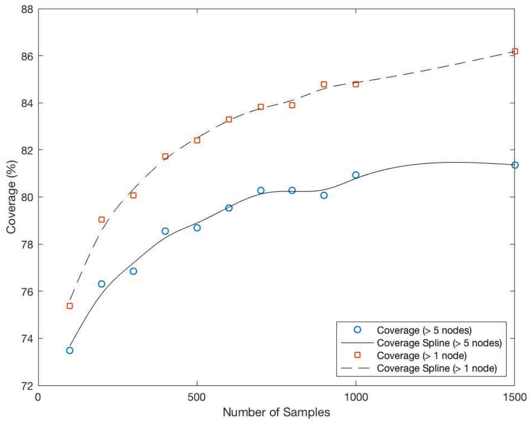


Figura 41 – Curves of coverage per samples.

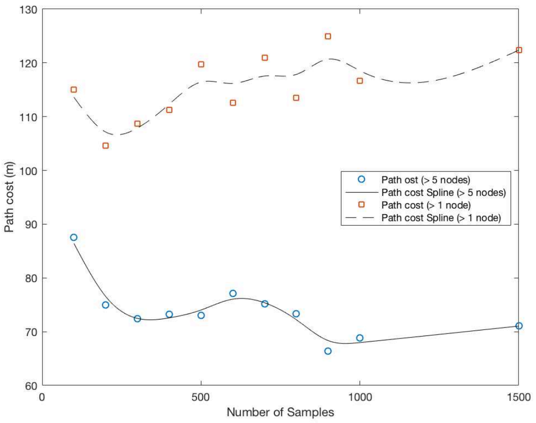


Figura 42 – Curves of path cost $(m)$ per samples.

more viewpoints probably will required a larger path.

## 4.5 DISCUSSION

Estimate the coverage is a novel problem based on the new and challenging applications of robotics and view planning because in many environments is unrealistic consider a complete coverage then we should have a measure to compare.

In this chapter we presented a new coverage estimation approach for complex structures with occluded parts, using the same metric to consider a cell visible as in the Chapter 3. This is a very time consuming process although didn't require much memory comparing with other methods as Englot e Hover (2017) or our model-based approach, mainly because we just save the occupied visible nodes.

Ellefsen, Lepikson e Albiez (2016) proposes a method similar with the one we use to evaluate the coverage of our algorithm for different numbers of random candidate viewpoints in Section 3.3. Although this process requires much more memory and taking into account the vehicle and sensor requirements have the limitations of our initial estimate coverage approach.

Despite the time consuming this process could be executed just once for the environment and allow us to determine a good initial number of random points for the model-based view planning. Moreover we can use the coverage estimation to compare different paths produced by our model-based algorithm.

## 4.6 DATA QUALITY ESTIMATION APPROACH

In inspection and reconstruction tasks several parameters can influence the backscatter (snippet) signal received by the multibeam, like power, gain, pulse length, absorption, spreading loss, sensor range, incident angle, seafloor roughness, acoustic properties, structure material etc. As better is our backscatter more information we can attribute to the seafloor and structure properties, to do this is important to correct the received backscatter signal to be invariant to parameters, as power, gain, pulse length, sensor range etc (PARNUM *et al.*, 2005).

To laser inspection tasks various sources of error can alter the gathered data, an important parameter is noise, which is affected by the distance from the sensor to the surface and the incident angle formed by the sensor ray and the surface's normal and other parameters (PRIETO *et al.*, 2003). For Prieto *et al.* (2003) the laser beam should be restricted to the range -35°$\leq \alpha \leq$ 35°at measurement time.

In some CPP approaches they use the local surface normal to select the viewpoints, as (SCHMID *et al.*, 2012; BIRCHER *et al.*, 2015) which can guarantee a low incident angle.

Although is not always possible to use the normals to point the viewpoints in complex structure.

Based on the impossibility to use the normals to positioned the viewpoints in various situations our previous presented approach (Chapter 3) uses a random distribution to generate the viewpoints and then we select as best viewpoints the ones that "sees" more nodes (FLORIANI *et al.*, 2017; PALOMERAS *et al.*, 2018), however the collected data quality is important in a inspection or reconstruction task. In other words, it's important how well the voxel can be seen. As previous mentioned, it depends of several parameters and factors, as the kind of sensor, the noise etc. But we can reduce the noise if we are able to use low incident angle and short distance between the sensor and the surface.

Ososinski e Labrosse (2014) proposes an equation to calculate the visibility of a voxel, the visibility is in a range from 0, which means not visible to 1 that means perfect conditions for visibility. Moreover they calculate the visibility for each voxel without taking into account the neighbor. Nevertheless Ososinski e Labrosse (2014) use the visibility value to estimate the coverage and not to select the "best"viewpoint.

In our proposal we calculate the data quality to use in the viewpoint selection

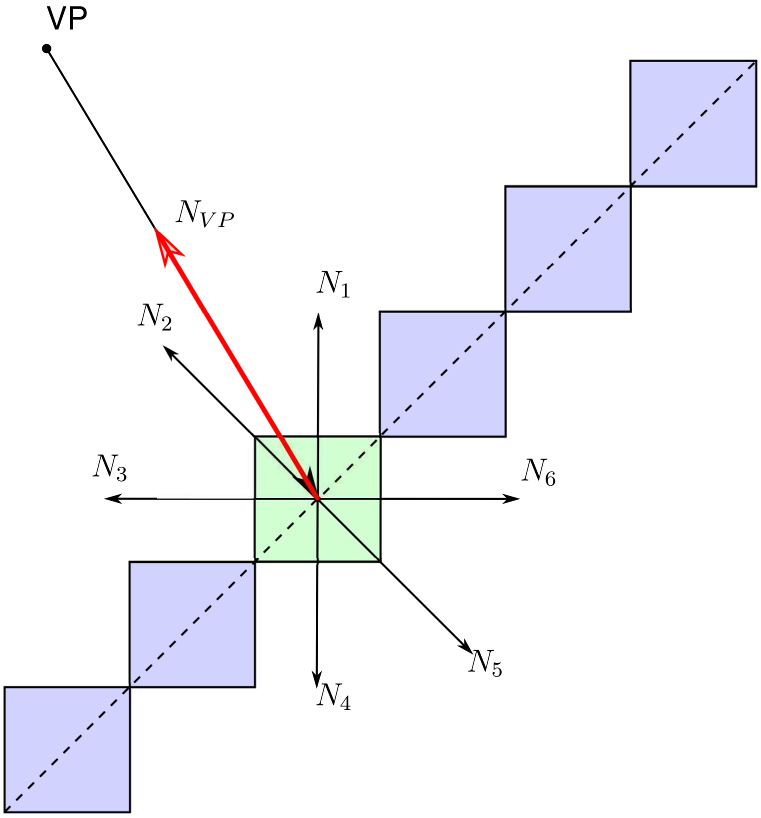


Figura 43 – Normals of a inclined plane.

(Section 3.2.2), with the data quality estimation we do not select as best viewpoint the one that see more cells but the one that have a higher value calculated using the number of visible cells and the data quality estimation of the VP. Furthermore we consider that the node is a part of a larger region, for this we use the neighbors to understand the region behavior using the normals, e.g. to calculate the normal of a cell[1] we evaluate the position of the occupied neighbors, as it's performed to generate the normals in a point cloud.

Initially we calculate the normals of a cell ($C_i$), these process can result in more than one normal for a cell ($C_i$), this occur because we are calculating the normals for a cell (point) in neighborhood with other cells that affects the surface representation and as the surface is open there is no inside or outside. Then we check if the normals is pointing to the viewpoint, the normals pointing to opposite direction to the viewpoint are eliminated because we are just interested on the ones pointing to the viewpoint.

An inclined plane (dotted line), the octree representation (violet and green cubes), the normals ($N_1, N_2 ... N_6$) and the Viewpoint (VP) are displayed in Figure 43. The red arrow ($N_{VP}$) is the vector originated in the center of the node and pointing to the viewpoint.

