TÉCNICO LISBOA

Optimization of Landmark Placement for Robot Navigation

Délio Fabrício dos Reis Bengaló

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor(s): Prof. Alberto Manuel Martinho Vale
Prof. Rodrigo Martins de Matos Ventura

Examination Committee

Chairperson: Prof. Paulo Jorge Coelho Ramalho Oliveira
Supervisor: Prof. Rodrigo Martins de Matos Ventura
Member of the Committee: Prof. João Manuel de Freitas Xavier

May 2021

Acknowledgments

Em primeiro lugar, gostaria de agradecer ao Professor Alberto Vale e ao Professor Rodrigo Ventura, cuja experiência foi inestimável para a realização desse projecto. Obrigado pela disponibilidade e por fornecerem as ferramentas de que eu precisava para concluir com sucesso a minha dissertação.

Aos meus pais quero agradecer pelo apoio e o amor que sempre me deram, levarei sempre comigo os valores que me transmitiram e que me fizeram chegar até aqui.

Finalmente, aos meus amigos que estão comigo desde o início, obrigado pela amizade e por todos os momentos que passamos ao longo desses anos.

Em especial agradeço a Alexandra pela paciência e por acreditar em mim. Obrigado por estar sempre presente.

Resumo

O uso de marcadores como pontos de referência têm sido cada vez mais frequente em diversas aplicações incluindo sistemas de localização em espaços fechados, robótica e realidade aumentada. De um modo geral, a localização dos marcadores é crucial para um desempenho eficiente e confiável da aplicação em questão, como por exemplo, no caso de localização em espaços fechados, onde o erro de posicionamento aumenta de forma significativa devido a obstáculos e oclusões, frequentes nesse tipo de ambiente. Também, na ausência de um método de localização absoluta, as estimativas de posição estão sujeitas a erros provenientes do escorregamento das rodas no caso de veículos terrestres e aos erros acumulativos típicos dos sensores inerciais usados em drones.

Esta tese apresenta um método de optimização para colocação de marcadores em um determinado ambiente, que é representado no plano como um polígono, para fins de navegação de robots terrestres, aéreos ou mesmo subaquáticos munidos de sensores não omnidirecionais. Além disso, o método desenvolvido leva em consideração as dimensões dos marcadores. Em primeiro lugar, é resolvido um problema relaxado assumindo marcadores pontuais e evoluindo para uma abordagem realista considerando pontos de referência não pontuais. O método implementado consiste em duas fases: primeiro, um conjunto de intervalos candidatos a receber um marcador, são calculados por algoritmos de complexidade polinomial; segundo, a colocação dos marcadores é formulado como um problema de Programação Linear Inteira e por intermédio de um solver padrão, obtém-se uma solução globalmente óptima.

Quando restringido ao mesmo período de tempo, o método desenvolvido obtém melhor cobertura do que um conjunto de meta-heurísticas considerados. Uma abordagem de optimização multi-critério permitiu estudar um possível *trade-off* entre a cobertura e o número de marcadores instalados.

Palavras-chave: Marcadores, Problema de Optimização, Programação Linear Inteira, Localização.

Abstract

Landmarks have an important role in a wide range of applications in different fields including indoor positioning, mobile robotics, and augmented reality. In the general case, the localization of the landmarks is crucial for efficient and reliable performance of the application, for instance, in the case of indoor positioning systems where the accuracy of the position estimation drops significantly in occlusion situations. In addition, the position estimation is affected by wheel slippage for ground vehicles and accumulative errors typical from inertial sensors used in aerial vehicles.

This thesis presents an optimization method for landmarks placement in a polygonal environment, for ground, underwater, and aerial autonomous vehicles featured with non-omnidirectional sensors. In addition, the developed method considers the dimensions of the landmarks. Firstly, a relaxed problem is solved by assuming pointwise landmarks and evolving to a realistic approach considering non-pointwise landmarks. The implemented method consists of two phases: first, a set of candidate intervals in which a landmark can be placed, are computed by polynomial-time algorithms, and second, the landmarks placement problem is formulated as an *Integer Linear Programming* (ILP) and a globally optimal solution is obtained through a standard ILP solver.

The method satisfies theoretical upper bounds established by *Art Gallery Theorems*. When constrained to the same period of time our method obtains better coverage than a set of meta-heuristic algorithms considered. A *Multi-criteria optimization problem* formulation allowed us to study a possible trade-off between the coverage and the number of landmarks.

Keywords: Landmarks, Mobile robotics, Localization, optimization problem.

Contents

List of Tables

List of Figures

Nomenclature

Abbreviations

AAL Ambient Assisted Living

AGP Art Gallery Problem

CCW Counter Clockwise

CG Computational Geometry

CW Clockwise

GPS Global Positioning System

ILP Integer Linear Programming

IMU Inertial Measurement Unit

IoT Internet of Things

IPFN Institute for Plasma and Nuclear Fusion

IPS Indoor Positioning Systems

ISR Institute of Systems and Robotics

ITER International Thermonuclear Experimental Reactor

MIP Mixed Integer Programming

RFID Radio Frequency Identification

RSSI Radio Signal Strength Intensity

SA Simulated Annealing

SCP Set Cover Problem

ToF Time-of-Flight

UAV Unmanned Aerial Vehicle

UWB Ultra Wide-Band

WLAN Wireless Local Area Networks

LPP Landmarks Placement Problem

Subscripts

i, j, k Computational indexes.

Operators

$\|\cdot\|$ Length of a line segment.

$|\cdot|$ Cardinality of set .

∂ Boundary of a polygon.

$ext(\cdot)$ Exterior of a polygon.

$int(\cdot)$ Interior of a polygon.

$V(\cdot)$ Point-visibility polygon.

$v(\cdot)$ Limited-visibility polygon, i.e. the intersection of the point visibility polygon and sensor detection area.

Sets

$\mathcal{V}(\cdot)$ Vertices of a polygon.

$\mathcal{W}$ Witness set.

$\mathcal{W}_j$ Set of witness poses that can identify the landmark l_j.

$E(\cdot)$ Edges of a polygon.

$H(\cdot)$ Holes (obstacles).

Variables

$\mathbf{M}$ Number of poses.

$\mathbf{N}$ Number of candidate intervals and potential landmarks.

ΔC_{ij} Cross-coverage error.

$\mathcal{A}$ Set of edges of the constraint graph of mutually-exclusive candidate intervals.

$\mathcal{G}$ Constraints graph build from the mutually-exclusive candidate intervals.

C Coverage.

C_{ij} Cross-coverage.

D_l Length of the landmarks.

d_l	Normalized length of the landmarks.
e	Edge.
G	Incremental Gain.
h	Number of obstacles (holes) of a polygon.
L	Maximum number of landmarks allowed.
l	Landmark.
n	Number of vertices and edges.
P	Polygon.
p	Arbitrary pose.
R	Sensor detection Range.
s	Sensor Detection Area.
v	Vertex.
w	Witness pose.

Chapter 1

Introduction

1.1 Motivation

The advances in electrical, mechanical and computer engineering and the rise of artificial intelligence in recent years have fueled the growth of *autonomous systems*. Robots, networks of sensors and actuators, autonomous vehicles and other devices have been used in several civilian and military applications, performing monotone, difficult, or even dangerous tasks to human beings in applications such as environment monitoring, mapping, survey, surveillance and rescue. The idea that motivates this present thesis is a real example of this trend, proposed by the Institute for Plasma and Nuclear Fusion (IPFN), which consists of using Unmanned Aerial Vehicles (UAV) inside nuclear facilities, intending to perform inspection operations to minimize hazardous exposures to radiation. Since 2008, IPFN in a collaboration with the Institute of Systems and Robotics (ISR) has been participating in Remote Handling activities of the International Thermonuclear Experimental Reactor (ITER) that aims to use fusion as a large-scale and carbon-free energy source. In a thermonuclear reactor, tasks including inspections and maintenance operations can become potentially dangerous exposures to radiation when performed by humans.

Aerial inspection is a recent demand application aiming to decrease human risk and reduce operational costs and the time of inspections [1]. There are several research projects in which UAVs perform inspections tasks in a wide range of terrains and situations such as bridges [2], power plant boilers [3], wind turbines [4] and tunnels.

Generally, navigation and/or localization are crucial aspects for autonomous systems with an essential role in system control. An efficient and reliable navigation and/or localization method increases the operational capabilities, reduces the risks and costs of the autonomous system. The usual practice in mobile robotics is to combine measurements obtained through satellite-based signals e.g. the Global Positioning System (GPS) with *inertial sensor-based methods* complemented with traditional filtering techniques to estimate the position and orientation. Inertial sensor-based are Relative (local) Localization methods that use data obtained from a set of sensors such as accelerometers, gyroscopes and magnetometers, also known as Inertial Measurement Unit (IMU). For instance, accelerometers and gy-

roscopes are used to measure linear acceleration and angular rate, respectively. The measurements are integrated once for the gyroscope and twice for the accelerometers to yield position and orientation, the integration starts from a known position and the estimations are continuously updated in time [5].

Inertial navigation systems are self-contained since they do not need external references, which makes them immune to occlusions, shadowing, or signal outage [5, 6]. However, when used over longer periods the sensor readings are impacted by drift and offsets, due to the integration, any small constant error increases without bound. Hence, the importance of merging with an Absolute (global) Localization method like GPS which does not require noisy data integration. On the other hand, for some indoor applications, GPS is not a reliable method, since GPS signals are much more attenuated as they propagate through buildings; multipath interference is more intense and disruptive regarding the signal acquisition indoors [7].

Therefore, solutions for the indoor localization problem have been provided, including radars, laser sensors, ultrasonic sonars, infrared sensors, radio frequency identification (RFID) devices and vision-based systems. Also technologies like WiFi, Ultra-Wide-Band (UWB), Wireless Local Area Networks (WLANs) and Bluetooth systems, which play an important role in the emerging fields of Indoor Positioning Systems (IPS), Ambient Assisted Living (AAL) and Internet-of-Things, among others [8]. Surveys of indoor localization systems and technologies for these emerging applications are presented in [8, 9].

However, despite the significant differences between the referenced solutions, a common feature in most of them is the need to deploy appropriate reference devices in the environment where possible targets are supposed to be localized and tracked. Usually, such devices (sometimes referred to as “anchor nodes”, “tags”, “markers” or “landmarks”, depending on whether they are active or passive and on the kind of sensing technology adopted) have known coordinates and/or orientation in a given reference frame [10]. For practical reasons, it is not convenient to consider an arbitrarily large number of artificial landmarks which in addition to the cost would require significant computational power. Hence, one question that immediately arises is: **Where to place the landmarks to have continuous position and orientation estimation?** The focus of this thesis is the subsequent *Optimization Problem*, which consists in determining the optimal placement of a set of landmarks for a given environment.

This thesis proposes a novel method to optimize the landmarks placement, in a polygonal shape environment, for navigation purposes of an agent (a robot, unmanned vehicle, or even a person) fitted with a detection system with limited field-of-view and finite range. This thesis builds upon the work of Cordeiro [11]. In his work, Cordeiro used a Monte Carlo algorithm to obtain the landmarks placement, which does not guarantee optimality.

The implemented method is based on a set of witness poses that are randomly generated or obtained from predefined trajectories and it involves two major steps:

- Pre-Processing phase where a visibility algorithm computes a set of potential locations (intervals along the edges of the polygon) for the landmarks;
- Optimization phase where the pose coverage problem is formulated as an ILP and finally, a standard ILP solver is used to obtain the optimal solution.

Although the approach of this thesis was originally developed bearing in mind a robot navigation application for both indoor and outdoor environments, it is also suitable for fields such as the advertising industry. The location of the information (in form of *posters*, *banners*, or *roll-ups*) is crucial to reach as many people as possible. In such applications, it may be interesting to consider the size of the "landmarks" (the device that supports the information), instead of a pointwise approach that is common in landmarks placement solutions, since in the advertising industry, the size of a poster or a banner is often not neglectable when compared with the walls of a building. Thus, due to this interesting application and the fact that real-world landmarks are having a certain dimension, it is proposed a method able to deal with non-pointwise landmarks.

1.2 Objectives

This thesis aims to find a solution to the problem of determining the optimal landmarks placement for a given closed environment modeled as a polygon, to maximize the coverage of the given environment. The implemented method relies on a simulation of a set of candidate poses of certain agents with limited field-of-view. The coverage is defined as the percentage of the total number of agent poses where the agent detection system identifies at least one landmark.

One of the main objectives is to develop a method that guarantees optimality, which remains an open issue in previous works [11].

Another objective is to study a possible trade-off between the required coverage and the number of landmarks (or the cost of landmark installation). To address this trade-off, this thesis formulates the landmarks placement problem as a *Multi-Criteria Optimization Problem*, where the coverage and the cost (number of landmarks) are two conflicting objective functions to be maximized and minimized respectively.

1.3 Contribution

The main contribution of this thesis is the development of a global optimization method for the landmarks placement problem. The method is implemented in two phases: a pre-processing phase and an optimization phase. For the pre-processing phase, two visibility algorithms are developed, for both pointwise and non-pointwise landmarks. The pre-processing algorithms compute in polynomial time the intersections of the visibility polygons of a set of poses to generate candidate intervals where a landmark can be placed. In the optimization phase, the landmarks placement problem is formulated as an ILP and it is solved with a state-of-the-art ILP solver which guarantees optimality on its own.

1.4 Thesis Outline

This thesis is organized as follows: In Chapter 2 are presented the important theoretical background and concepts that may be unfamiliar to the reader, allowing a sustained and comprehensible unfolding

of this project. It is also presented a survey of the state-of-the-art solutions for the Landmark Placement Problem (LPP).

In Chapter 3 it is presented an ILP formulation of the LPP as well as the key elements to the formulation of the problem including the environment, the detection system of the agents (the sensors), and the landmarks.

Chapter 4 presents a discussion of the chosen methodology regarding the advantages and disadvantages of the method implemented. In addition, it is given a detailed explanation of the pre-processing algorithms.

Chapter 5 shows the most important results obtained from the simulation of the implemented algorithm in different environments. The results of a comparison with several meta-heuristic algorithms are presented to evaluate the performance of the method.

Finally, in Chapter 6 are presented the conclusions of the results achieved in this thesis as well as the suggestions and possible improvements for future works.

Chapter 2

Background

This chapter provides general insight on the fundamental concepts for the comprehension of this thesis including the *Art Gallery Problem* (AGP) a well-known approach to tackle sensor and landmarks placement problems, point visibility concept, and *Visibility algorithms* to compute the region that is visible to a given point. Finally, a literature review is presented aiming to highlight the most common and newest solutions to similar problems.