To check if the normal is pointing to the viewpoint, we calculate the dot product (Eq. 10) between the vector ($N_{VP}$) and the normal to evaluate the angle between the

[1] All the cells have the same size, shape and orientation

normal and the vector ($N_{VP}$).

$$\alpha = N_{VP} * Norm \tag{10}$$

- If $0 > \alpha > 90$ the vectors are in the same direction;
- If $\alpha = 90°$ the vectors are perpendicular, singularity;
- If $90° < \alpha < 180°$ the vectors are in the opposite direction.

The analyses of $\alpha$ allow us to determine which normals must be used to evaluate the visibility quality for each pair: VP and cell, as shown in Fig. 43,there is no reason to use $N_4$, $N_5$ and $N_6$ in order to evaluate the visibility quality, because those normals are pointing in an opposite direction of the $N_{VP}$.

After select the required normals, we calculate the *normal's mean* ($N_m$) to each cell ($C_i$), in the example presented in Fig 43, the mean has the same direction and orientation of $N_2$.

The next step is to estimate the angle ($\alpha_{mean}$) between the $N_{VP}$ and the $N_m$. Moreover, in this step we can define a maximum $\alpha$ that produce an acceptable result, otherwise we will consider that cell ($C_i$) as not seen.

As Ososinski e Labrosse (2014) we normalize the angle ($\alpha$) and the distance with the Eqs. 11 and 12.

$$A(VP, node) = \frac{-\alpha_{mean} + \frac{\pi}{2}}{\frac{\pi}{2}} \tag{11}$$

$$D(VP, node) = \frac{-ds + Max_{range}}{Max_{range} - Min_{range}} \tag{12}$$

With normalized $\alpha$ ($A(VP, node)$) and the distance ($D(VP, node)$)[2] between the node center and the *VP* (Eq. 13), the data quality ($V_q$) can be calculated for each cell ($C_i$).

$$V_q = a \cdot A(VP, node) + b \cdot D(VP, node) \tag{13}$$

Then we sum the $V_q$ of all visible nodes for each $VP$ (Eq. 14)

$$Total_{Vq}(VP) = \sum V_q \tag{14}$$

The next step is to select the best viewpoints. This selection uses the quality index ($VW$), calculated using Eq. 15, which combine the information from the $Total_V q(VP)$ and the number of visible nodes ($getOccupiedNodes(VP)$) by $VP$. The

[2] The $D(VP, node)$ is 1 when the distance (ds) between the sensor and the surface are equal to the $Min_{range}$ and 0 if the distance (ds) is equal to the $Max_{range}$

$Vw$ variate from 0 to 1, where 1 means an incident angle of 0°and a distance from sensor to the surface equal to the $Min_{range}$ and 0 the worst case.

$$Vw = \frac{Total_{Vq}(VP) + getOccupiedNodes}{Total_{Vq}(VP)} \tag{15}$$

The best viewpoint is the one that have a bigger $Vw$, then, the cells which are visible from this candidate viewpoint are removed from the octomap of all other candidate viewpoints and a new $Vw$ is calculated using the remaining visible nodes. This process is repeated until there are no more candidates viewpoints, or the number of cells in the octomap for any remaining candidate viewpoint.

## 4.7 DATA QUALITY ESTIMATION APPROACH - RESULTS

We create a metric using the incident angle and the distance between the viewpoint and the surface called data quality which can mainly reduce the noise of the data gathered to use as criteria to select the *best* viewpoints. To evaluate the novel select viewpoint criteria we performed some tests, in the tree structure environment as shown in Fig. 40 and compare with the results obtained on Chapter 3. Later, tests on real environment should be performed and the point cloud obtained compared. Although, those tests are out of the scope of this thesis.

Initially, we set the same number of samples (1000) in the two comparing selecting criteria cases (sees more cells and $Vw$), then we compare the results for the firsts selected viewpoint. The comparing criteria where the number of cells seen and the data quality for those by each case.

Table 3 shows the number of visible nodes, the view quality and the product of the number of visible nodes and the quality per viewpoint. After four views using the first approach we can 'see' 810 nodes with a quality mean of 0.315, them using the novel approach we 'see' less nodes (616) but with a quality mean of 0.570. Where the $VP_1$ is the *best* viewpoint using each case, the $VP_2$ is the second *best* viewpoint and then successively.

Using as stop criteria in the first case (sees more cells) that no remaining VP can see more than 5 cells and in the second case ($Vw$) that no remain VP can have a $Vw > 2.5$ the discrete coverage path planned has almost the same length in both cases but the first case sees more cells, around 1% more cells, for the example presented.

In other words, our novel approach can produce a better data quality, seen almost the same number of visible cells with almost the same path length, nevertheless, as previous mentioned, this approach should be tested on a real experiment, or in a simulation with some noise to produce more realistic tests, although those test are not in the scope of this thesis.

Tabela 3 – Comparation of different select viewpoints.

| | | Nodes$*$Quality $(VP_1)$ | Nodes | $Vw$ |
|---|---|---|---|---|
| $(VP_1)$ | ‘Sees‘ more cells | 40.749 | 238 | 0.171 |
| | $Vw$ | 117.389 | 184 | 0.638 |
| $(VP_2)$ | ‘Sees‘ more cells | 116.412 | 223 | 0.522 |
| | $Vw$ | 106.414 | 159 | 0.669 |
| $(VP_3)$ | ‘Sees‘ more cells | 82.393 | 176 | 0.468 |
| | $Vw$ | 75.297 | 163 | 0.462 |
| $(VP_4)$ | ‘Sees‘ more cells | 16.889 | 173 | 0.098 |
| | $Vw$ | 56.145 | 110 | 0.510 |

## 4.8 DISCUSSION

Another improvement of the proposed method is take into account the data quality to select the best views. There are other method that estimate the data quality nevertheless they didn’t use this information to select the used views. In other words, a data quality criteria allow us to select not the view that “sees” more nodes but the view that “sees” more and better the nodes. Considering that several parameters affects the backscatter (snippet) signal received by the multibeam and the error on laser sensors, it’s important to minimize the effects of those parameters in the data gathered, based on it, our novel approach proposes to minimize the incident angle and the distance from the sensor to the surface. A possible trouble of this algorithm is the normal obtaining, which is obtained using a discretization of the environment, for each patch a normal is calculated, then the normal obtaining depends of the environment discretization. If we have an *a priori* CAD model we can discretize as much as necessary or as our system can process.

Then our approach allow us to look for a solution combining data quality and energy consuming. Once more, real tests should be performed in real situations to evaluate the proposed approach, but it is out of the scope of this thesis.

To produce a more realistic result of the coverage we must consider the vehicle position uncertainty. In the Chapter 5 we present an approach for model-based view planning with position uncertainty.

# 5 DISCRETE CPP UNEDR POSITION UNCERTAINTY OF THE VEHICLE

A new coverage estimation considering the worst case of the uncertainty of position is presented in this chapter. In Section 5.1 we introduce the problem of position uncertainty and some discussion about it. Our method is presented in Section 5.2 and the results in Subsections 5.2.2 and 5.2.4. Section 5.3 contains some discussion about this chapter.

## 5.1 INTRODUCTION

Place the sensor is the key problem in the discrete and continuous CPP, although in the underwater environment the lack of GPS or others global localization systems makes the position error of the sensors systems grows up during the inspection, operation. The difference between the vehicle position and the planned ones is the pose error. Pose error can produce not well or uncovered regions during the inspection and covering regions that were not required to be inspected. Moreover, assumption of an idealized path execution is unrealistic (GALCERAN, 2014).

Several methods for motion planning under uncertainty were proposed, which can be classified in tree groups based on the origin of uncertainty: (a) motion uncertainty, (b) sensing uncertainty and partial observations, and (c) uncertainty on the environment (VASQUEZ-GOMEZ; SUCAR; MURRIETA-CID, 2017; VAN DEN BERG; ABBEEL; GOLDBERG, 2011).

To handle with uncertainty on a model-based continuous CPP, Galceran, Campos *et al.* (2014) proposes a re-planning approach using a Stochastic Trajectory Optimization for Motion Planning (STOMP) to reshape the path based on the sensors measurements *in situ* to keep the specified offset distance from the target on a bathymetry mapping task.

On a model-based discrete CPP for inspection tasks, Palomeras *et al.* (2018) uses a Iterative Closest Point (ICP)-based pose-graph Simultaneous Localization and Mapping (SLAM) to deal with the uncertainty in autonomous mapping of 3D structures minimizing the navigation error. Fig. 44 shows the navigation error in $m$ with respect to the ground truth according to dead reckoning navigation and corrected navigation using SLAM.

Those approaches try to minimize the pose uncertainty during the mapping. Nevertheless, even using a SLAM technique and loops in the path, the pose uncertainty could be larger than 1 m using a few viewpoints and a path length shorter than 40 meters.

Taking it into account, it's required that the model-based discrete CPP algorithm consider the uncertainty, preferably associated with a technique to minimize the error during the operation. As we do not have a curve to model the pose uncertainty, of

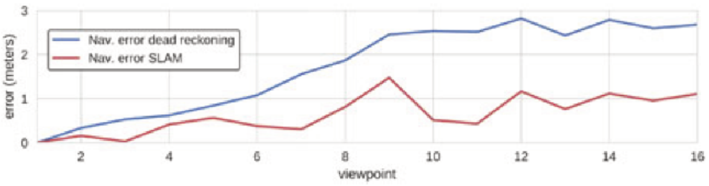


Figura 44 – Navigation error with respect to the ground: dead reckoning navigation (blue) and corrected navigation with SLAM (red) from Palomeras *et al.* (2018).

the Girona 500 AUV and there is no sensor to avoid the navigation drift, a compass could minimize the drift, although can induce further problems when working close to metallic structures. We estimate the minimum coverage under the worst case of pose uncertainty. Performing these estimation we will have the estimation for the best case, without pose uncertainty, and the worst case, with the maximum uncertainty using a ICP-based pose-graph SLAM technique on Girona 500 AUV. Future work should be done to improve the uncertainty estimation.

For a mobile disk robot under position and velocity uncertainty in a plane Bretl e Hutchinson (2013) proposes a new coverage path planning algorithm. Considering a bounded error of position and velocity they assumed the worst-case of uncertainty, which is the bound of the circle of the error. As their approach is conservative, the guarantee of the complete coverage comes at the cost of a longer path. Nevertheless they were the first to guarantee the complete coverage under bounded position and velocity error (GALCERAN, 2014).

Vasquez-Gomez, Sucar e Murrieta-Cid (2017) presents a next-best-view (NBV) holistic approach for three-dimensional (3D) object reconstruction with mobile manipulator under uncertainty. To do it they generate random samples in the control space based on an normal error distribution which produces many expected utility that are used to select the next-best-view.

## 5.2 COVERAGE ESTIMATION UNDER UNCERTAINTY APPROACH

The method we proposed generates a discrete CPP to coverage complex 3D structures under motion uncertainty situation. We propose two different approaches, then present their results, compare and discuss them.

### 5.2.1 Approach I - Normal Distribution

We propose a strategy using a normal distribution to estimate the error of the position, as presented in Algorithm 4, like Vasquez-Gomez, Sucar e Murrieta-Cid (2017). As our CPP is discrete, for each candidate viewpoint (sample) we generate $N$ possible positions that the vehicle can reach when go to that candidate viewpoint. This

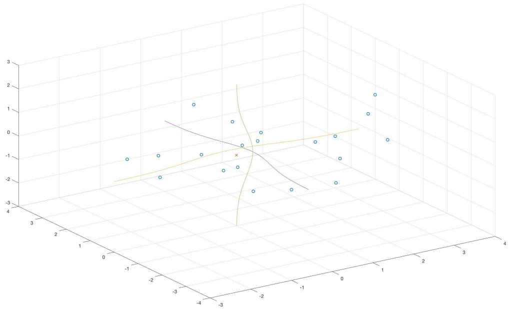


Figura 45 – Generated random points i each axis and combine them.

procedure differs from the one presented by Vasquez-Gomez, Sucar e Murrieta-Cid (2017) because they calculate the position error for each intermediate position when the vehicle travels between two viewpoints, however this procedure takes to much time.

Initially we generate the list of candidate viewpoint following the steps presented in Section 3.2 until the lines 6 of the Algorithm 1. For each candidate of view point we generate a list of Possible Reach Position (PRP) (Algorithm 4, line 3) under the uncertainty, where M is the number of random points, which are generated using a normal distribution where the center is the candidate viewpoint and the standard deviation is given by user based on the experience, knowledge of the vehicle and environment, Fig. 45 shows the combined result for the tree axis (X, Y, Z).

For each possible reach position generated we check the same criteria as the ones in Section 3.2.2, (a) the PRP must be inside a free cell (Algorithm 2, lines 6), (b) at least one occupied cell must be visible from the viewpoint under sensors constraints (Algorithm 2, lines 7 to 9), and (c) the viewpoint is in a safest distance from the obstacles (Algorithm 2, line 8 and 9). If any of these criteria is not accomplished the candidate viewpoint (VP) that generate that list of possible reach positions (PRP) is removed from the list of candidate views. Then we build an octomap of visible cells from each PRP and if a cell is visible for the candidate viewpoint and the whole list of PRP we say that cell is visible (Algorithm 4, line 15). Otherwise the cell is considered not visible, because that cell may not be visible.

**Algorithm 4** - Viewpoint

```
Input Octomap
Ensure: Selected Viewpoints
    Initialization :
VP - Candidate Viewpoints
for $VP = 0$ to $X$ do
    Norm = List of Possible reach position - PRP(VP)
    for i = 0 to M do
        if ($Norm[i]\_occupancy < 0.5$) then
            Norm[i] is placed in a free node
            Raycasting from Norm[i]
            $ds = sqrt(vp^2 - raycast^2)$
            if $ds > min_{range}$ && $ds < max_{range}$ && $ds > Safedistance$ then
                Norm[i] is acceptable
            end if
        end if
    end for
    if All Norm [i] are acceptable then
        Intersection of the visible cells from Norm and VP
    end if
end for
return $N$
```

### 5.2.2 Results - Normal Distribution

Initially we compare our approach under uncertainty against the algorithm presented in the Chapter 3. The first challenging environment was the Tree Control Structure. To make the test faster we clustered the cells to a new size of 0.15625 meters side length.

Table 5.2.2 presents the results of our simulations, in the first column are the results of the first algorithm (presented in Chapter 3), in the second column the results for the view path planning under uncertainty with a standard deviation of 1.0 meter and in the third column using a standard deviation of 2.0 meters. To all simulations we use 500 viewpoints (VP), 20 possible reach position per VP and stop if there is no VP that sees more than 1 cell. All execution times have been obtained with an Virtual Machine (Ubuntu 16.04) installed in a Intel i5 (2.7GHz) laptop with 8 GB of RAM with 4 GB of RAM dedicate to the Virtual Machine.

| | Original | Std = 1.0 m | Std = 2.0 m |
|---|---|---|---|
| N. of used views | 109 | 125 | 116 |
| N. of visible cells | 8173 | 6545 | 6405 |
| Path Lenght (m) | 184.35 | 221.03 | 195.28 |
| Time Consuming (s) | 525.06 | 587.05 | 587.15 |

### 5.2.3 Approach II - Worst Case

Our first approach deals with the uncertainty using a probabilistic way, then we can probabilistic guarantee the coverage. But in some cases we have to guarantee the coverage even in the worst case, as Bretl e Hutchinson (2013). Initially we consider our position error bounded by an sphere, because our problem is 3D, the size of the sphere is given by the knowledge of maximum error of the position of the vehicle in operation, with an online SLAM. For the G500 based on the experiments performed and the position error in Figure 44 we can determine the radius of the sphere.