2.1 Definitions

Point

In *Euclidean geometry*, a point is a primitive notion that represents an exact location in the space and has no length, width, or thickness. In *Point-Set Topology* a point is interpreted as an element of a set called *space*. In this thesis, a point is described by an ordered set of numbers (coordinates) so that relations between points are represented by relations between coordinates. For instance, in a two-dimensional *Euclidean space* a point is represented in Cartesian coordinates by an ordered pair (x, y) of numbers.

Line

In Euclidean geometry, a line is an object with no width, thickness, and with an infinite length, uniquely identified by two points. Regarding a two-dimensional Euclidean space, Analytical geometry describes a line as the set of points whose coordinates satisfy a given linear equation:

$$L = \{(x, y)|ax + by = c\} \tag{2.1}$$

here a, b and c are constant real numbers.

Line segment

A line segment is a finite portion of a line bounded by two distinct points, called endpoints. A line segment with endpoints A and B is denoted by $\overline{AB}$. And the length of the line segment, which is the shortest

distance between the endpoints, is denoted by $\|\overline{AB}\|$. A line segment $\overline{AB}$ contains precisely those points of the line through A and B that lie between A and B; the endpoints themselves are included in the line segment.

Note: Often in this thesis, the term interval is used to refer to a line segment mainly a subset of a given line segment.

Connectedness

To understand the definition of the following mathematical objects, it is important to introduce the concepts of *connectedness* and *path-connectedness*. Intuitively a connected space does not fall apart into two or more pieces [12].

Definition 2.1.1. A topological space S is connected if it does not admit partition into two non-empty open subsets A and B such that $S = A \cup B$ and $A \cap B = \emptyset$.

To introduce the notion of path-connectedness, one must first define the concept of *path*.

Definition 2.1.2. Given two points x and y in a topological space S, a path in S from x to y is a continuous map $f : [0, 1] \to X$ such that $f(0) = x$ and $f(1) = y$. Thus, such a path is said to join x and y.

Definition 2.1.3. A topological space S is path-connected if there is a path joining any to points x and y in S.

Intuitively, a path-connected space is *simply connected* if it consists of one piece and it does not have any "holes". In other words, S is simply connected if any closed curve in S can be shrunk continuously into a point within S. For the following definition consider that $\mathcal{C}$ and $\mathcal{D}$ denote the unit circle and the closed unit disk in the Euclidean plane respectively.

Definition 2.1.4. A topological path-connected space S is *simply-connected* if for any closed curve in S defined by $f : \mathcal{C} \to S$ exists a continuous map $F : \mathcal{D} \to S$ such that F restricted to $\mathcal{C}$ is f.

If a topological space S is path-connected but not simply connected, then S is said to be *multiply-connected* or *h-connected* where $h \geq 1$ is the number of holes in S.

Polygon

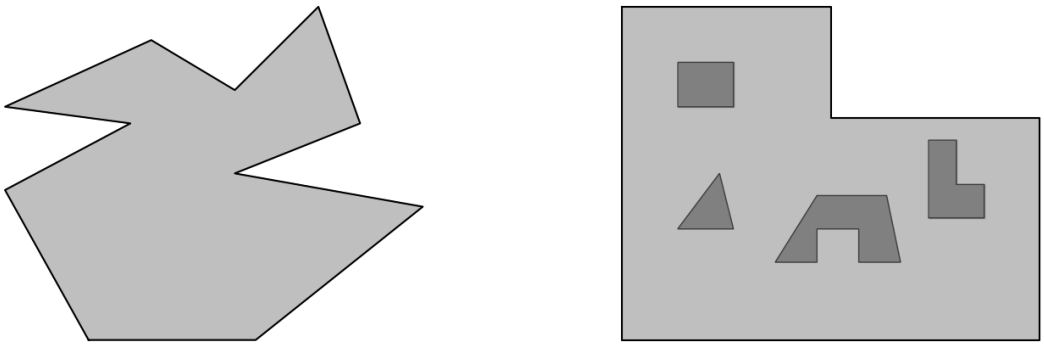
A polygon P is a closed and path-connected space in a two-dimensional Euclidean space, bounded by a finite set of line segments, called *edges*. The endpoints of an edge of P, are called *vertices*. The boundary of P consists of *cycles* of edges and it is denoted by ∂P. Two consecutive edges in a cycle share a vertex [13, 14].

A polygon is classified as:

- *Simple polygon* if it is a simply connected space. Thus, the boundary of the polygon consists of only one cycle of edges;

- *Polygon with holes* or with obstacles, if it is multiply-connected. Thus, the boundary consists of two or more cycles of edges.

Note that in some literature sources the term "*simple polygon*" refers to polygons that do not cross themselves, and "*self-intersecting polygons*", denotes the polygons in which two nonconsecutive edges intersect. However, this thesis does not consider self-intersecting polygons. The term "simple polygon" will refer only to a polygon without obstacles. Figure 2.1 illustrates both a simple and a polygon with holes.



(a) (b)

Figure 2.1: (a) A simple polygon and (b) a polygon with holes.

Definition 2.1.5. Consider R^2, a two-dimensional Euclidean space. A polygon $P \subset R^2$ is said to be a **simple polygon** if its boundary is a *Jordan Curve* i.e. if there are exactly two disjoint path-connected subsets of $R^2 \setminus \partial P$, which are the *interior* and *exterior* of P respectively. Thus, P is simply connected.

Definition 2.1.6. A polygon $P \subset R^2$ is a **polygon with holes** if P is multiply-connected.

A polygon P consists in the union of its boundary and its interior here defined as $int(P)$, the complementary region of P in the plane is called exterior and it is denoted as $ext(P)$.

$$P = \partial P \cup int(P) \tag{2.2}$$

Let P be a polygon with holes. The set of holes in P is denoted by $H(P) = \{P_1, \ldots, P_h\}$ where h is the number of holes in P. P_k with $1 \leq k \leq h$ is a simple polygon enclosed by an outer polygon P_0, commonly referred to as *shell* of P. And P has the following proprieties:

- $P_k \subset P_0, 1 \leq k \leq h,$
- $\partial P_k \cap \partial P_j = \emptyset, \forall_{k,j} : k \neq j$
- $int(P) = int(P_0) \setminus [P_1 \cup \ldots \cup P_h]$

The set of vertices of an arbitrary polygon P is denoted by

$$\mathcal{V}(P) = \{v_1, \ldots, v_n\}$$

and the set of edges will be denoted by

$$E(P) = \{e_1, \ldots, e_n\}$$

where n is the number of vertices and edges of P, if P is a simple polygon (the boundary consists of only one cycle of edges) then an edge $e_i = \overline{v_i v_{i+1}}$ with $i = 1, \ldots, n$ and $v_{n+1} = v_1$. In the case that P is a polygon with holes, then $e_i = \overline{v_i v_{i+1}}$ only if v_i and v_{i+1} belong to the same cycle of edges. Two vertices v_i and v_{i+1} belong to the same cycle of edges if the line segment $e = \overline{v_i v_{i+1}}$ connecting the two vertices is an edge of P:

$$\exists k \in \{0, \ldots, h\} : e \subset \partial P_k$$

Figure 2.2 shows the representation of a polygon with 10 vertices and 1 hole.

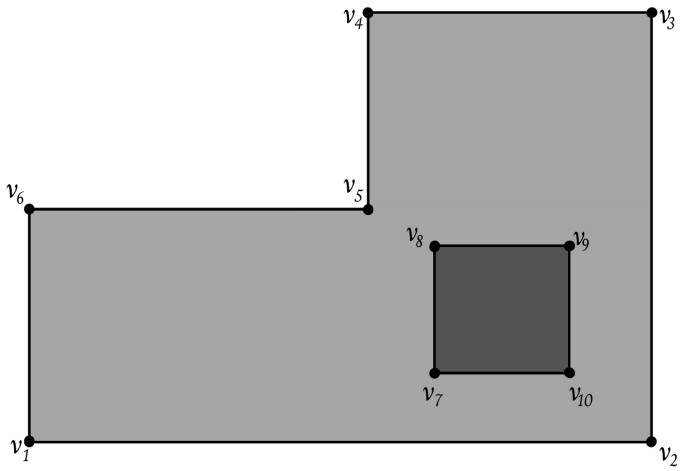


Figure 2.2: Representation of a polygon with one hole, where the list of edges $E = \{\{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_6), (v_6, v_1)\}, \{(v_7, v_8), (v_8, v_9), (v_9, v_{10}), (v_{10}, v_7)\}\}$ is a set of two cycles of edges, one for the outer and other the inner (hole) boundary.

2.1.1 Point Visibility

The notion of visibility is present in everyday tasks, seeing an object means identifying the portions of the objects that are visible to the current position of the observer and determining shapes and sizes of the visible portions [13]. Determining the portion of a geometric object that is visible from a given source is a well-studied and the most fundamental problem in Computational Geometry (CG) [14]. However, this thesis considers visibility computations only in polygons.

Definition 2.1.7. Given a polygon P and a point $x \in P$, x *sees* or *covers* a point $y \in P$ if the line segment $\overline{xy}$ that connects both points lies inside the polygon: $\overline{xy} \subseteq P$. The points x and y are both said

to be visible from each other [15, 16].

Figure 2.3 illustrates different situations of point-visibility. The portion of P that is visible from x is

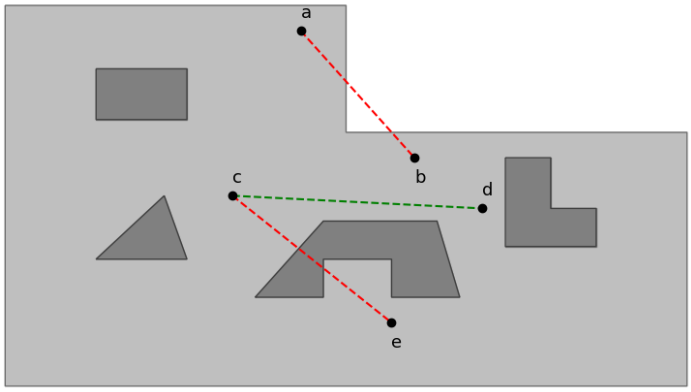


Figure 2.3: Concept of point-visibility in a polygon. The points c and d are mutually visible as there is a direct line-of-sight, the point e is not visible from c and vice versa because an obstacle interrupts the direct line-of-sight. Also, a is not visible from b because the line segment $\overline{ab}$ does not lies inside the polygon.

called the *point-Visibility Polygon* of x in P and it is denoted by $V(x)$. The point-visibility polygon can be defined as $V(x) = \{p \in P : x \text{ sees } p\}$.

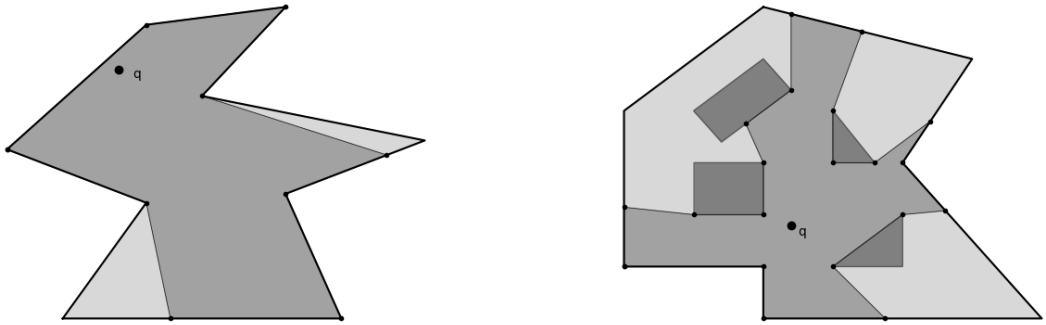
Definition 2.1.8. A polygon P' is said to be *star-shaped* if exists a point $x \in P'$ such that all points in P' are visible to x. The set of all such points x of P' is called *kernel* of P'.

By construction $V(x)$ is a star-shaped polygon [13]. And x belongs to the kernel of $V(x)$. The Kernel of a star-shaped polygon is always convex and it results from the intersection of all interior half-planes determined by edges of the polygon. In figure 2.4 is presented the visibility polygon of a pose q in a *simple polygon* (a) and in a *polygon with holes*. There are several visibility algorithms to compute the point-visibility polygon [17–20].

2.2 Point Visibility Algorithm

In this thesis, it is used the Erdem and Sclaroff [21] algorithm to compute the point-visibility polygon. This algorithm performs a radial sweep over a range of $[0, 2\pi]$ around an interior point x of a polygon with holes, P, computing the visible line segments of the boundary of the polygon.

Regarding the work of Erdem and Sclaroff [21], a polygon is represented as a list of edges in Cartesian coordinates ordered in counterclockwise (CCW) and clockwise (CW) for the outer boundary (∂P_0) and inner boundaries (the holes) respectively. Is important to remind the reader that P_0 refers to the shell of the polygon with holes P and ∂P_0 denotes the outer boundary of P.



(a) (b)

Figure 2.4: The visibility polygon of a point q (a) in a simple polygon and (b) in a polygon with holes.

$$ELC = \{(vc_1^s, vc_1^e), (vc_2^s, vc_2^e), \ldots, (vc_i^s, vc_i^e), \ldots, (vc_n^s, vc_n^e)\},$$

where vc_i^s and $vc_i^e \in \mathbb{R}^2$ are the start and end vertices of ith edge.

The first step is to convert ELC to its polar coordinates representation:

$$ELP = \{(vp_1^s, vp_1^e), (vp_2^s, vp_2^e), \ldots, (vp_i^s, vp_i^e), \ldots, (vp_n^s, vp_n^e)\}$$

where $vp_i = \{\theta_i, r_i\}$ denotes the angle and radius of the ith vertex, θ_i is measured between the horizontal half-line starting from x to the right. With this formulation only edges satisfying $\theta^s < \theta^e$ are fully or partially visible from q, those edges which do not satisfy this condition are eliminated in this step. This condition is evaluated by computing the cross product $\overrightarrow{u} \times \overrightarrow{e_i}$, between the edge $\overrightarrow{e_i}$ [1] and the vector $\overrightarrow{u}$ which is the vector connecting the point q with the start vertex of e_i, note both $\overrightarrow{u}$ and $\overrightarrow{e_i}$ are vectors of R^2. The cross product is positive only when $\theta^s < \theta^e$ [21]. To proceed with an angular scan of the polygon, the vertices in polar coordinates are sorted in increasing angular order, forming an ordered list of vertices. The algorithm scans the polygon counterclockwise evaluating each vertex in the ordered list, the visibility polygon is constructed by tracking the visible edge during the angular sweep.