In other words, for a specified Viewpoint posed on $(x, y, z)$, the vehicle could be in any position inside the uncertainty ball of constant radius $\alpha$ centred on $(x, y, z)$. Then we should determine the visibility for all points in the bound of uncertainty ball and do the intersection of them, because a cell is visible from Viewpoint posed on $(x, y, z)$ under uncertainty just if is visible from all the positions in the uncertainty ball.

Evaluate all the points of the sphere is unrealistic, on the other hand, discretize the uncertainty ball implies in an error in the result - *e.g.*, using an icosahedron (12 vertices) we have an error of 7.7% in the uncertainty ball. Based on this, as well as, we looking for the worst case we decided to use a polyhedron circumscribing the uncertainty sphere. Tree different polyhedron was used, an cube (Fig. 46A), an icosahedron (Fig. 46B) and a polyhedron with 42 vertices (Fig. 46C).

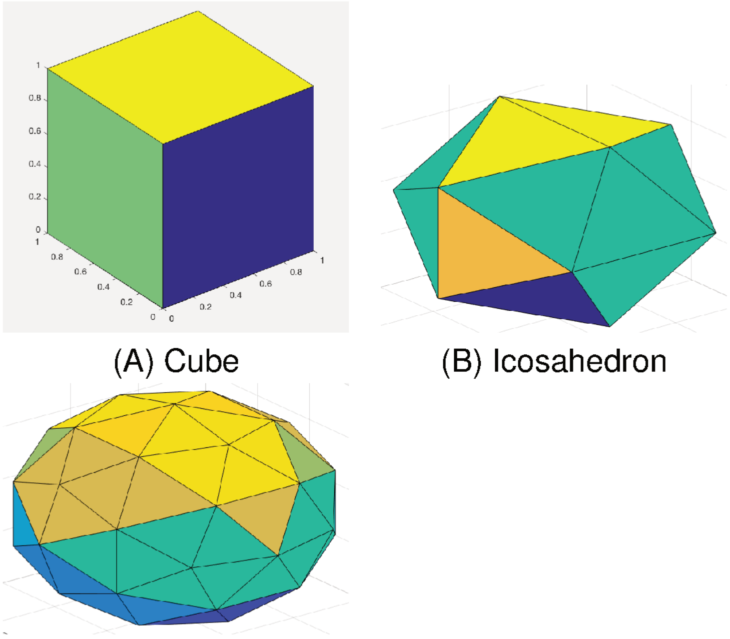
Using the cube we are over estimation the worst case in 41.42% using 8 new viewpoints, with the icosahedron in 23.61% using 12 new viewpoints and the polyhedron with 42 vertices 6.56% using 42 new viewpoints. More vertices to circumscribe the uncertainty sphere produces a closer result comparing with the theoretical of the sphere although this requires more time and memory. Specially to perform the intersection between the uncertainty viewpoints.

### 5.2.4 Results - Worst Case

To evaluate our approach we compare the result of our algorithm presented in the Chapter 3, the algorithm presented in Section 5.2.1 and the approach presented in Section 5.2.3 in the same environment (Tree Control Structure).

Table 5.2.4 presents the results of our tests, in the first column are the results of the first algorithm (presented in Chapter 3), in the second column the results for the view path planning under uncertainty with a standard deviation of $\frac{1}{3}$ meter (20 points)[1], in the third column a cube circumscribing the uncertainty 1 $m$ radius sphere bound, fourth column the icosahedron and fifth column a 42 vertices polyhedron. All execution times have been obtained with an Virtual Machine (Ubuntu 16.04) installed in a Intel i5 (2.7GHz) laptop with 8 GB of RAM with 4 GB of RAM dedicate to the Virtual Machine.

[1] These standard deviation guarantee that 99.7% of the points are inside the sphere of radius 1.0 meter.



(A) Cube

(B) Icosahedron

(C) 42 Vertices

Figura 46 – Worst case - polyhedrons representation.

| | Original | Normal | Cube | Icosahedron | 42 Vertices |
|---|---|---|---|---|---|
| N. of used views | 109 | 128 | 120 | 114 | 117 |
| N. of visible cells | 8173 | 6629 | 6010 | 6503 | 6536 |
| Path Length (m) | 184.35 | 212.26 | 208.18 | 191.76 | 198.46 |
| Time Consuming (s) | 525.06 | 571.90 | 577.88 | 575.19 | 575.87 |

## 5.3 DISCUSSION

Motion uncertainty is a key problem in coverage and viewpoint path planning. Minimize the pose uncertainty is essential, specially in the underwater environment, because the accumulated pose error could be large (affecting in the inspection). Although the uncertainty is not negligible even using the techniques to minimize, e.g. re planning, SLAM.

In this chapter we presented a discrete CPP under the worst case of motion uncertainty. First we propose an approach using normal distribution to estimate the probabilistic pose of the vehicle when try to reach a defined position. Then an algorithm to the worst case is introduced.

Different from the approach propose by Bretl e Hutchinson (2013) our worst case is not a function of time, because we first evaluate and select the viewpoints and then the sequence of viewpoints and the path. To consider the uncertainty as a time function we must evaluate the uncertainty and the path, together. In other words, this would require a new model-based CPP which select the viewpoints, the sequence and path, together.

Our algorithm is conservative and guarantee the coverage at the cost of a longer path, although is the first approach for a model-based discrete CPP application in a 3D environment.

# 6 CONCLUSION

This chapter concludes this thesis by presenting a summary of work completed in Section 6.1, reviewing the contributions of the thesis in Section 6.2 and future work are outlined in Section 6.3.

## 6.1 SUMMARY OF COMPLETED WORK

This thesis has addressed the discrete CPP problem in the underwater domain to inspection 3D structures tasks. In Chapter 1 we introduced the path planning, view planning and coverage path planning, as well as, presented the objectives of this thesis. In Chapter 2 we reviewed the state of the art of path planning, view planning and coverage path planning. The review puts emphasis on the model-based discrete CPP problem, was not possible to put an emphasis on the underwater domain because there is not much work for the underwater environment. The existence of few works addressing the inspection on the subsea environment and the high importance of this environment demonstrate the necessity of novel researches focus on the subsea domain, because it have many particularities, as the lack of a ubiquitous absolute positioning (like a GPS), sensors limitation, the reduced visibility, currents, navigation drift and all sorts of unexpected disturbances.

In Chapter 3 we proposed a novel model-based discrete algorithm for inspection complex 3D underwater structures (as industrial facilities such as oil fields, dams or offshore wind turbines) using an AUV, although this algorithm allow ROV pilots to find the path to follow. The advantages of this method is produce a high coverage results, more than 95% using our approach with at least 500 samples on the test scenarios, considering as 100% the coverage obtained using a fine grid distribution with almost 300,000. Using 5000 samples for both examples we almost coverage 100% with a shortest path comparing to the fine grid distribution. It's important to mention that if parts of the cells are visible that cell is said visible, then the size of the cell from the *a priori* model will have an important influence on the coverage of each cell. Moreover the method can be adapted to other vehicles, acquisition sensor.

To have a more better coverage estimation, which is an important comparing criteria specially because in our model-based discrete CPP we do not guarantee the complete coverage, we presented a novel estimation covarege algorithm in Chapter 4. This is a computationally very expensive process although allow us to estimate the maximum coverage achievable with a specific vehicle and his set of sensors, moreover helps us to choose the number of samples for our model-based approach and the most suitable path (coverage per length traveled). We also introduce an quality index in this Chapter to select the viewpoints to be used based on the number of visible cells and the index data quality proposed

Chapter 5 presents a model-based discrete CPP and estimate the coverage under the worst case of pose uncertainty. The lack of GPS or other global localization systems in the underwater environment makes the position error of the vehicle grows up during the operation, some approaches are being used to minimize the pose uncertainty but didn't eliminate. Then we propose an conservative algorithm that estimates the coverage for the worst case.