Only the interior region of P_0 and the outside region of the holes may form the visibility polygon. As the vertices of P_0 are ordered CCW, the inside region is the intersection of the left half-planes defined by each edge. By ordering the vertices of the holes clockwise, the algorithm will work correctly, by detecting that the exterior of holes is the feasible region to build the $V(x)$.

2.3 Art Gallery Problem

The AGP is a classic problem in CG. The AGP and its variants have the goal of determining the number and the positions of a set of guard points that are sufficient to see an entire polygon, such that every

[1] Here $\overrightarrow{e_i}$ is the oriented line segment from the starting vertex to the end vertex of the edge e_i.

point in the polygon is seen by at least one of the guards. In the primal version of the AGP, the guards are static points that can survey 360º about their fixed positions and the art gallery room is a polygon P without holes and with n vertices. The goal is to determine the number of guards that suffice to oversee the inner walls of any polygon with n vertices.

Several variations of the problem have been intensively studied [14], resulting in an extensive collection of theorems and algorithms addressed to find the minimum number of guards as well as to compute their position. One of the most important theorems, known as the *Chavátal's art gallery theorem* established that $\lfloor \frac{n}{3} \rfloor$ guards are occasionally necessary and always sufficient to cover a polygon (without holes) of n vertices [14]. Since then, this theorem has an important role in CG and has been further developed by computer scientists in the scope of partitioning algorithms. Fisk [22] gave a concise sufficiency proof of this theorem. Fisk proof consists of: first, triangulate the polygon P (in such a way that any new vertex is added) and next *3-coloring* the vertices of the triangulation graph [16], so that adjacent vertices in the graph have always different colors; choosing one of the three colors classes it will result in a set of vertices that can see the entire polygon. The smallest set of the three classes is at most $\lfloor \frac{n}{3} \rfloor$. This elegant proof led the way to several similar results and algorithms, such as the proof presented in [23] for an *orthogonal polygon* [2], which consists in decomposing the polygon in convex quadrilaterals rather than triangles and assigning four different colors to adjacent vertices of the resulting graph, from which yields that $\lfloor \frac{n}{4} \rfloor$ guards are always sufficient and occasionally necessary to guard any orthogonal polygon with n vertices. Based on the same argument a $O(n \log n)$ *guard placement algorithm* is presented in [24].

An important variant of the AGP is one in which each guard is mobile and allowed to patrol a line segment $s \in P$. A point $y \in P$ is said to be visible to s if exists a point $x \in s$ such that y is visible to x. According to the degree of mobility, the guards can be classified as an *edge guard* when the path of the guard is restricted to an edge of P; a *diagonal guard* when s is an internal diagonal of P; and finally, a *line guard*, which is any line contained in P. The following table presents a summary of some important results in AGPs.

Polygon type	Stationary Guards	Mobile Guards
General Simple	$\lfloor \frac{n}{3} \rfloor$	$\lfloor \frac{n}{4} \rfloor$ O'Rourke [15]
Simple Orthogonal	$\lfloor \frac{n}{4} \rfloor$	$\lfloor (n+4)/16 \rfloor$
With Holes	$\lceil (n+h)/3 \rceil$	

Table 2.1: Theoretical bounds on the smallest number of guards necessary to cover a polygon with n vertices.

Several variants of the AGP have been proved to be NP-hard even for restricted and special cases [25–27]. Due to the complexity of these problems, researchers have been working in approximation

[2] An orthogonal polygon is one whose edges are all aligned with a pair of orthogonal coordinate axes, which are taken to be horizontal and vertical.

algorithms for some of the variants of the AGP rather than an exact solution.

Greedy strategies are natural and common approaches to address the AGP, by adding guards iteratively until coverage is achieved, choosing at each step a guard that maximizes the contribution to the coverage. Greedy approximation algorithms for the AGP explore the similarities with the Set Cover Problems (SCPs) and benefit from well-studied approximation algorithms for the SCP [28–30]. Greedy methods are common approaches in the set cover problems, yielding $O(\log n)$ approximation ratio. Approximated solutions for the minimum vertex and edge guard problems are presented in [31], these solutions can be computed in $O(n^4)$ for a polygon without holes and $O(n^5)$ for a polygon with holes. The algorithms in [31] partition the polygon into convex components and construct sets from these components. Then the Greedy heuristic approximation algorithm presented in [29] for the set cover problem is used to locate the minimum *guard set*. Although, the algorithm presented in [29] has an $O(\log n)$ approximation ratio, for the AGP it is an upper bound that may not be reached due to geometric constraints. Other approximation algorithms of the AGP including can be found in [32–35].

Recently, different approaches have been used to achieve an efficient approximation of the AGP including randomized, heuristics algorithms and computational geometry results and techniques. A randomized approximation algorithm is presented in [33] to locate the smallest set of vertex guards in a polygon P with n vertices. For a simple polygon, the algorithm runs in $\mathcal{O}(nc_{opt}^2 \log^4 n)$ expected time with an approximation ratio of $\mathcal{O}(\log c_{opt})$, where c_{opt} is the cardinality of the optimum set. And for a polygon with h holes, the expected running time is $\mathcal{O}(nc_{opt}^3 \operatorname{polylog} n)$. However, the randomized approximation algorithm does not guarantee solutions and the quality of the approximation is correct with high probability. A practical iterative primal-dual approach is presented in [36] to solve discrete instances of the AGP named *Art Gallery Problem with Witness* (AGPW). Where given a finite set of witness points $W \subset P$ and the goal is to determine the minimum set of guards that are sufficient to cover all points in W. The basic idea is to discretize the problem to formulate the AGP as a SCP. The algorithm successively computes tighter lower and upper bounds by refining the witness and guard sets during the iterations while seeking to reach an exact solution. The solutions of the primal and dual problems are used to refine the witness and the guard candidate sets to reduce the duality gap. The procedure stops when the lower and upper bounds coincide. Even though the algorithm obtains provably optimal solutions, finding theoretical proof of the convergence remains an open issue. Similar approaches can be found in [37, 38]. The main challenge of these approaches is not only to determine the ideal witness set but also to handle the huge SCP instance that ensues. It would be interesting to consider an integration of the visibility algorithms presented in this thesis with iterative primal-dual algorithms for the exact solution of the AGP as the ones presented in [36–38] that are also based on a witness set.

Based on an incremental algorithm, a new heuristic is presented in [39] to solve restricted instances of the AGP. A list of heuristic approximation algorithms based on greedy strategies is presented in [40] for the AGP.

Generally, these approximation algorithms of the AGP have a common principle, which is to reduce the AGP problem to an instance of the SCP [25, 31]. However, despite the relation between the two families of problems is hard to apply straightforward IP formulation. This difficulty arises from the fact

for the AGP, the set to be covered and the covering set is both in most cases uncountably infinite [41]. Typically the discretization of a continuous problem leads to imperfections, from which can be driven only approximately optimal solution of the general continuous problem.

In summary, the AGP remains an open subject with significant contributions to several modern applications such as sensor and landmarks placement, motion planning, computer graphics and computer-aided architectural design [16]. The appeal of these trending applications makes researchers around the world seek more practical approaches and solutions while taking advantage of the solid theoretical base from decades of research on the AGP. Similar works as the one found in [21] and the one presented in this thesis aim to combine the well-known theoretical fundamentals of CG with realistic assumptions that are more suitable in real worlds applications, especially regarding sensors and landmarks placement problems.

2.4 Optimization Problem

Mathematical Optimization plays an important role in several areas such as economics, computer science, logistics and engineering. The essence of an *Optimization problem* consists in finding the ”best” configuration of a set of parameters to achieve some goals hence is required to identify an *objective*, a quantitative measure of the performance of the system. The objective is a scalar function of certain parameters of the system, called *variables* or *unknowns*. The variables can be constrained by budget, time, space or any type of *constraints* that yield from the nature of the system at hand.

Mathematically, an optimization problem consists of maximizing or minimizing a scalar function f, subject to constraints on its variables. To formulate an abstract optimization problem consider:

- f a scalar that denotes the *objective function* to be maximized or minimized;
- x the vectors of variables or unknowns;
- c_i the constraints on the values of x.

Finally, the optimization problem can be written as follows:

$$\min_x f \quad \text{subject to} \quad \begin{cases} c_i(x) = 0 & i \in \mathcal{E} \\ c_i(x) \geq 0 & i \in \mathcal{I} \end{cases} \tag{2.3}$$

here $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for equality and inequality constraints, respectively.

Depending on the nature of the variables, optimization problems can be classified into three categories:

- A *discrete optimization problem* has discrete variables and the optimal solution is found from a finite or infinite countable set. A discrete optimization problem may contain not only integers and binaries variables but also more abstract variable objects including permutations, set and graphs.
- A continuous optimization problem has continuous variables.

- Mixed Integer Programming (MIP) are optimization problems that have both discrete and continuous variables.

Due to the *smoothness*, continuous optimization problems are usually solved by local solvers, which using derivative information achieve in most cases very efficient optimization. On the other hand, the objective function of discrete optimization problems generally presents several non-differentiable points, as well as sudden variations in the objective function and constraints which yields an extremely *non-convex behavior* of the feasible set [42]. The difference between discrete and continuous optimization problems regarding the nature of the variables and behavior of the objective leads to distinct approaches and algorithms to find the optimal solution.

Continuous optimization algorithms are in general *derivative*-based methods including *Line search*, *Trust region*, *Newton-methods* and others [42]. These methods are iterative and use derivative information available at the current iterate step to generate a quadratic model of the objective function. Based on the model it is chosen future steps in the gradient direction to minimize (or maximize) the objective function.

In the case of discrete optimization especially for *Integer Programming problems*, in which typically the lack of smoothness of the objective function does not allow the use of derivative information in the optimization, the most popular methods are *Branch-and-Bound* and *Branch-and-Cut*. To understand better these kinds of methods the next section is focused on a particular subset of discrete optimization that is the Integer Programming problem.

2.5 Integer Programming Problem

Some authors consider discrete optimization as consisting of *Integer Programming* and *Combinatorial Optimization Problem* (COP) which variables would be other than integers, for instance, graph structures, however, these two topics are closely intertwined and COP models are often referred to as integer programming models. Classical combinatorial problems such as *Knapsack Problem*, *Traveling Salesman Problem* and the Set Covering Problem are often modeled as IP [43]. In this type of problem, the vector of variables is binary, representing *on/off* decisions.

The most general definition for an IP is:

$$\begin{aligned} &\min f(x) \\ &g_i(x) = 0 \quad i \in \mathcal{E} \\ &g_i(x) \geq 0 \quad i \in \mathcal{I} \\ &x = (x_1, \ldots, x_n) \in \mathcal{Z}^n \end{aligned} \tag{2.4}$$

However, this section focus on optimization problems in which the objective function and constraints are linear, such kind of problems are called *Integer Linear Programming*. On some occasions, one will refer to ILP as just Integer Programming. The canonical form of Integer (Linear) Programming can be written

as:

$$\begin{aligned} &\min cx \\ &Ax \leq b \\ &x \geq 0, integer \end{aligned} \tag{2.5}$$

Where A is a m by n matrix, b is a m-dimensional column vector and c a n-dimensional row vector. The vector of variables is denoted by $x \in \mathcal{Z}_n$. As said before ILPs are generally solved by algorithms such as Branch-and-Bound and Branch-and-Cut [44].

The branch-and-bound algorithm can be described as the process of trying to construct a proof that a solution is optimal, based on successive partitioning of the feasible set. During this partitioning process are estimated lower bounds (or upper bound in case of a maximizing problem) that are used to construct a proof of optimality **without exhaustive search** [43, 44]. For ILP the lower bounds are derived from a Linear Programming *relaxation*. Relaxation of an ILP is the process of removing the integrality constraint of each variable.

The ILP in 2.5 can alternatively be written as:

$$z_{IP} = \min\{cx : x \in S\}, \quad S = \{x \in \mathcal{Z}_+^n : Ax \leq b\} \tag{2.6}$$

By removing the integrality constraints we obtain:

$$z_{LP} = min\{cx : x \in \mathcal{P}\}, \quad \mathcal{P} = \{x \in \mathcal{R}_+^n : Ax \leq b\} \tag{2.7}$$

Solving the LP relaxation, we obtain x^0, which in general is not an integer. The cost $c(x^0)$ of the obtained solution is a lower bound on the optimal cost of the Integer Programming problem. Thus, if x^0 is an integer then it is the solution of the original problem, in other words, if $x^0 \in S$ and $c(x^0) = z_{LP}$ then x^0 is the optimal solution of the IP. Back to the case where a component x_i^0 of x^0 is not an integer, the initial problem (2.5) is split into two subproblems adding two mutually exclusive constraints:

$$\begin{aligned} Problem1 \qquad &\min cx \\ &Ax \leq b \\ &x \geq 0, integer \\ &x_i \leq \lfloor x_i^0 \rfloor \end{aligned} \tag{2.8}$$

$$\begin{aligned} Problem2 \qquad &\min cx \\ &Ax \leq b \\ &x \geq 0, integer \\ &x_i \geq \lfloor x_i^0 \rfloor + 1 \end{aligned} \tag{2.9}$$

The solution of the Integer Programming problem is in the feasible region of one of these two sub-problems. Then, one of the subproblems is chosen and the procedure is repeated. The branch-and-bound algorithm can be understood as a *binary tree*, where the feasible region of the original problem is the root node and the other nodes are the feasible set of the subsequent sub-problems. The process of splitting the feasible region of a node by generating new problems represents the branching of the node's two children.

A given node j of the tree can be pruned by the following criteria:

- Infeasibility $\mathcal{P}^j = \emptyset$ where P^i is the feasible region of the *ith* problem;
- Optimality $x^j \in \mathcal{Z}_+^n$ and it is the solution of the IP
- Value dominance $z_{LP}^j \geq z_{LP}^m$ where z_{LP}^m is the optimal cost of a node m, any descendent node of j with a solution x would have a solution cost $c(x) \geq z_{LP}^j \geq z_{LP}^m$, hence the node j is pruned

The Branch-and-cut combines branch-and-bound and the *cutting plane* method. The principle of the cutting plane method is to add new constraints to an ILP that does not exclude integer feasible points, then the optimal integer solution is preserved. Such constraints are called cutting planes. In summary, this strategy consists of successively adding such constraints to an ILP, until the solution to the LP relaxation is an integer. For a more detailed explanation of Integer and combinatorial optimization and the solving algorithms, one refers to the book of Wolsey and Nemhauser [43].