## 6.2 REVIEW OF CONTRIBUTIONS

This thesis has contributed to model-based viewpoint path planning and to the field of underwater robotics in following main respects.

- *MODEL-BASED DISCRETE CPP* This thesis has proposed a novel 3D model-based discrete CPP method to inspect complex 3D underwater structures.
- *COVERAGE ESTIMATION* This thesis proposes a new coverage estimation taking into account the vehicle and sensor limitations.
- *MODEL-BASED DISCRETE CPP UNDER THE WORST CASE OF POSE UNCERTAINTY* We present an offline discrete CPP algorithm which is able to estimate the coverage under the worst case of pose uncertainty. The proposed method goes beyond traditional discrete coverage path planners, where uncertainty is ignored in the planning phase.

## 6.3 COMPELLING AREAS FOR FUTURE WORK

This thesis opens the door for many possible future works.

- *Future work in 3D inspection tasks* The 3D model-based viewpoint path planning method we presented in Chapter 3 produced interesting results in simulation. Besides we use a simulation environment to evaluate our approaches it is interesting an application in real environments with AUV or other underwater device.
- *Coverage under Uncertainty* The model-based discrete CPP presented deal with the uncertainty considering the worst case and produce newsworthy results. An interesting future reseach is modeling the uncertainty as function of time and/or distance travelled and/or other parameters that are relevant to the uncertainty.

## 6.4 LIST OF PUBLICATIONS

Next, we present a list of publications that have resulted from this thesis and connect them to the content of this document.

- FLORIANI, B. L. et al. Model-based underwater inspection via viewpoint planning using octomap. In: IEEE. **Oceans, 2017.** [S.l.], 2017. p. 1-8.[1]
- FLORIANI, B. L. et al. Offline viewpoint planning for underwater using octomap. In: COBEM, 2017. **24th ABCM International Congress of Mechanical Engineering**
(doi://10.26678/ABCM.COBEM2017.COB17-0474).

A manuscript entitled *Model-Based Viewpoint Path Planning Using View Quality Estimation* has been submitted for publication in the China Ocean Engineering. And two other manuscript are being written.

[1] This paper was selected to participate of the Student Poster Competition

# REFERÊNCIAS

ACAR, Ercan U *et al.* Morse decompositions for coverage tasks. **The international journal of robotics research**, SAGE Publications Sage UK: London, England, v. 21, n. 4, p. 331–344, 2002.

ALEXIS, Kostas *et al.* Uniform coverage structural inspection path–planning for micro aerial vehicles. *In:* IEEE. INTELLIGENT Control (ISIC), 2015 IEEE International Symposium on. [*S.l.*: *s.n.*], 2015. p. 59–64.

ALMADHOUN, Randa *et al.* A survey on inspecting structures using robotic systems. **International Journal of Advanced Robotic Systems**, SAGE Publications Sage UK: London, England, v. 13, n. 6, p. 1729881416663664, 2016.

ANTONELLI, Gianluca; FOSSEN, Thor I; YOERGER, Dana R. Underwater robotics. *In:* SPRINGER handbook of robotics. [*S.l.*]: Springer, 2008. p. 987–1008.

APPLEGATE, David L *et al.* **The traveling salesman problem: a computational study**. [*S.l.*]: Princeton university press, 2006.

ATKAR, Prasad N; CHOSET, Howie *et al.* Exact cellular decomposition of closed orientable surfaces embedded in/spl Rfr//sup 3. *In:* IEEE. ROBOTICS and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. [*S.l.*: *s.n.*], 2001. p. 699–704.

ATKAR, Prasad N; GREENFIELD, Aaron *et al.* Hierarchical segmentation of surfaces embedded in R3 for auto-body painting. *In:* IEEE. ROBOTICS and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. [*S.l.*: *s.n.*], 2005. p. 572–577.

ATKAR, Prasad N; GREENFIELD, Aaron *et al.* Uniform coverage of automotive surface patches. **The International Journal of Robotics Research**, SAGE Publications, v. 24, n. 11, p. 883–898, 2005.

BANTA, Joseph E *et al.* Best-next-view algorithm for three-dimensional scene reconstruction using range images. *In:* INTERNATIONAL SOCIETY FOR OPTICS e PHOTONICS. INTELLIGENT Robots and Computer Vision XIV: Algorithms, Techniques, Active Vision, and Materials Handling. [*S.l.*: *s.n.*], 1995. p. 418–430.

BARRAQUAND, Jerome; LATOMBE, Jean-Claude. Robot motion planning: A distributed representation approach. **The International Journal of Robotics Research**, Sage Publications Sage CA: Thousand Oaks, CA, v. 10, n. 6, p. 628–649, 1991.

BARRAQUAND, Jérôme; LATOMBE, Jean-Claude. A Monte-Carlo algorithm for path planning with many degrees of freedom. *In:* IEEE. PROCEEDINGS., IEEE International Conference on Robotics and Automation. [*S.l.*: *s.n.*], 1990. p. 1712–1717.

BIRCHER, Andreas *et al.* Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. *In:* IEEE. ROBOTICS and Automation (ICRA), 2015 IEEE International Conference on. [*S.l.*: *s.n.*], 2015. p. 6423–6430.

BOHLIN, Robert; KAVRAKI, Lydia E. Path planning using lazy PRM. *In:* IEEE. ROBOTICS and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on. [*S.l.*: *s.n.*], 2000. p. 521–528.

BOOR, Valérie; OVERMARS, Mark H; VAN DER STAPPEN, A Frank. The Gaussian sampling strategy for probabilistic roadmap planners. *In:* IEEE. ROBOTICS and automation, 1999. proceedings. 1999 ieee international conference on. [*S.l.*: *s.n.*], 1999. p. 1018–1023.

BRETL, Timothy; HUTCHINSON, Seth. Robust coverage by a mobile robot of a planar workspace. *In:* IEEE. ROBOTICS and Automation (ICRA), 2013 IEEE International Conference on. [*S.l.*: *s.n.*], 2013. p. 4582–4587.

CAO, Zuo LIang; HUANG, Yuyu; HALL, Ernest L. Region filling operations with random obstacle avoidance for mobile robots. **Journal of Robotic systems**, Wiley Online Library, v. 5, n. 2, p. 87–102, 1988.

CARRERAS, Marc *et al.* Sparus II, design of a lightweight hovering AUV. *In:* SARTI. MARTECH 2013 5th International Workshop on Marine Technology. [*S.l.*: *s.n.*], 2013.

CASHMORE, Michael *et al.* Planning inspection tasks for auvs. *In:* IEEE. OCEANS-SAN Diego, 2013. [*S.l.*: *s.n.*], 2013. p. 1–8.

CHEN, Shengyong; LI, Youfu; KWOK, Ngai Ming. Active vision in robotic systems: A survey of recent developments. **International Journal of Robotics Research**, Sage Publications, Inc., v. 30, n. 11, p. 1343–1377, 2011.

CHEN, SY; LI, YF. Automatic sensor placement for model-based robot vision. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 34, n. 1, p. 393–408, 2004.

CHIN, Wei-pang; NTAFOS, Simeon. Optimum watchman routes. **Information Processing Letters**, Elsevier, v. 28, n. 1, p. 39–44, 1988.

CHOSET, Howie. Coverage for robotics–A survey of recent results. **Annals of mathematics and artificial intelligence**, Springer, v. 31, n. 1, p. 113–126, 2001.

CHOSET, Howie. **Robotic Motion Planning: Sample-Based Motion Planning**. 2007. Disponível em: `https://www.cs.cmu.edu/~motionplanning/`. Acesso em: 17 jan. 2019.

CHOSET, Howie M *et al.* **Principles of robot motion: theory, algorithms, and implementation**. [*S.l.*]: MIT press, 2005.

CHOSET, Howie; PIGNON, Philippe. Coverage path planning: The boustrophedon cellular decomposition. *In:* SPRINGER. FIELD and service robotics. [*S.l.*: *s.n.*], 1998. p. 203–209.

CHOSET, Howie; PIGNON, Philippe. Coverage path planning: The boustrophedon decomposition. *In:* PROCEEDINGS of the International Conference on Field and Service Robotics. [*S.l.*: *s.n.*], 1997. p. 3–91.

CHOSET, Howie *et al.* Exact cellular decompositions in terms of critical points of morse functions. *In:* IEEE. ROBOTICS and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on. [*S.l.*: *s.n.*], 2000. p. 2270–2277.

CHRISTOFIDES, Nicos. **Worst-case analysis of a new heuristic for the travelling salesman problem**. [*S.l.*], 1976.

CHVATAL, Vasek. A combinatorial theorem in plane geometry. **Journal of Combinatorial Theory, Series B**, Elsevier, v. 18, n. 1, p. 39–41, 1975.

COLES, Amanda Jane *et al.* Forward-Chaining Partial-Order Planning. *In:* ICAPS. [*S.l.*: *s.n.*], 2010. p. 42–49.

CONNOLLY, CI. The determination of next best views. *In:* IEEE. ROBOTICS and automation. Proceedings. 1985 IEEE international conference on. [*S.l.*: *s.n.*], 1985. p. 432–435.

CORMEN, Thomas H *et al.* **Introduction to algorithms**. [*S.l.*]: MIT press, 2009.

DANTZIG, George; FULKERSON, Ray; JOHNSON, Selmer. Solution of a large-scale traveling-salesman problem. **Journal of the operations research society of America**, INFORMS, v. 2, n. 4, p. 393–410, 1954.

DEB, Kalyanmoy *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE transactions on evolutionary computation**, IEEE, v. 6, n. 2, p. 182–197, 2002.

DIJKSTRA, Edsger W. A note on two problems in connexion with graphs. **Numerische mathematik**, Springer, v. 1, n. 1, p. 269–271, 1959.

DOBSON, Andrew; BEKRIS, Kostas E. Improving sparse roadmap spanners. *In:* IEEE. ROBOTICS and Automation (ICRA), 2013 IEEE International Conference on. [*S.l.*: *s.n.*], 2013. p. 4106–4111.

DOBSON, Andrew; KRONTIRIS, Athanasios; BEKRIS, Kostas E. Sparse roadmap spanners. *In:* ALGORITHMIC Foundations of Robotics X. [*S.l.*]: Springer, 2013. p. 279–296.

DORIGO, Marco; GAMBARDELLA, Luca Maria. Ant colony system: a cooperative learning approach to the traveling salesman problem. **IEEE Transactions on evolutionary computation**, IEEE, v. 1, n. 1, p. 53–66, 1997.

DORNHEGE, Christian; EYERICH, Patrick *et al.* Semantic attachments for domain-independent planning systems. *In:* NINETEENTH International Conference on Automated Planning and Scheduling. [*S.l.*: *s.n.*], 2009.

DORNHEGE, Christian; KLEINER, Alexander; KOLLING, Andreas. Coverage search in 3D. *In:* IEEE. SAFETY, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on. [*S.l.*: *s.n.*], 2013. p. 1–8.

ELLEFSEN, Kai Olav; LEPIKSON, Herman A; ALBIEZ, Jan C. Planning inspection paths through evolutionary multi-objective optimization. *In:* ACM. PROCEEDINGS of the 2016 on Genetic and Evolutionary Computation Conference. [*S.l.*: *s.n.*], 2016. p. 893–900.

ENGLOT, Brendan J. **Sampling-based coverage path planning for complex 3D structures**. 2012. Tese (Doutorado) – Massachusetts Institute of Technology.

ENGLOT, Brendan; HOVER, Franz. Planning complex inspection tasks using redundant roadmaps. *In:* ROBOTICS Research. [*S.l.*]: Springer, 2017. p. 327–343.

FAIGL, Jan. Approximate solution of the multiple watchman routes problem with restricted visibility range. **IEEE transactions on neural networks**, IEEE, v. 21, n. 10, p. 1668–1679, 2010.

FARSI, M *et al.* Robot control system for window cleaning. *In:* AUTOMATION and Robotics in Construction Xi. [*S.l.*]: Elsevier, 1994. p. 617–623.

FLORIANI, Bruno Locks *et al.* Model-Based Underwater Inspection via Viewpoint Planning using Octomap. *In:* IEEE. OCEANS, 2017. [*S.l.*: *s.n.*], 2017. p. 1–8.

GABRIELY, Yoav; RIMON, Elon. Spanning-tree based coverage of continuous areas by a mobile robot. **Annals of mathematics and artificial intelligence**, Springer, v. 31, n. 1-4, p. 77–98, 2001.

GALCERAN, E. **Coverage path planning for autonomous underwater vehicles**. 2014. Tese (Doutorado) – Ph. D. Thesis, University of Girona.

GALCERAN, Enric; CAMPOS, Ricard *et al.* Coverage path planning with realtime replanning for inspection of 3d underwater structures. *In:* IEEE. ROBOTICS and Automation (ICRA), 2014 IEEE International Conference on. [*S.l.*: *s.n.*], 2014. p. 6586–6591.

GALCERAN, Enric; CARRERAS, Marc. Planning coverage paths on bathymetric maps for in-detail inspection of the ocean floor. *In:* IEEE. ROBOTICS and Automation (ICRA), 2013 IEEE International Conference on. [*S.l.*: *s.n.*], 2013. p. 4159–4164.

GOLDEN, Bruce *et al.* Approximate traveling salesman algorithms. **Operations research**, INFORMS, v. 28, 3-part-ii, p. 694–711, 1980.

GUIBAS, Leonidas *et al.* Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. **Algorithmica**, Springer, v. 2, n. 1-4, p. 209–233, 1987.

HARALICK, Robert M; STERNBERG, Stanley R; ZHUANG, Xinhua. Image analysis using mathematical morphology. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, n. 4, p. 532–550, 1987.

HART, Peter E; NILSSON, Nils J; RAPHAEL, Bertram. A formal basis for the heuristic determination of minimum cost paths. **IEEE transactions on Systems Science and Cybernetics**, IEEE, v. 4, n. 2, p. 100–107, 1968.

HELSGAUN, Keld. An effective implementation of the Lin–Kernighan traveling salesman heuristic. **European Journal of Operational Research**, Elsevier, v. 126, n. 1, p. 106–130, 2000.

HERNÁNDEZ VEGA, Juan David *et al.* Online path planning for autonomous underwater vehicles under motion constraints. Universitat de Girona, 2017.

HERNÁNDEZ, Juan David *et al.* Autonomous Underwater Navigation and Optical Mapping in Unknown Natural Environments. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 16, n. 8, p. 1174, 2016.

HERT, Susan; TIWARI, Sanjay; LUMELSKY, Vladimir. A terrain-covering algorithm for an AUV. *In:* UNDERWATER Robots. [*S.l.*]: Springer, 1996. p. 17–45.

HORNUNG, Armin *et al.* OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. **Autonomous Robots**, 2013. Software available at `http://octomap.github.com`. DOI: `10.1007/s10514-012-9321-0`. Disponível em: `http://octomap.github.com`.

HSU, David; LATOMBE, J-C; MOTWANI, Rajeev. Path planning in expansive configuration spaces. *In:* IEEE. ROBOTICS and Automation, 1997. Proceedings., 1997 IEEE International Conference on. [*S.l.*: *s.n.*], 1997. p. 2719–2726.

HUANG, Wesley H. Optimal line-sweep-based decompositions for coverage algorithms. *In:* IEEE. ROBOTICS and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on. [*S.l.*: *s.n.*], 2001. p. 27–32.

HUANG, Wesley H. The minimal sum of altitudes decomposition for coverage algorithms. **Rensselaer Polytechnic Institute Computer Science Technical Report**, v. 6, 2000.