This method combines the branch-and-bound and the cutting plane method.

2.6 Multi-criteria optimization problems

Several real-world optimization problems involve simultaneous optimization of multiple objectives, these types of optimization problems are called *Multi-objective* or *Multi-criteria* optimization problems. A multi-criteria optimization is a useful tool in different areas especially in situations that require trade-offs where at least two objective functions are conflicting. Mathematically one can define a multi-criteria optimization problem as:

$$\begin{aligned} &\min(f_1(x), \ldots, f_k(x)) \\ &\text{subject to} \quad x \in S \end{aligned} \tag{2.10}$$

With the objective functions $f_i : \mathcal{X} \to \mathcal{R}$ to be minimized (or maximized) simultaneously, where $\mathcal{X}$ is the decision variable space (in case of integer programming $\mathcal{X} = \mathcal{Z}_+^n$). The feasible set S is a subset of the variable space $\mathcal{X}$ defined by the constraints of the problem that are not explicit in (2.10).

In the case where objective functions are conflicting, there is no single solution that is simultaneously optimal for all objective functions, instead, this kind of problem gives rise to a set of trade-off optimal solutions known as *Pareto-optimal* or *non-dominated* solution.

Definition 2.6.1. A decision vector x_p is said to be Pareto optimal or non-dominated solutions if none of the components of the *objective vector* $f(x_p) = (f_1(x_p), \ldots, f_k(x_p))$ can be improved without deterioration to at least one of the other components.

The set of non-dominated solutions is called the *Pareto frontier* or *Pareto optimal set*.

2.7 Landmarks Placement for Robot self-Localization

The rising demand for applications such as mobile robotics, indoor positioning system and ambient assisted living services has led to an increasing interest in indoor localization. Over the last few years, a lot of research has been made to improve and design more reliable solutions for the indoor localization problem [9]. The ability to navigate accurately is fundamental for mobile robots to execute a variety of tasks including surveillance, transportation and manipulation, especially indoors which often contain ambiguous areas, obstacles and traffic (of people or other robots) that are challenges to the attempt of maintaining robust navigation. A common approach to overcome this challenge is the use of artificial landmarks from which is possible to measure the relative position and orientation [6, 10]. Several applications in mobile robotics rely on artificial landmarks as standalone or integrated with different technologies to achieve accurate and robust localization [45–48]. Different types of landmarks are used in several indoor localization solutions such as based on time-of-flight (ToF), radio signal strength intensity (RSSI) measurements [49, 50], detection of radio frequency identification (RFID) tags [51, 52] and vision-based systems [53].

For practical reasons it is not convenient to consider an arbitrarily large number of artificial landmarks since generally, the computational power on-board of the robot is limited, which imposes substantial limits on the number of landmarks that can be placed [54], as an excessive number of landmarks pose serious scalability and cost issues [10]. Therefore, it is of interest to select the smallest number possible of landmarks while still providing reliable and accurate localization. This raises the optimization problem of where to place the landmarks within the environment in order to ensure a reliable localization. The landmarks placement problem has been addressed with several approaches.

AGP-based approaches

A common strategy is to face the landmark placement as an AGP, which benefits from the theoretical knowledge yielding from decades of research [14, 15]. As established in theory $\lfloor \frac{n}{3} \rfloor$ guards are sufficient and occasionally necessary to cover a polygon of n vertices, in this case, it is assumed that the guards have complete peripheral perception and can detect features in the environment that are either infinitely far or arbitrarily close. The LPP can be reformulated in terms of a sensor placement problem exchanging landmarks by sensors [55]. In [55] an algorithm based on the AGP is proposed to compute the position of visual landmarks, where is assumed that the landmarks are planar patch and that can be pasted in planar walls of the environment that which its floor plan is simplified by a polygonal shape. The objective is to maximize the area in which the robot has clear line sight to at least one landmark. The

basic principle is that the optimal landmarks placement maximizes the sum of the areas of intersection between the visibility region (the range of distances and the range of angles from where the visual system can extract information from a landmark) of each landmark and the polygonal environment. A *Simulated Annealing* algorithm is proposed to determine the optimal set of landmarks. As highlighted by the author, this approach may be complementary to path planning methods based on the assumption that a given set of landmarks already exists [56, 57] such that the landmarks create regions where both control and positioning sensing are ensured. An AGP-based approach to solving the problem of colored landmarks is proposed in [58]. The same landmark placement problem is considered, however, is added the constraint that two landmarks of the same color can not simultaneously be visible from any point of the map, the aim is to ensure that the landmarks are distinguishable. A theoretical bound was established for the minimum number of classes (landmarks colors) needed for any guard set.

The main drawback is the fact that in real-world applications most of the sensors are not omnidirectional neither have an infinite range. Thus, has emerged more realistic approaches to address the problem, which take into consideration characteristics of the sensors such as field-of-view, resolution (in case of optical systems) and even the dynamic motion model of the robot. A discrete instance of the AGP employing realistic requirements of computer vision applications is proposed in [21] to solve the general camera placement problem. In this method, each coverage region is converted into a discrete domain represented as a grid. The goal is to find a set of cameras that maximize a given cost function. This approach is more suitable for real-world applications when compared with other AGP-based methods since it considers realistic configurations of the cameras such as limited FoV, finite depth-of-field (DoF) and resolution constraints. Despite the similarities, this approach distinguishes from our method as it assumes active guards, as the landmarks are exchanged by cameras which simplify the coverage problem in comparison with a passive-landmark placement problem where the detection of a landmark does not depend only on the position of the robot but also on its orientation, this, obviously in case the onboard sensor has a limited field limited field-of-view.

Control-based approaches

In [54] the extended problem of placing landmarks for localization and control of a mobile robot that carries out, predefined navigation tasks are addressed. It is used a linearized version of the system dynamic and sensor observation model to incrementally place the landmarks. The goal is to find the minimum set of landmarks for which a bound is guaranteed on the maximum deviation from the desired trajectory. This method is suitable for real-world mobile robotics applications as it can deal with arbitrary trajectories. In addition, the maximum allowed deviation of the robot can be defined individually for every part of the trajectories according to the risks and consequences of a collision in different regions of the environment.

A similar approach that aims to optimally place landmarks given a predefined trajectory of an unmanned vehicle (UV) is presented in [59]. The goal is to ensure that the system is always observable from a control standpoint, guaranteeing that the error in state estimates of the position and heading of the UV remains bounded while taking into account collision avoidance constraint and the restricted FoV

of the cameras. It is assumed a discrete set of potential landmark locations in contrast with [54] where the landmarks are selected from continuous space. In this approach, the trajectory of the UV is defined by a set of target points, that the vehicle visits exactly once until it reaches a destination point. Thus, the landmarks placement problem is formulated as a graph-theoretic problem where portions between two consecutive target positions are the edges of the graph. The problem is divided into sub-problems (one for each edge) in the UV path, a greedy algorithm is then used to obtain the optimal solution of each sub-problem, and finally, the solutions to the sub-problems are put together to construct a feasible solution of the global problem.

Geometrical non-AGP approach

A solution for building an accurate and reliable localization system based on combing artificial and natural landmarks is presented in [60]. This method selects the landmark positions on the walls of an indoor environment close to a given set of localization points. The landmark placement is done by defining and optimizing a *confidence level* describing the expected localization estimation error. The optimization criteria, the confidence level is based on geometrical information such as the distance from the observation point and the landmarks, the observation angle and the numbers of visible landmarks.

The optimal sensor placement with limited detection area in ideally unbounded rooms is addressed in [10]. The goal is to determine the minimum number of landmarks for any given configuration of the sensors such that at least one landmark deploys inside the sensor detection area. The problem is tackled as a *tiling problem*, where the vertices of the tiles coincide with the position of the landmarks and the goal is converted into determining the maximum distance between the landmarks that guarantee the detection of at least one landmark. A numerical and analytical solution was presented to solve the optimization problem. The strength of this approach is that the configuration parameters of the sensors are not neglected in contrast with [54] where is considered a circular field of view. However, the main drawback is that this approach is not immune to occlusions, since in real environments the detection of at least one landmark is not guaranteed due to obstacles.

Discussion

Depending on the application and its specifications there are several approaches to address the landmarks placement problem. In general, AGP-based approaches consider ideal sensors, which make these methods unsuitable for many applications. Control-based approaches are interesting especially in robot self-localization, however, the main drawback is the limited number of specific trajectories used in the optimization. Some methods presented in this survey, especially the ones based on the AGP rely on approximation or meta-heuristics algorithms which, despite high efficiency do not guarantee optimality, in contrast with our method that uses a standard ILP solver to obtain the optimal solution. A common aspect between our method and the majority presented here in this survey is the discretization of the landmarks placement problem, either considering a finite set of poses from specific trajectories or a finite set of potential landmarks locations from which is selected an optimal subset. Of course, discretization

may lead to imperfections. Increasing the cardinality of the sets of poses or potential landmarks leads to more robust solutions, however penalizing the overall cost of the applications. Our method considers non-pointwise landmarks, which to the best of our knowledge is barely addressed in the literature, and in our understanding, this feature may be of interest in applications such as indoor advertising placement.

Chapter 3

Landmark Optimal Placement Problem

3.1 Problem Statement

In theory, an infinite number of landmarks ensure that there are no occlusion or non-covered areas. However, it is not realistic to consider an arbitrarily large number of landmarks, due to the cost of the system and computation capability required to the pose estimation. Hence, it is of great interest to optimize the placement of the landmarks. As mentioned in Chapter 1, real-world landmarks are obviously non-pointwise. In addition, for applications as indoor advertising would be more realistic to consider the size of the landmarks in the placement problem.

Several solutions to the landmarks placement problem are discussed in Chapter 2, however, the topic remains an open issue, due to the difficulty to find a solution for the general continuous problem and discretization is a common strategy to obtain a tractable solution [10, 21].

This chapter addresses the problem of determining the optimal location of a given set of landmarks both pointwise and non-pointwise in a given environment. Next is presented the key elements for the problem formulation, namely the environment, landmarks and the characteristics of the detection system.

Environment

In Robotics is usual to model a complex real-world environment to a simplified 2D representation, especially regarding Automated Guided Vehicles (AGVs), where the robot maintains contact with the floor, similar simplifications are also suitable for UAV applications where the vehicle keeps its altitude approximately constant. This approach reduces a complex 3D problem to a simpler problem in the plane.

Therefore, this thesis considers a simplified 2D layout representation of the real environment, The map of the environment is assumed to be a closed and connected planar region which, for instance, represents a building floor plan, that is allowed to have *holes* caused by obstacles e.g. columns, walls, furniture. Hence, the map of the environment is modeled as a polygon, consequently, every curve surface is approximated by a set of line segments.

Landmarks

Throughout this thesis, it is considered passive landmarks installed on the edges of a given environment. The landmarks are assumed to be readable by means of a sensor (e.g. cameras or RFID readers), for instance, standard QR codes or state-of-the-art ArUco [61] for cameras and RFID tags for RFID readers [52]. It is considered two possible representations for a landmark l:

- A point representation $l = \{(x, y)\}$, where the landmark is referred to as a *pointwise landmark*;
- A line segment representation $l = \overline{ab}$, where a and b are the endpoints of the line segment. In this case, the line segment has a fixed length:

$$D_l = \|\overline{ab}\| = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

where (x_a, y_a) and (x_b, y_b) are the coordinates of the endpoints a and b respectively. Note that in both representations, a landmark l is treated as a set of points: a set with a single element for a pointwise landmark and a line segment (a set of points) for a non-pointwise landmark. For P, a polygonal representation of a given environment, the location of the landmarks is restricted to the edges of P:

Assumption 1. A landmark can be represented by a set of points. A set with a single element in case of pointwise landmark and a line segment for a non-pointwise landmark.

Assumption 2. The landmarks are restricted to the edges of P. Thus for a landmark l exists an edge e of P such that l is a subset of e, in other words:

$$\exists e \in E(P) : l \subseteq e \tag{3.1}$$

Sensors

It is assumed agents fitted with a sensor system that allows perceiving the external environment by identifying specific landmarks. The sensor detects the landmarks within a limited region of space, characterized by a *field-of-view* angle denoted as θ and a detection range R. The pose of the sensor is represented by $p = (x, y, \psi)$, where x and y are the Cartesian coordinates of the position and ψ denotes the orientation, that is the angle measured between the optical axis and the X^W axis of a world reference frame $\langle W \rangle = (O^W, X^W, Y^W)$. Throughout this work, the region within which a landmark can be detected by a sensor is referred to as the Sensor Detection Area (SDA) [10]. The field-of-view of a sensor is usually defined as the angular range, in which objects can be observed for a fixed orientation of the sensor [62, 63]. In the case of optical sensors, the FoV is in fact a solid angle through which the detector can sense electromagnetic radiation and it is required to specify an angular range in the main directions: vertical, horizontal and diagonal. However, this work considers only the horizontal direction, since the problem at hand is two-dimensional in the horizontal plane.

These assumptions are suitable for real sensors such as standard RGB-D cameras where the visibility depends on parameters as, for example, the field-of-view, depth-of-field and resolution. Also, *light*

detection and ranging sensors (LiDAR) can be modeled similarly. Without loss of generality, the FoV angle is assumed to be less than π, such that the SDA can be approximated by an isosceles triangle with a vertex angle $0 < \theta < \pi$ and height R. Thus, the two equal sides of the triangle have lengths $r = R/cos(\theta)$. The SDA is denoted by s.

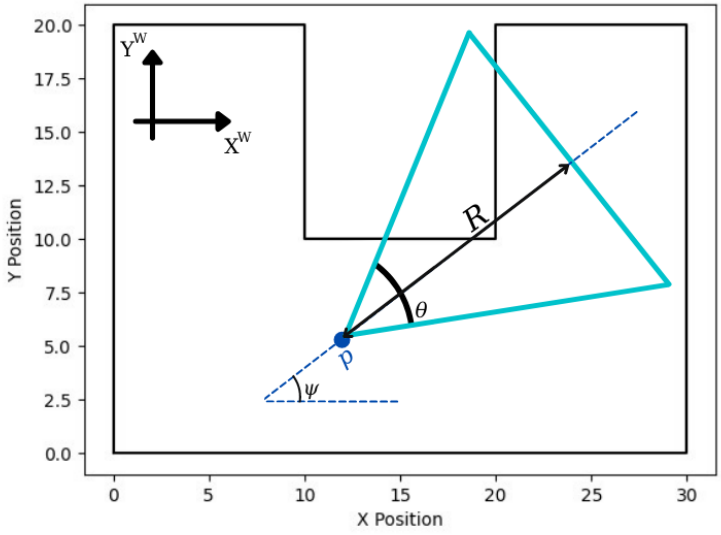


Figure 3.1: Field-of-view and detection range of the sensors.

In a reference frame with origin in the pose p, and with the x-axis aligned with the optical axis, a point with coordinates (x, y) in such a reference frame is within s if:

$$0 < x \leq R \quad \wedge \quad \left|\frac{y}{x}\right| \leq \tan\left(\frac{\theta}{2}\right) \tag{3.2}$$

A point q is visible to a pose p if the following condition is verified:

$$(\overline{qp} \subset P) \quad \wedge \quad (q \in s) \tag{3.3}$$

where $\overline{qp}$ denotes the line segment that connects q with the pose p. In other words, a given point q is visible to p if:

- (i) the line segment connecting q with the position of the sensor is a subset of the polygon and
- (ii) q is inside the sensor detection area.

Note that the first term of condition 3.3 is the concept of *point visibility* presented in [13] and [31], such that the set of points of P that satisfy the first term is the visibility polygon $V(p)$. However, only a subset

of $V(p)$ satisfies condition 3.3 which are the set of points of P that are inside the SDA s. In other, words only the intersection of the visibility polygon $V(p)$ with the detection area s is visible to the pose p. From now on this intersection region is referred to as *limited-visibility polygon* and it is denoted by.

$$v(p) = V(p) \cap s.$$

By definition, the visibility polygon $V(p)$ is a star-shaped polygon. Hence $v(q_i)$ is also star-shaped.

Theorem 3.1.1. For an internal pose p of a polygon P, the limited-visibility polygon $v(p)$ is star-shaped.

Proof. By construction $V(p)$ is star-shaped and p belongs to the kernel of $V(p)$. The limited-visibility polygon is given by $v(p) = V(p) \cap s$, where $s \subset R^2$ is a convex and connected space since $p \in v(p)$, one can conclude that every point $y \in v(p)$ is visible to p, thus $v(p)$ is star-shaped. □

Assumption 3. A given landmark l is visible to a sensor located at the pose p with an SDA s if for every point $q \in l$ the condition 3.3 is satisfied.

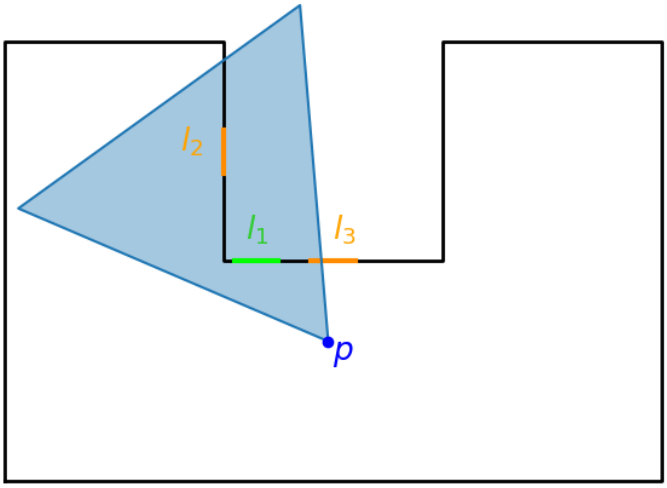


Figure 3.2: The landmark l_1 is visible from the pose p. But l_2 and l_3 are not visible, although l_2 is inside the detection area there is no direct line-of-sight from p while l_3 is not totally inside the detection area.

3.2 Analytical formulation

This thesis presents an ILP formulation to address the optimal landmark placement problem. The method relies on a simulation of a large set of possible poses inside a polygon P, from now on, such poses are called *witness* poses and the witness set is denoted by $\mathcal{W} = \{w_1, \ldots, w_M\}$.

By computing the limited-visibility polygon of each witness, it is possible to determine the regions of the polygon where a landmark can be seen by a maximum number of poses. The intersections between

the visibility polygons of each pose determine whether a certain portion of the boundary of P is visible or not and to which subset of poses it is visible. This way is computed a set of *candidate intervals* along the edges of P, where can be placed a landmark.

Definition 3.2.1. A candidate interval is a connected subset of an edge of P, this subset is totally visible to a non-empty set of witness poses.

Thus, it is redundant to place more than one landmark in the same candidate interval as they would be visible to the same set of poses.

Assumption 4. At most one landmark is placed at each candidate interval.

Consider $\mathcal{L} = \{l_1, \ldots, l_N\}$, a set of candidate intervals where can be placed a landmark. The candidate interval l_j is associated with a non-empty subset of poses $\mathcal{W}_j$, which corresponds to the set of poses that can successfully decode a landmark placed at l_j. The candidate intervals along the edges are computed such that every witness pose can see at least one potential landmark:

$$\bigcup_{j=1}^{N} \mathcal{W}_j = \mathcal{W} \tag{3.4}$$

The primal goal is to select a subset of potential landmarks such that the number of *covered* poses is maximized.

Definition 3.2.2. A witness w_i is said to be covered by a set of candidate intervals $\mathcal{L}$ if at least one candidate interval $l \in \mathcal{L}$, is visible from w_i.

This formulation resembles the *Set-Covering Problem* presented in [43] and [64]. The Set-Covering Problem is a classical combinatorial optimization problem, where given a ground set U and a collection of subsets $\mathcal{S} = \{S_1, \ldots S_N\}$ such that for $1 \leq i \leq N$, $S_i \subseteq U$. The objective is to determine the minimum number of sets from $\mathcal{S}$ such that their union is U:

$$\min |S'| \quad \text{s.t} \bigcup_{S_i \in S'} S_i = U \tag{3.5}$$

where $|S'|$ is the cardinality of the set S'. In [64] a particular variant of this problem is considered, which is the *Maximum Coverage Problem*, that in addition to the traditional Set-Covering Problem is given an integer constant k and the objective is now to pick at most k sets from $\mathcal{S}$ such that the size of the union is maximized. In other words, the goal is to find a subset $S' \subseteq S$ such that:

$$\max \left| \bigcup_{S_i \in S'} S_i \right| \quad \text{s.t} \, |S'| \leq k \tag{3.6}$$

Similarly to [43] and [21], the pose coverage problem in hand is formulated as a Binary Integer Programming, where the authors analytically formulate the set-covering problem in matrix notation as follows:

$$\min cx \quad \text{s.t} \quad Ax \geq b \tag{3.7}$$

Where A is an $M \times N$ incidence matrix, N is the number of elements in the covering set and M is the number of elements in the set to be covered, b is an $M \times 1$ vector whose *ith* element is the right-hand-side coefficient of constraint i, c is a $1 \times N$ vector where c_i is the associated cost of the *ith* element of x with $1 \leq i \leq N$. In turn, x is an $N \times 1$ decision variables vector [21].

Problems 3.5 and 3.6 are important to the formulation of the problem at hand since that for practical reasons it is common to constrain the number of landmarks that may not allow the coverage of the entire environment, which is the goal of problem 3.5. From the conflicting objectives of maximizing coverage and minimizing the number of landmarks arises a *multi-criteria optimization problem*, which allows computing the Pareto Frontier and that can be used as a baseline approach to study a trade-off between the coverage and the number of landmarks. The simplest way to obtain a multi-criteria optimization problem is to adapt the problem in 3.6, transforming the maximum number of landmarks L in an objective function to be minimized.

Let f_1 be a function over a set of landmarks, that gives the number of poses in $\mathcal{W}$ that are covered by the set of landmarks. And let f_2 be the cardinality of the considered set of landmarks (i.e the number of landmarks). The multi-criteria optimization problem can be formulated as follows:

$$\max(f_1(x), -f_2(x)) \quad \text{s.t} \quad \mathcal{C} \tag{3.8}$$

where $\mathcal{C}$ is the set of constraints and x is the decision variable vector that in the case of this present thesis is the positions of a set of landmarks. Figure 3.3 illustrates problems 3.5 and 3.6 in the case of polygon coverage.

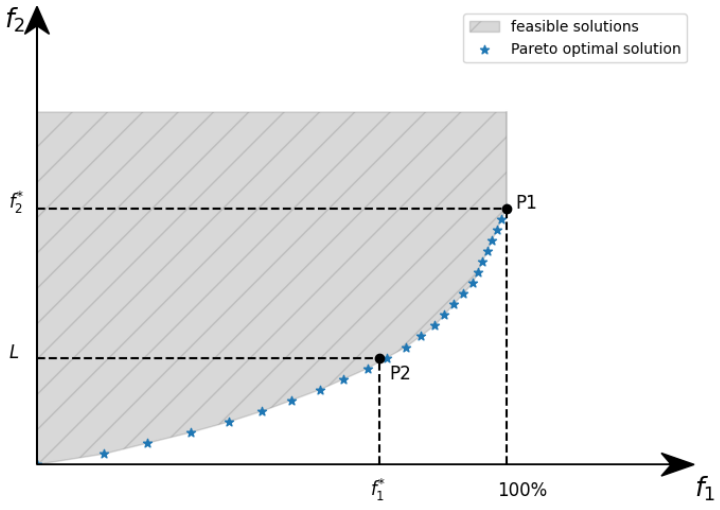


Figure 3.3: Illustration of Pareto frontier in solution space. The point $P1$ determines the minimum number of landmarks f_2^* that cover the entire set of poses $\mathcal{W}$, while $P2$ determines the maximum coverage f_1^* (in percentage) obtained for at most L landmarks.

Next, is presented the mathematical formulation of the coverage problems 3.5 and 3.6. For sake of simplicity, it is first considered a pointwise and followed by an extension to a non-pointwise landmark.

Pointwise landmarks

A pointwise landmark is a single point in the boundary of P, hence, it is assumed that the landmarks can fit within any position in a certain edge of P without any space constraints. Let x be an $N \times 1$ binary decision variable vector and A a $M \times N$ incidence matrix where the element $a_{ij} = 1$ if the witness w_i observes the *j-th* candidate interval, thus $w_i \in \mathcal{W}_j$, otherwise $a_{ij} = 0$. The minimum Set-Cover problem presented in expression 3.5 is now formulated as:

$$\min \sum_{j=1}^{N} x_j \quad \text{s.t} \quad \begin{cases} \sum\limits_{j=1}^{N} a_{ij} x_j \geq 1, \quad i = 1, \ldots, M \\ x_j \in \{0, 1\} \end{cases} \tag{3.9}$$

the M constraints in problem 3.9 guarantee that for any subset of potential landmarks selected the entire witness set $\mathcal{W}$ is covered.

To formulate the Maximum Coverage problem presented in expression 3.6 in matrix notation, it is necessary to define y, an $M \times 1$ binary decision variable and its respective constraints:

$$\sum_{j=1}^{N} a_{ij} x_j \geq y_i \tag{3.10}$$

Note that the *ith* element of y only can take the value of 1 when the *ith* pose is covered (i.e the left-hand side of 3.10 inequality is greater or equal to 1. Therefore, the number of covered poses for a given distribution of landmarks is given by:

$$\sum_{i=1}^{M} y_i$$

Finally, problem 3.6 can be formulated in matrix notation as follows:

$$\max \sum_{i=1}^{M} y_i \quad \text{s.t} \quad \begin{cases} \sum\limits_{j=1}^{N} x_j \leq L \\ \sum\limits_{j=1}^{N} a_{ij} x_j \geq y_i, \quad i = 1, \ldots, M \\ x_j \in \{0, 1\} \end{cases} \tag{3.11}$$

where L is a limit for the number of landmarks that can be used in the environment.

Non-pointwise landmarks

The first difference between a pointwise and non-pointwise approach is the space limitations that arise from the fact that not all edges of P can receive a landmark, since, the edge must have a length greater or equal to the length of the landmarks.

Hence, the procedure to compute the candidate intervals is more complex than it is for a pointwise landmark. As some intersection regions might not satisfy the space constraints, in contrast with a pointwise approach in which the length of a candidate interval is not relevant, since the landmark (a point) can always be placed in the interval. Figure 3.4 illustrates a situation where a portion of the boundary

of P (I_2) is visible to the poses p_1 and p_2, however, due to the length of the landmark considered it is not possible to place one landmark such that it is visible to both poses. Note that for the pointwise approach, the intersection of the limited-visibility polygons would generate three candidate intervals: I_1, I_3 and I_2. The first two are visible to only one of the poses p_1 and p_2 respectively and the third one is simultaneously visible two both poses. However, for a non-pointwise approach, it may be necessary to consider two overlapping candidate intervals: $I'_1 = I_1 \cup I_2$, which is visible only to p_1 and $I'_3 = I_3 \cup I_2$ visible to p_2. For instance, consider the optimization problem 3.11, with $L = 1$, i.e. only one landmark is allowed to be installed within the environment. In this case, it is not possible to cover p_1 and p_2 simultaneously, since for any possible location, the landmark is at most visible to only one of the poses, and the intervals I'_1 and I'_3 are the only possibilities to cover at least one of the pose with only one landmark.

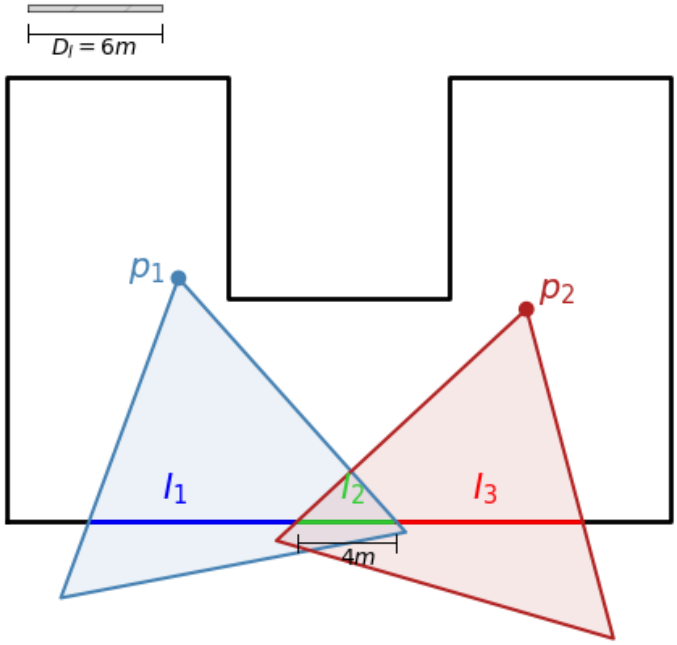


Figure 3.4: The Intervals I_1 and I_3 are visible only for the poses p_1 and p_2 respectively. On the other hand, I_2 is visible from both poses, however, the length D_l of a landmark is greater than $\|I_2\|$.