HUANG, Yuyu; CAO, Z; HALL, E. Region filling operations for mobile robot using computer graphics. *In:* IEEE. ROBOTICS and Automation. Proceedings. 1986 IEEE International Conference on. [*S.l.*: *s.n.*], 1986. p. 1607–1614.

JACOBI, Marco. Autonomous inspection of underwater structures. **Robotics and Autonomous Systems**, Elsevier, v. 67, p. 80–86, 2015.

JANOUŠEK, Petr; FAIGL, Jan. Speeding up coverage queries in 3D multi-goal path planning. *In:* IEEE. ROBOTICS and Automation (ICRA), 2013 IEEE International Conference on. [*S.l.*: *s.n.*], 2013. p. 5082–5087.

JIN, Jian; TANG, Lie. Coverage path planning on three-dimensional terrain for arable farming. **Journal of Field Robotics**, Wiley Online Library, v. 28, n. 3, p. 424–440, 2011.

JING, Wei *et al.* Sampling-based view planning for 3d visual coverage task with unmanned aerial vehicle. *In:* IEEE. INTELLIGENT Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. [*S.l.*: *s.n.*], 2016. p. 1808–1815.

KAFKA, Přemysl; FAIGL, Jan; VÁŇA, Petr. Random Inspection Tree Algorithm in visual inspection with a realistic sensing model and differential constraints. *In:* IEEE. ROBOTICS and Automation (ICRA), 2016 IEEE International Conference on. [*S.l.*: *s.n.*], 2016. p. 2782–2787.

KARAMAN, Sertac; FRAZZOLI, Emilio. Sampling-based algorithms for optimal motion planning. **The international journal of robotics research**, Sage Publications Sage UK: London, England, v. 30, n. 7, p. 846–894, 2011.

KAVRAKI, Lydia *et al.* Probabilistic roadmaps for path planning in high-dimensional configuration spaces. **IEEE Transactions on Robotics and Automation**, IEEE, v. 12, n. 4, p. 566–580, 1996.

KAYRAKI, L *et al.* Probabilistic roadmaps for path planning in high-dimensional configurations space. **Proc IEEE Trans Robot Autom**, v. 12, n. 4, p. 566–80, 1996.

KHATIB, Oussama. Real-time obstacle avoidance for manipulators and mobile robots. *In:* AUTONOMOUS robot vehicles. [*S.l.*]: Springer, 1986. p. 396–404.

KOENIG, Nathan; HOWARD, Andrew. Design and use paradigms for gazebo, an open-source multi-robot simulator. *In:* IEEE. INTELLIGENT Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. [*S.l.*: *s.n.*]. p. 2149–2154.

KUFFNER, James J; LAVALLE, Steven M. RRT-connect: An efficient approach to single-query path planning. *In:* IEEE. ROBOTICS and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on. [*S.l.*: *s.n.*], 2000. p. 995–1001.

LAND, Stephanie; CHOSET, Howie. Coverage path planning for landmine location. *In:* THIRD International Symposium on Technology and the Mine Problem. [*S.l.*: *s.n.*], 1998.

LATOMBE, Jean-Claude. **Robot motion planning**. [*S.l.*]: Springer Science & Business Media, 2012. v. 124.

LAVALLE, Steven M. **Planning algorithms**. [*S.l.*]: Cambridge university press, 2006.

LAVALLE, Steven M. Rapidly-exploring random trees: A new tool for path planning. Citeseer, 1998.

LAVALLE, Steven M; KUFFNER JR, James J. Randomized kinodynamic planning. **The International Journal of Robotics Research**, SAGE Publications, v. 20, n. 5, p. 378–400, 2001.

LAWLER, Eugene L *et al.* **The traveling salesman problem: a guided tour of combinatorial optimization**. [*S.l.*]: Wiley New York, 1985. v. 3.

LEE, D; LIN, Arthurk. Computational complexity of art gallery problems. **IEEE Transactions on Information Theory**, IEEE, v. 32, n. 2, p. 276–282, 1986.

LIM, Ronny Salim; LA, Hung Manh; SHENG, Weihua. A robotic crack inspection and mapping system for bridge deck maintenance. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 11, n. 2, p. 367–378, 2014.

LIN, Shen; KERNIGHAN, Brian W. An effective heuristic algorithm for the traveling-salesman problem. **Operations research**, INFORMS, v. 21, n. 2, p. 498–516, 1973.

LOZANO-PEREZ, Tomas. A simple motion-planning algorithm for general robot manipulators. **IEEE Journal on Robotics and Automation**, IEEE, v. 3, n. 3, p. 224–238, 1987.

LOZANO-PEREZ, Tomas. Spatial planning: A configuration space approach. **IEEE transactions on computers**, IEEE, n. 2, p. 108–120, 1983.

MAI, Christian *et al.* Subsea infrastructure inspection: A review study. *In:* IEEE. UNDERWATER System Technology: Theory and Applications (USYS), IEEE International Conference on. [*S.l.*: *s.n.*], 2016. p. 71–76.

MASSIOS, Nikolaos A; FISHER, Robert B *et al.* **A best next view selection algorithm incorporating a quality criterion**. [*S.l.*]: Department of Artificial Intelligence, University of Edinburgh, 1998.

MAVER, Jasna; BAJCSY, Ruzena. Occlusions as a guide for planning the next view. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 15, n. 5, p. 417–433, 1993.

MORAVEC, Hans; ELFES, Alberto. High resolution maps from wide angle sonar. *In:* IEEE. PROCEEDINGS. 1985 IEEE international conference on robotics and automation. [*S.l.*: *s.n.*], 1985. p. 116–121.

NILSSON, Nils J. **A mobile automaton: An application of artificial intelligence techniques**. [*S.l.*], 1969.

O'ROURKE, Joseph. **Art gallery theorems and algorithms**. [*S.l.*]: Oxford University Press Oxford, 1987. v. 57.

OLIVIERI, P *et al.* Coverage path planning for eddy current inspection on complex aeronautical parts. **Robotics and Computer-Integrated Manufacturing**, Elsevier, v. 30, n. 3, p. 305–314, 2014.

OLLIS, Mark; STENTZ, Anthony. First results in vision-based crop line tracking. *In:* IEEE. ROBOTICS and Automation, 1996. Proceedings., 1996 IEEE International Conference on. [*S.l.*: *s.n.*], 1996. p. 951–956.

OSOSINSKI, Marek; LABROSSE, Frédéric. Multi-viewpoint visibility coverage estimation for 3D environment perception volumetric representation as a gateway to high resolution data. *In:* IEEE. COMPUTER Vision Theory and Applications (VISAPP), 2014 International Conference on. [*S.l.*: *s.n.*], 2014. p. 462–469.

PALOMERAS, Narcıs *et al.* Autonomous Mapping of Underwater 3-D Structures: From View Planning To Execution. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 3, p. 1965–1971, 2018.

PAPADOPOULOS-ORFANOS, Dimitri; SCHMITT, Francis. Automatic 3-D digitization using a laser rangefinder with a small field of view. *In:* IEEE. 3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in. [*S.l.*: *s.n.*], 1997. p. 60–67.

PAPADOPOULOS, Georgios; KURNIAWATI, Hanna; PATRIKALAKIS, Nicholas M. Asymptotically optimal inspection planning using systems with differential constraints. *In:* IEEE. ROBOTICS and Automation (ICRA), 2013 IEEE International Conference on. [*S.l.*: *s.n.*], 2013. p. 4126–4133.

PARNUM, IM *et al.* The effect of incident angle on statistical variation of backscatter measured using a high-frequency multibeam sonar. *In:* PROCEEDINGS of the Acoustics 2004 Conference of the Australian Acoustical Society. [*S.l.*: *s.n.*], 2005.

PRIETO, Flavio *et al.* A CAD-based 3D data acquisition strategy for inspection. **Machine Vision and Applications**, Springer, v. 15, n. 2, p. 76–91, 2003.