Let us consider now that two landmarks can be installed in the environment. In this case, it is only possible to cover p_1 and p_2 with two disjoint landmarks only if $\|I_1 \cup I_2 \cup I_3\| \geq 2D_l$. Let I_a and I_b, be two overlapping candidate intervals, visible to pose a and b respectively, such that $\|I_a \cup I_b\| < 2D_l$. It is trivial that there is no pair of disjoint landmarks l_1 and l_2 with $l_1 \subseteq I_a$ and $l_2 \subseteq I_b$, that can cover the poses a and b.

Since, for practical reasons, there can be no overlapping landmarks, it is possible to select only one

interval from a pair of intervals to place a landmark. Such pair of candidate intervals are referred to as *mutually exclusive*. These intervals form a graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ denotes the set of nodes of the graph, a node of the graph is a candidate interval that has a mutually exclusive relation with at least another interval in the graph; $\mathcal{A}$ is the set of *edges* [1] of the graph, an edge of the graph (u, v) means that the candidate intervals with index u and v are mutually exclusive.

Definition 3.2.3. Two overlapping candidate intervals I_a and I_b such that $\|I_a \cap I_b\| < D_l$, are mutually exclusive if every pair of landmarks $l_1 \subseteq I_a$ and $l_2 \subseteq I_b$ are not disjoint.

To prevent mutually exclusive intervals from simultaneously receiving a landmark, a new restriction is added to problems 3.9 and 3.11 for each pair of mutually exclusive intervals. Thus, for a non-pointwise landmarks, the minimum set-cover problem is now formulated as follows:

$$\min \sum_{j=1}^{N} x_j \quad \text{subject to} \quad \begin{cases} \sum\limits_{j=1}^{N} a_{ij} x_j \geq 1, \quad i = 1, \ldots, M \\ x_v + x_u \leq 1, \quad \forall (u, v) \in \mathcal{A} \\ x_j \in \{0, 1\} \end{cases} \tag{3.12}$$

and the maximum-coverage problem is formulated as follows:

$$\max \sum_{i=1}^{M} y_i \quad \text{s.t} \quad \text{subject to} \quad \begin{cases} \sum\limits_{j=1}^{N} x_j \leq L \\ \sum\limits_{j=1}^{N} a_{ij} x_j \geq y_i, \quad i = 1, \ldots, M \\ x_v + x_u \leq 1, \quad \forall (u, v) \in \mathcal{A} \\ x_j \in \{0, 1\} \end{cases} \tag{3.13}$$

[1] Not to be confused with edges of a polygon

Chapter 4

Methodology

4.1 Methodology Outline

An ILP formulation for the landmark placement problem is described in the previous chapter, which consists of finding an arrangement of landmarks from a finite set of candidate intervals to receive a landmark, in order to cover a representative set of poses. Similar approaches in which a finite set of potential markers are discussed in section 2.7. It is also common to consider a finite set of poses, often referred to as witness sets, to find tractable solutions for problems such as the AGP. The main challenge of optimizing landmarks placement based on a witness set is to determine an ideal set of poses that is representative of the environment. This might raise some questions such as: **How many witnesses are needed for a given environment (polygon)?**; **How should be generated the witness set?**.

A method that does not depend on a witness set, potential landmarks, or even predefined trajectories is presented in [55]. However, this elegant solution only maximizes the influence area of the landmarks, in other words, only maximizes the probability of a potential robot pose to detect a landmark, but does not guarantee that a robot in the influence zone of a landmark can see the landmark. In addition, it is used a SA algorithm to place the landmarks, which on its own do not guarantee optimality and on some occasions may be needed to check near all possible combinations of the decision variables which takes $O(2^N)$ time.

In contrast with meta-heuristic solvers like SA, an appropriated Integer Linear Programming formulation approach, in theory, improves the expected time complexity and guarantees an optimal solution. Despite the worst-case computational complexity of standards ILP solvers, such as branch-and-bound search, is the same as the complexity of the brute force search, however, a branch-and-bound approach is generally applied to an NP-hard global optimization problem for which the worst-case complexity gives a little or no insight into the performance of the approach [65]. Those are the reasons why is proposed an ILP to solve the landmarks placement problem.

In figure 4.1 is presented the overview scheme of the implemented method.

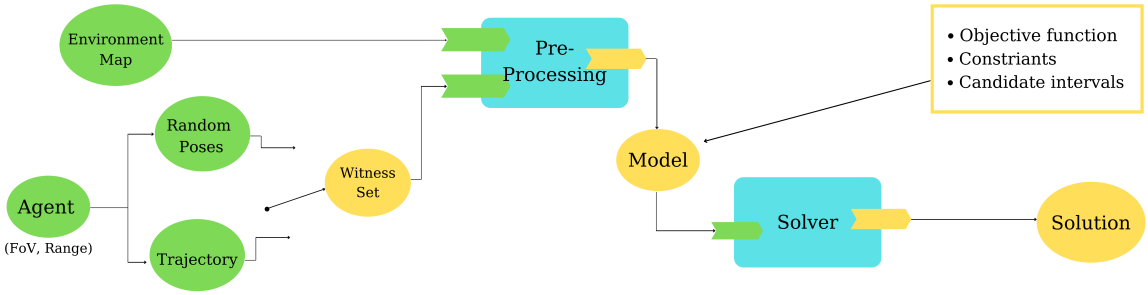


Figure 4.1: Block diagram of the implemented method. The pre-processing algorithm receives as input the map of the environment P; and a witness set $\mathcal{W}$. And has as output a set of candidate intervals $\mathcal{L}$; a set $\{\mathcal{W}_1, \ldots, \mathcal{W}_N\}$ where $\mathcal{W}_j$ is the subset of poses that can see candidate interval l_j; and in case of non-pointwise landmark a graph constraint $\mathcal{G}$ of mutually-exclusive candidate intervals. Finally, the coverage problem is formulated as an ILP and solved.

4.1.1 Pre-Processing

The Pre-Processing phase takes as input the polygon P and the witness set $\mathcal{W}$. And has as output a set $\mathcal{L} = \{l_1, \ldots, l_N\}$ of candidate intervals, each one associated with a potential landmark; the set of subsets $\{\mathcal{W}_1, \ldots, \mathcal{W}_N\} \subset 2^{\mathcal{W}}$, where an element $\mathcal{W}_j$ is the subset of poses that can see the potential landmark placed at l_j and finally the graph constraint $\mathcal{G}(\mathcal{N}, \mathcal{A})$ with all pairs of mutual exclusive intervals in the entire polygon. Algorithm (1) implements the procedure to compute the set of candidate intervals. This algorithm will be also referred to as *Visibility Intersection Algorithm* (VIA). The objective of this algorithm is to compute and identify the intersections of the limited-visibility polygons of all witnesses.

Then, it is identified the edges of the limited-visibility polygon $v(w_i)$ that are also a subset of an edge of P. An edge e' of $v(w_i)$ that is a subset of an edge e_k of P is a candidate interval visible to w_i. Every candidate interval is identified by a *3-tuple*, $\varepsilon = (k, i, u)$ where $k = 1, \ldots, n$, is the index of the edge e_k of P that contains the candidate interval, the second element of the tuple is the index of the witness pose w_i from which is computed the limited-visibility polygon. Finally, the third element is a unique identifier of the candidate interval. The set of all 3-tuples that identify each candidate interval is denoted by $\mathcal{E}$.

From lines, 4 to 14 is performed the identification of the candidate intervals and constructed the set $\mathcal{E}$. The function `VisIntersect` computes the intersections of all identified candidate intervals contained in e_k for every edge e_k of P. Thus, the VIA has as output a set of smaller candidate intervals that are also subsets of the edge e_k.

Time and space complexity

The standard $\mathcal{O}$ notation has been used to evaluate the time and space complexity of the algorithm presented above. The goal is to understand the asymptotic behavior of the running time and memory of the algorithm as the size of the input increases.

To compute the visibility polygon of a pose in a polygon P with n vertices it used the $\mathcal{O}(n \log n)$ time complexity algorithm presented in [21]. Hence, it is needed $\mathcal{O}(nM \log n)$ to obtain the visibility polygon of the M poses. Computing the intersection between the sensor detection area (line 6 of algorithm 1) takes at most $\mathcal{O}(n \log n)$ [66].

Algorithm 1 Candidate landmarks location Algorithm

1: `Input:` Polygon P with n vertices, witness set $\mathcal{W} = \{w_1, \ldots, w_M\}$.
2: `Output:` A list of candidate locations for the landmarks $\mathcal{L} = \{l_1, \ldots, l_N\}$; $\{\mathcal{W}_1, \ldots, \mathcal{W}_N\}$ and the constraint graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ of mutual exclusive intervals
3: $\mathcal{E} = \emptyset$
4: **for** $i = 1, \ldots, M$ **do**
5: Compute visibility polygon $V(w_i)$
6: Compute $v(w_i) = V(w_i) \cap s_i$
7: **for** every edge e_k of P **do**
8: **for** every edge e' of $v(w_i)$ **do**
9: **if** $e' \subseteq e_k$ **then**
10: $\mathcal{E} \leftarrow \mathcal{E} \cup \{(k, i, u)\}$
11: **end if**
12: **end for**
13: **end for**
14: **end for**
15: $\mathcal{L} = \emptyset$
16: **for** $k = 1, \ldots, n$ **do**
17: $\mathcal{L}_k$ = `VisIntersect(`k`, `$\mathcal{E}$`)`
18: $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{L}_k$
19: **end for**

The loop of line 8 of the pseudo-code presented above, search in the list of edges of the limited visibility polygon $v(w_i)$, thus, the time complexity of this search yields from the number of edges of $v(w_i)$ which is complex to estimate a tight upper bound. As an upper bound of the number of edges of the limited visibility polygon, it is taken the worst-case number of edges of the unlimited point-visibility polygon $V(w_i)$: in a simple polygon at the worst-case $V(w_i)$ has n vertices, however for a polygon with holes the number of edges may increase due to the obstacles, thus it is trivial that for every obstacle (holes) can be added two new vertices, if h is the number of holes in polygon P then the visibility polygon may have at most $n + 2h$ vertices.

Proposition 4.1.1. The limited visibility polygon $v(q_i)$ has at most $n + 2h$ edges and vertices.

In the absence of a tighter upper bound, proposition 4.1.1 leads to a rough worst-case scenario of the loop of line 8, with a time complexity of $\mathcal{O}(n + h)$. It is also considered that the number of holes is much smaller than the number of vertices of P, in other words: $h << n$.

Thus, the from line 4 to 14 the time complexity is

$$T_{4-14} = \mathcal{O}\left(M\left(n \log n + n \log n + n^2\right)\right) = \mathcal{O}\left(Mn^2\right)$$

With a similar procedure is computed the space complexity:

$$S_{4-14} = \mathcal{O}\left(Mn\right)$$

which is the space complexity of set $\mathcal{E}$.

From lines, 16 to 19 is computed the intersections of all visible portions of ∂P. Intersections only occur between to subset of the same edge of P, thus for every edge e_k of P is called the functions `VisIntersect(`k`, `$\mathcal{E}$`)` or `VisIntersectNP(`k`, `$\mathcal{E}$`)` for a pointwise and non-pointwise landmarks respec-

tively. For both cases, the running time depends on the size of the set $\mathcal{E}$ and the time complexity is computed afterward in sections 4.2 and 4.3 respectively. The first function is executed in $\mathcal{O}\left(M^2 \log M\right)$ and the latter one in $\mathcal{O}\left(M^3 \log M\right)$.

The running time from lines 16 to 19 of the pointwise algorithm is

$$T_{16-19} = \mathcal{O}\left(nM^2 \log M\right)$$

and for the non-pointwise algorithm is

$$T^{NP}_{16-19} = \mathcal{O}\left(nM^3 \log M\right)$$

Finally, the total running time of the pointwise algorithm is:

$$T_{4-14} + T_{16-19} = \mathcal{O}\left(Mn^2 + nM^2 \log M\right)$$

and for the non-pointwise is:

$$T_{4-14} + T^{NP}_{16-19} = \mathcal{O}\left(Mn^2 + nM^3 \log M\right)$$

These results lead to the conclusion that both algorithms have polynomial time complexity.

Implementation

Algorithm 1 was implemented in Python, and the input data of the polygon and the set of poses are given as independent text files. The output data is written as a mathematical model of the problems presented in chapter 3 using the Zuse Institute Mathematical Programming Language (ZIMPL) [67] which is a mathematical modeling language that translates a model of a problem into a linear or (mixed-)integer mathematical program.

The computational geometry operations namely computing the visibility-polygon are performed using the C++ open-source VisiLibity library [68].

4.1.2 Optimization

The Optimization phase has been addressed in chapter 3, namely the formulation of the problem as an ILP, where is given a detailed explanation of the decision variables, objective function and constraints. That is why, in this chapter, is only given complementary information regarding the analytical formulation of the problem, especially related to the constraint of overlapping candidate landmark locations.

4.2 Algorithm for Pointwise landmarks

To simplify the intersections computations, the initial step is to parameterize the endpoints of each visible portion of the edge e_k, the parameterization is performed such that the entire edge e_k goes from 0 to 1.

Algorithm 2 considers a point landmark as it is assumed that any interval along the edge can receive a landmark. The fundamental principle of the algorithm is that from an input list of overlapping intervals, each one known to be visible to one of the witness poses $w_i \in \mathcal{W}$ is possible to generate an output list of disjoint intervals (the intervals may intersect only in one endpoint) that are visible to a group of poses. The resulting intervals are defined by a pair of consecutive endpoints and each new interval is visible to a set of poses $\mathcal{W}_z \in 2^{\mathcal{W}}$. The set $\mathcal{W}_z$ is computed by the function `Intersection-Query`, which determines which intervals of the input list that intersect the new interval F_z formed by two consecutive endpoints. A pose w_i is added to the list $\mathcal{W}_z$ if the interval F_z lies totally inside of an input candidate interval that is visible to the pose i.

Algorithm 2 `VisIntersect(`k`,` $\mathcal{E}$`)` for a pointwise landmark

`Input:` Edge $e \in E(P)$ and set $\mathcal{E}$ of *detectable* portions of ∂P
`Output:` List of potential landmarks in the edge e_k: $\mathcal{L}_k = \{l_1, \ldots, l_{N'}\}$
$\mathcal{I} = \emptyset$
for every candidate interval c indexed by $\varepsilon = (k, i, u) \in \mathcal{E}$ **do**
    Normalize c to an interval $\iota \subseteq [0, 1]$ such that the edge e_k goes from 0 to 1
    $\mathcal{I} \leftarrow \mathcal{I} \cup \{\iota\}$
end for
Build a balanced binary Interval Tree $T(\mathcal{I})$ according to the value of the left endpoint of each interval
Construct an ordered set $\mathcal{F}$ of the endpoints from the intervals in $\mathcal{I}$
$I = \emptyset$
for $z = 1, \ldots, |\mathcal{F}|$ **do**
    In $T(\mathcal{I})$ perform an `Intersection-Query` for $F_z = [f_z, f_{z+1}]$ and build the list of poses $\mathcal{W}_z$ that can see the interval F_z
    if $W_z \neq \emptyset$ **then**
        $I \leftarrow I \cup \{(k, \mathcal{W}_z, z)\}$
    end if
end for
Transform the intervals in I to line segments (candidate intervals in the edge) and construct the set of potential landmarks $\mathcal{L}_k$

Intersection Query

To handle intersections queries, it was used an *interval tree* [69–71], which is a common interval data structure used in problems such as *Interval intersection*, *Interval stabbing* and *Interval cover*. A dynamic Interval tree allows queries in $\mathcal{O}(k \log M)$, where M is the number of intervals in the tree and k is the number of reported output (output-sensitive), however considering a worst-case in which all intervals in the tree intersect a given query interval than the worst-case scenario to report all intersections is $\mathcal{O}(M \log M)$ [69].

Output

The output of the VIA is the list $\mathcal{L}_k$ where $l_z \in \mathcal{L}_k$ is a potential landmark associated with one candidate interval that is a subset line segment of the edge e_k and it is represented by 3-tuple (k, W_z, z) with $k \in \{1, \ldots, n\}$, $W_z \in 2^{\mathcal{W}}$ and z is a unique identifier of the potential landmark and the candidate interval.

Running time

Every edge has $\mathcal{O}(M)$ input candidate intervals, thus the loop of line 4 has a time complexity of $\mathcal{O}(M)$. Constructing an interval tree with $\mathcal{O}(M)$ intervals take $\mathcal{O}(M)$ time, from lines 11 to 16 is performed a loop that search in a list of endpoints of $\mathcal{O}(M)$ intervals, in each iteration of the loop is performed an intersection query taking $\mathcal{O}(M \log M)$. Finally, it is transformed the output intervals into line segments which also take $\mathcal{O}(M)$ time. Thus the total time complexity of the algorithm is $\mathcal{O}(M^2 \log M)$.

4.3 Algorithm for non-Pointwise landmarks

In contrast with the method described in the previous section, for a non-pointwise landmark approach, it is not possible to guarantee that any visible interval along ∂P can receive a landmark due to the assumption that a "partially visible" landmark is considered as not visible. The main difference between the two formulations is that for a pointwise landmark the candidate locations are disjoint, however, when considered a landmark with dimension it is important to assume the possibility of intersections between candidate intervals. In the algorithm, this difference is implemented by replacing the function `VisIntersect` to `VisIntersectNP` in the algorithm 1. The pseudo-code of `VisIntersectNP` is presented in Algorithm 3.

Algorithm 3 `VisIntersectNP(`e_k`,` $\mathcal{E}$`)` for a non- pointwise landmark

1: `Input:` Edge $e_k \in E(P)$ and set $\mathcal{E}$ of *detectable* portions of ∂P
2: `Output:` List of potential landmarks in the edge e_k: $\mathcal{L}_k = \{l_1, \ldots, l_W\}$ and S a list of pairs of mutually exclusive intervals
3: **if** $D_l \leq$ `length(`e_k`)` **then**
4: $\mathcal{I} = \emptyset$
5: **for** every candidate interval c indexed by $\varepsilon = (k, i, u) \in \mathcal{E}$ **do**
6: Normalize c to an interval $\iota \subseteq [0, 1]$ such that the edge e_k goes from 0 to 1
7: $\mathcal{I} \leftarrow \mathcal{I} \cup \{\iota\}$
8: **end for**
9: Normalize the length of the landmark D_l to a value $d_l \in [0, 1]$ such that $d_l = 1$ if D_l = `length(`e_k`)`)
10: Build a balanced binary Interval Tree $T(\mathcal{I})$ according to the value of the left endpoint of each interval
11: Construct an ordered set $\mathcal{F}$ of the endpoints from the intervals in $\mathcal{I}$
12: $I = \emptyset$
13: $S_k = \emptyset$
14: **for** f_z in $\mathcal{F}$ **do**
15: `Find` the nearest endpoint such that $f_z - f \geq d_l$
16: In $T(\mathcal{I})$ perform an `Intersection-Query` for $F_z = [f_z, f]$ and build the list of poses $\mathcal{W}_z$ that can see the interval F_z
17: **if** $W_z \neq \emptyset$ and F_z is not a subset of an interval already in I **then**
18: $I \leftarrow I \cup \{(k, \mathcal{W}_z, z)\}$
19: Find all the intervals that with F_z forms mutually exclusive pair and add to the set of space constraints S_k
20: **end if**
21: **end for**
22: Transform the intervals in I to line segments (candidate intervals in the edge) and construct the set of potential landmarks $\mathcal{L}_k$
23: **end if**

The intersection queries are performed in a similar way as explained for the pointwise algorithm. The

main difference is that now, the interval $F_z = [f_z, f]$ is not between consecutive endpoints of the input candidate intervals but between two endpoints such that the distance between them is greater or equal to the normalized landmark length d_l.

The time complexity of the algorithm is computed similarly as done previously for the pointwise algorithm. The total time complexity of the algorithm is $\mathcal{O}\left(M^3 \log M\right)$.

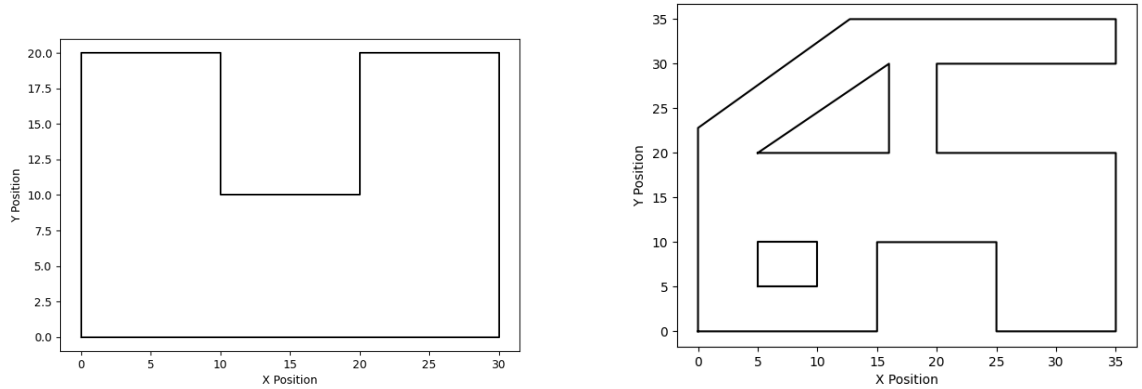
Output

In addition to the list of potential landmarks $\mathcal{L}_k$ it is also computed a list of constraints S_k, one constraint to each pair of mutually exclusive candidate intervals. The constraints are computed by performing an intersection query on a new tree that is formal with the intervals already found visible. S_k is a list of 2-tuples, where each tuple contains the indexes of a pair of mutually exclusive intervals.

4.4 Verification and Validation

This section presents a numerical validation of the designed method. It is approximated the point version of the AGP to a landmark coverage problem based on a finite witness set. Is important to remember that in classical AGP, the guards (landmarks in the case of this thesis) can survey 360º around its position with an infinite range.

The goal is to compare the experimental results with the theoretical bounds on the number of guards (landmarks) needed to cover a certain type of polygon with n vertices and h holes (obstacles). For validation/experimental purposes were considered the polygons shown in figure 4.2. The polygons are referred in this thesis as Room 1 and Room 2 respectively.



(a) Room 1: Simple Polygon

(b) Room 2: Polygon with holes

Figure 4.2: Polygons used in validation experiments.

The polygon shown in figure 4.2 (a) is an orthogonal polygon with 8 vertices, according to theory $\lfloor \frac{n}{4} \rfloor$ guards are sufficient to guard any orthogonal polygon with n vertices, hence 2 guards are enough to guard the considered polygon. A polygon with 20 vertices and 2 holes is presented in figure 4.2 (b).

According to the theory, such polygon can be covered by $\lceil (n+h)/3 \rceil$, hence 8 guards are needed for the second polygon.

Figures 4.3 and 4.4 show empirical results that although are not sufficient to prove the optimality, give an important validation of the designed method, as the theoretical limits on the number of guards were respected in several simulations tests for the considered polygons.

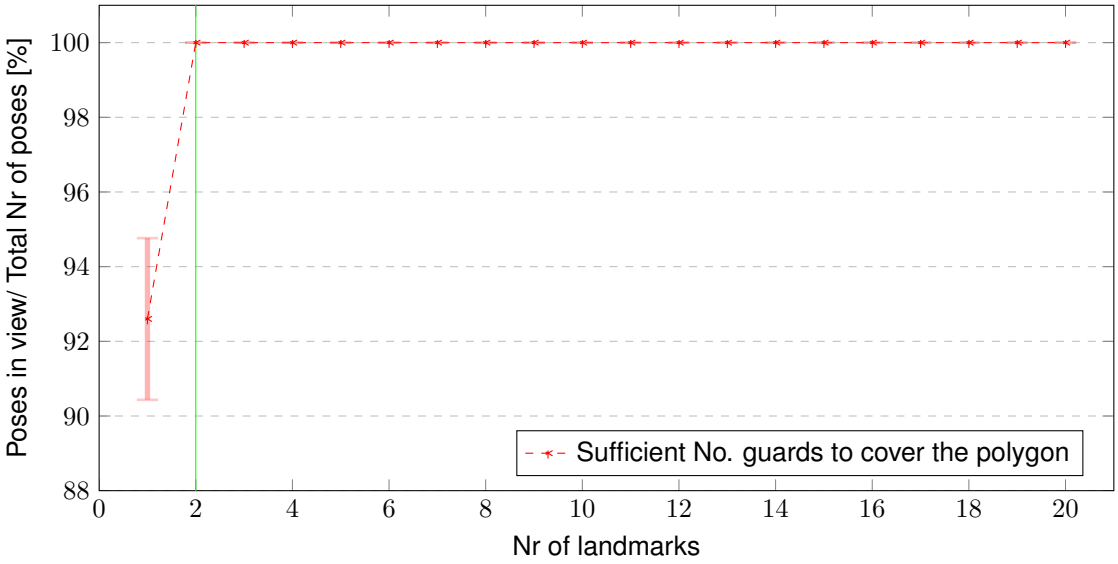


Figure 4.3: Room 1: Coverage percentage per number of landmarks when considered unlimited FoV and infinite range, according to the results, 2 guards were always sufficient to cover all witness poses, which respect the theoretical bound of 2 guards to cover a simple orthogonal polygon with 8 vertices.

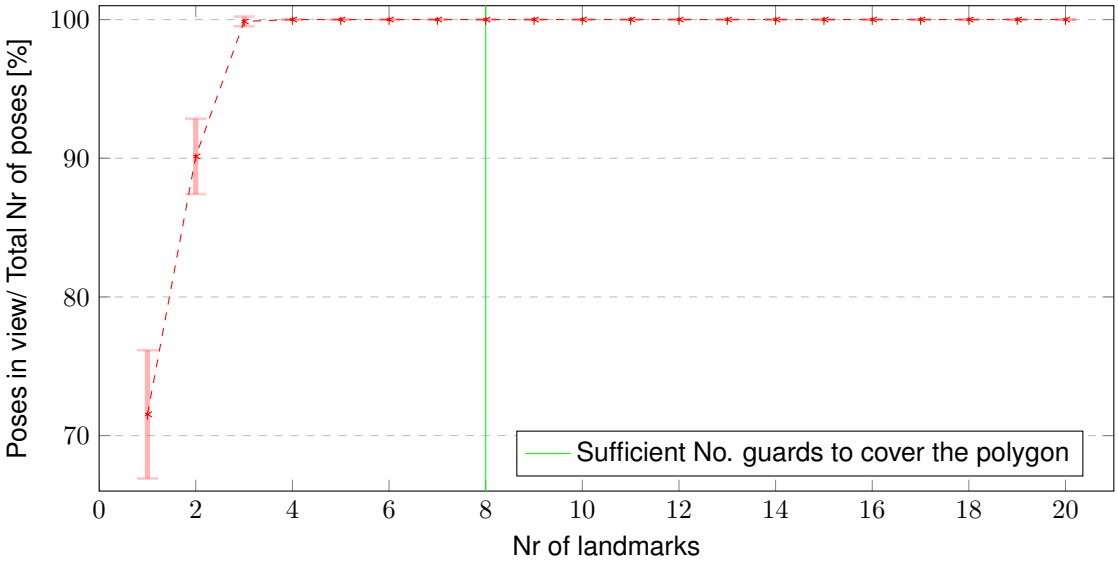
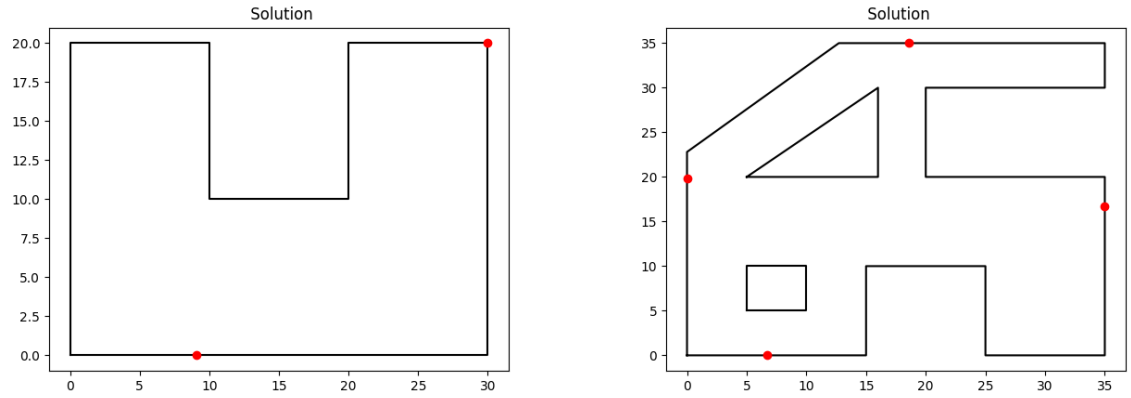
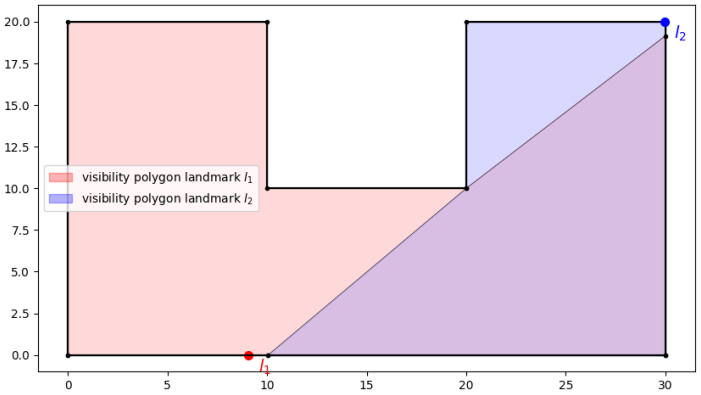


Figure 4.4: Room 2: Coverage percentage per number of landmarks when considered unlimited FoV and infinite range, 3 guards were always sufficient to cover all witness poses, which respect theoretical bound of 8 guards needed to cover a polygon with 2 holes and 20 vertices.