PRIM, Robert Clay. Shortest connection networks and some generalizations. **Bell system technical journal**, Wiley Online Library, v. 36, n. 6, p. 1389–1401, 1957.

PUDNEY, Christopher John. **Surface modelling and surface following for robots equipped with range sensors**. 1994. Tese (Doutorado) – University of Western Australia.

QUIN, Phillip *et al.* Efficient neighbourhood-based information gain approach for exploration of complex 3D environments. *In:* IEEE. ROBOTICS and Automation (ICRA), 2013 IEEE International Conference on. [*S.l.*: *s.n.*], 2013. p. 1343–1348.

REED, Michael K; ALLEN, Peter K. Constraint-based sensor planning for scene modeling. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 22, n. 12, p. 1460–1467, 2000.

REINELT, Gerhard. **The traveling salesman: computational solutions for TSP applications**. [*S.l.*]: Springer-Verlag, 1994.

RIBAS, David *et al.* Girona 500 auv: From survey to intervention. **IEEE/ASME Transactions on Mechatronics**, IEEE, v. 17, n. 1, p. 46–53, 2012.

ROBERTS, Mike *et al.* Submodular trajectory optimization for aerial 3d scanning. *In:* INTERNATIONAL Conference on Computer Vision. [*S.l.*: *s.n.*], 2017.

ROSENKRANTZ, Daniel J; STEARNS, Richard Edwin; LEWIS, Philip M. Approximate algorithms for the traveling salesperson problem. *In:* IEEE. SWITCHING and Automata Theory, 1974., IEEE Conference Record of 15th Annual Symposium on. [*S.l.*: *s.n.*], 1974. p. 33–42.

SAFAK, Görkem. **The art-gallery problem: A survey and an extension**. [*S.l.*]: Skolan för datavetenskap och kommunikation, Kungliga Tekniska högskolan, 2009.

SÁNCHEZ, Gildardo; LATOMBE, Jean-Claude. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. *In:* ROBOTICS Research. [*S.l.*]: Springer, 2003. p. 403–417.

SCHMID, Korbinian *et al.* View planning for multi-view stereo 3d reconstruction using an autonomous multicopter. **Journal of Intelligent & Robotic Systems**, Springer, v. 65, n. 1, p. 309–323, 2012.

SCIAVICCO, Lorenzo; SICILIANO, Bruno. **Modelling and control of robot manipulators**. [*S.l.*]: Springer Science & Business Media, 2012.

SCOTT, William R. Model-based view planning. **Machine Vision and Applications**, Springer, v. 20, n. 1, p. 47–69, 2009.

SCOTT, William R; ROTH, Gerhard; RIVEST, Jean-François. View planning for automated three-dimensional object reconstruction and inspection. **ACM Computing Surveys (CSUR)**, ACM, v. 35, n. 1, p. 64–96, 2003.

SHIMADA, Kenji; GOSSARD, David C. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. *In:* ACM. PROCEEDINGS of the third ACM symposium on Solid modeling and applications. [*S.l.*: *s.n.*], 1995. p. 409–419.

SI, Hang; TETGEN, A. A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. **Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany**, p. 81, 2006.

SICILIANO, Bruno; KHATIB, Oussama. **Springer handbook of robotics**. [*S.l.*]: Springer, 2008.

SOUZA, Luiz Antonio Pereira de. **Revisão crıtica da aplicabilidade dos métodos geofısicos na investigação de áreas submersas rasas**. 2006. Tese (Doutorado) – Universidade de São Paulo.

STENTZ, Anthony. Optimal and efficient path planning for partially-known environments. *In:* ICRA. [*S.l.*: *s.n.*], 1994. p. 3310–3317.

SUCAN, Ioan A; MOLL, Mark; KAVRAKI, Lydia E. The open motion planning library. **IEEE Robotics & Automation Magazine**, IEEE, v. 19, n. 4, p. 72–82, 2012.

TARABANIS, Konstantinos A; ALLEN, Peter K; TSAI, Roger Y. A survey of sensor planning in computer vision. **IEEE transactions on Robotics and Automation**, IEEE, v. 11, n. 1, p. 86–104, 1995.

TARBOX, Glenn H; GOTTSCHLICH, Susan N. Planning for complete sensor coverage in inspection. **Computer Vision and Image Understanding**, Elsevier, v. 61, n. 1, p. 84–111, 1995.

TARJAN, Robert Endre; VAN WYK, Christopher J. A linear-time algorithm for triangulating simple polygons. *In:* ACM. PROCEEDINGS of the eighteenth annual ACM symposium on Theory of computing. [*S.l.*: *s.n.*], 1986. p. 380–388.

TRUCCO, Emanuele *et al.* Model-based planning of optimal sensor placements for inspection. **IEEE Transactions on Robotics and Automation**, IEEE, v. 13, n. 2, p. 182–194, 1997.

VAN DEN BERG, Jur; ABBEEL, Pieter; GOLDBERG, Ken. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. **The International Journal of Robotics Research**, SAGE Publications Sage UK: London, England, v. 30, n. 7, p. 895–913, 2011.

VASQUEZ-GOMEZ, J Irving; SUCAR, L Enrique; MURRIETA-CID, Rafael. Hierarchical ray tracing for fast volumetric next-best-view planning. *In:* IEEE. COMPUTER and Robot Vision (CRV), 2013 International Conference on. [*S.l.*: *s.n.*], 2013. p. 181–187.

VASQUEZ-GOMEZ, J Irving; SUCAR, L Enrique; MURRIETA-CID, Rafael. View/state planning for three-dimensional object reconstruction under uncertainty. **Autonomous Robots**, Springer, v. 41, n. 1, p. 89–109, 2017.

VASQUEZ-GOMEZ, J Irving; SUCAR, L Enrique; MURRIETA-CID, Rafael; LOPEZ-DAMIAN, Efrain. Volumetric next-best-view planning for 3d object reconstruction with positioning error. **International Journal of Advanced Robotic Systems**, SAGE Publications Sage UK: London, England, v. 11, n. 10, p. 159, 2014.

WALLAR, Alex; PLAKU, Erion; SOFGE, Donald A. A planner for autonomous risk-sensitive coverage (PARCov) by a team of unmanned aerial vehicles. *In:* IEEE. SWARM Intelligence (SIS), 2014 IEEE Symposium on. [*S.l.*: *s.n.*], 2014. p. 1–7.

WANG, Pengpeng; KRISHNAMURTI, Ramesh; GUPTA, Kamal. Generalized watchman route problem with discrete view cost. **International Journal of Computational Geometry & Applications**, World Scientific, v. 20, n. 02, p. 119–146, 2010.

WHAITE, Peter; FERRIE, Frank P. Autonomous exploration: Driven by uncertainty. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 19, n. 3, p. 193–205, 1997.

WHITFIELD, Stephen. **Supervised Autonomy Bridges the Gap in ROV Inspections**. 2018. Disponível em: `https://www.spe.org/`. Acesso em: 4 fev. 2019.

YASUTOMI, Fumio; YAMADA, Makoto; TSUKAMOTO, Kazuyoshi. Cleaning robot control. *In:* IEEE. ROBOTICS and Automation, 1988. Proceedings., 1988 IEEE International Conference on. [*S.l.*: *s.n.*], 1988. p. 1839–1841.

YOUAKIM, Dina; RIDAO, Pere. Motion planning survey for autonomous mobile manipulators underwater manipulator case study. **Robotics and Autonomous Systems**, Elsevier, v. 107, p. 20–44, 2018.

ZANONI, Fábio Doro; BARROS, Ettore Apolônio de. A real-time navigation system for autonomous underwater vehicle. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, Springer, v. 37, n. 4, p. 1111–1127, 2015.

ZELINSKY, Alexander *et al.* Planning paths of complete coverage of an unstructured environment by a mobile robot. *In:* PROCEEDINGS of international conference on advanced robotics. [*S.l.*: *s.n.*], 1993. p. 533–538.