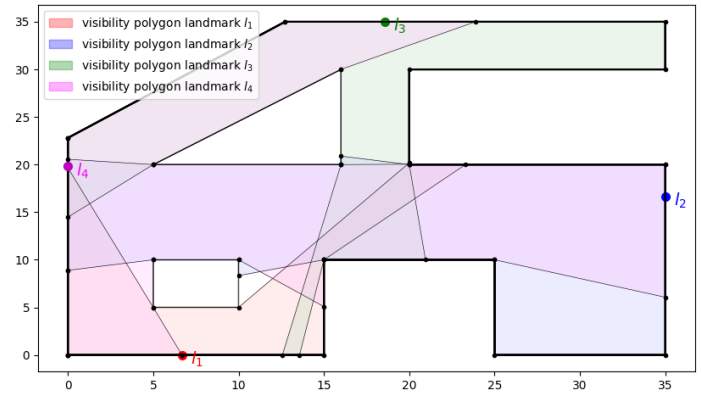


(a) Room 1 (b) Room 2

Figure 4.5: Solutions to the AGP approximated problem for an orthogonal and a polygon with holes.



(a) Room 1



(b) Room 2

Figure 4.6: Covered area by the approximated solution of the AGP in Room 1 (a) and Room 2 (b).

Chapter 5

Numerical Results

This chapter presents the most important results of several simulation tests performed to the landmarks placement optimization method implemented in this thesis. The experiments consider two sources of witness sets: random poses generated with uniform distribution; sample poses from predefined trajectories in the polygons presented in figure 4.2.

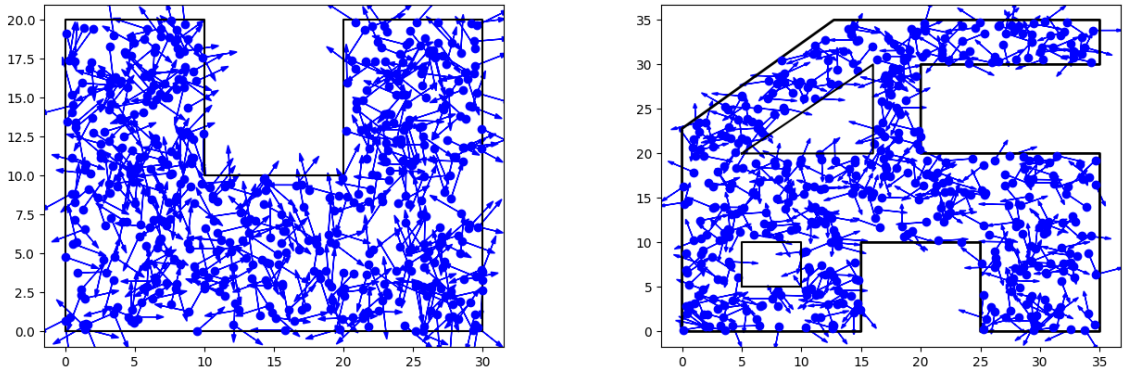
In addition, we present a comparative analysis between our method and a set of meta-heuristics algorithms used to solve the landmarks optimal placement problem. For this purpose, the method has been implemented in python, the experiments have been performed on a 1.6 GHz quad-core computer with 8 GB of RAM. In the comparative analysis we consider a time limit of 300 seconds for each algorithm to determine the optimal solution, in the case where the time limit is reached, the procedure is interrupted and the current best variable vector is taken as the solution.

It is explored the behavior of the algorithms when varying different parameters such as the maximum detection range and the landmarks length. For sake of simplicity, all the witnesses poses are associated with the same type of sensors, hence, with the same parameters configurations. Sensors with characteristics similar to those used by Cordeiro [11] are assumed. Cordeiro assumed a standard drone camera that has a custom-built lens with a 94º FoV and a resolution of 1920x1080p. Cordeiro also considers three different detection ranges R. Where two of these ranges correspond to the maximum distances that a standard drone camera can detect two specific state-of-the-art visual markers: $R = 4.4[m]$ for ArUco and $R = 13.181[m]$ for Whycon respectively. And finally is considered an infinite maximum detection range.

Optimization based on random Witness set

Figures 5.1 and 5.2 present examples of randomly generated witness sets and their respective Pareto frontiers. The following results were obtained based on a statistical analysis of several simulations of different witness sets.

Figures 5.3 and 5.4 highlight that the maximum coverage increases with the detection range of the sensors and with L, the number of landmarks. However the incremental gain, i.e. the difference between the coverage obtained for $L-1$ and L, decreases.



(a) (b)

Figure 5.1: Set of 500 random possible poses for Room 1 (a) and Room 2 (b).

$$G(L) = C(L) - C(L-1) \tag{5.1}$$

where $C(L)$ is the maximum coverage percentage obtained for L landmarks and $G(L)$ is the respective incremental gain.

The author in [11] suggests the stabilization of the coverage percentage as a criterion for a trade-off between the number of landmarks and coverage. Since after a certain point, adding more landmarks to the system only increases the overall system cost and complexity without significant gains in coverage [11]. A coverage assessment based on this criteria is presented in section 5.3 aiming to achieve a trade-off between the coverage and the number of landmarks.

Figures 5.5 and 5.6 show the estimated *Probability Density Function* (PDF) of the maximum-coverage percentage for a given number of landmarks. As one can see in these results, consecutive clusters (for L and $L+4$) tend to be closer as the number of landmarks increases, suggesting a behavior similar to that found in [11].

The dispersion of the maximum coverage shown in figures 5.7-5.9 is an important result that shows the consistency and accuracy of the method implemented in this thesis. A key aspect regarding consistency and accuracy of the method is the selection of the witness set of poses, figure 5.9 shows that increasing the cardinality of the witness set reduces the standard deviation of the maximum coverage, as it approximates the general continuous polygon coverage problem. However, it is important to emphasize that a large number of witness poses increase the computational workload of the pre-processing algorithm and a larger output is generated, which requires more time to read and convert the output of the pre-processing algorithms into an ILP, as well as the optimization time, is also penalized.

Figures 5.7 and 5.8 show a higher dispersion of maximum coverage for intermediate numbers of landmarks. For a detection range of $R = 4.4m$, the dispersion around the mean value of the maximum coverage always increases for the number of landmarks considered in the experiments. Eventually, if considered a greater number of landmarks, a similar behavior to that observed for the other two detection range values would be verified. As the number of landmarks increases, it is more likely to cover any set

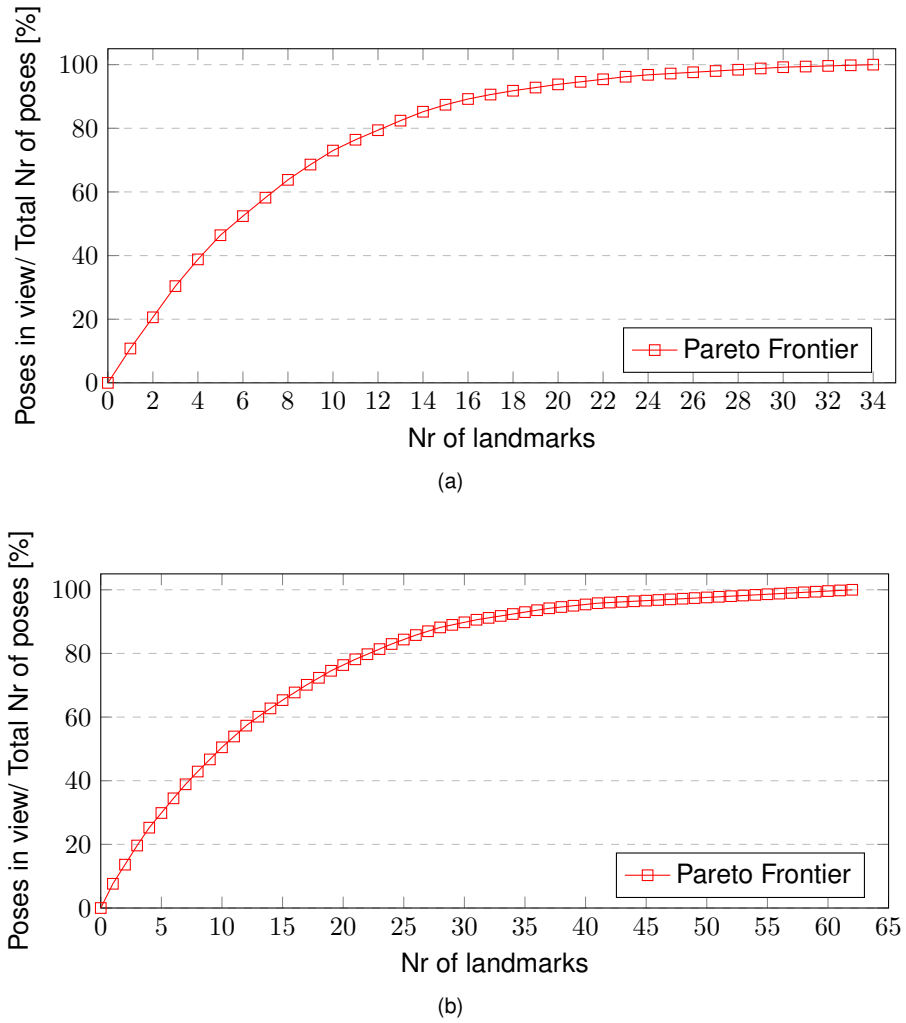


Figure 5.2: Pareto frontier of a given set of random poses for Room 1 (a) and for Room 2 (b).

of poses and the coverage tends to 100%, which roughly explains the less dispersion of the coverage for larger numbers of landmarks. A similar argument can be made to explain the low dispersion for a fewer number of landmarks: the coverage decreases as the number of landmarks approximate to zero; since it is not possible to have coverage less than zero the distribution is flattened for smaller numbers of landmarks which decreases the standard deviation of the maximum coverage. For intermediate values of L it is either possible to have coverage below and above the average (central value), explaining the higher dispersion that occurs for intermediate values. However, it is needed a more formal and detailed explanation of this behavior than the one given in this thesis.

For a robustness assessment, the method is trained and then tested in different witness sets. Therefore, a ***Cross-coverage test*** was performed. The procedure consists of optimizing the landmarks placement for a given set of poses (train) and testing the obtained solution in different sets of poses (testing). The aim is to compare the coverage percentage obtained under training and testing conditions. Let us consider the following witness sets $\{\mathcal{W}_1, \ldots, \mathcal{W}_t\}$ in a polygon P where $\mathcal{W}_i$ and $\mathcal{W}_j$ are two independent sets of random poses for any $i, j \in \{1, \ldots, t\}$ and $i \neq j$. And let x_i and x_j be the solution vectors of length L (maximum number of landmarks) of both $\mathcal{W}_i$ and $\mathcal{W}_j$ respectively. For $i \neq j$, let C_{ij} denotes

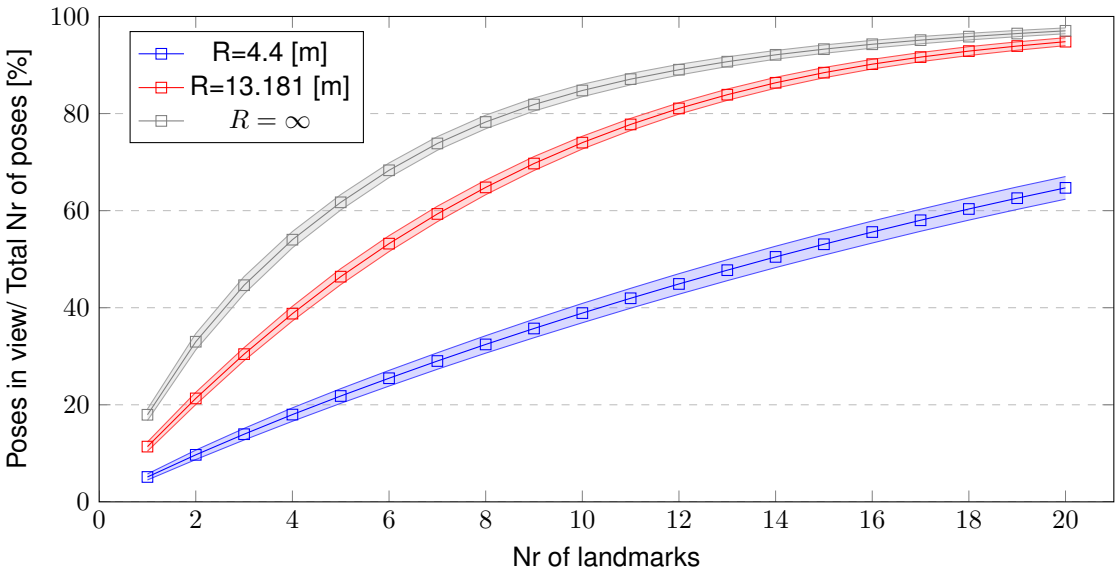


Figure 5.3: Room 1: Coverage graph per number of landmarks, considering 500 poses.

the percentage of $\mathcal{W}_j$ that is covered by solution x_i (testing). Hence, for $i = j$ C_{ij} is equal to the maximum coverage that is obtained with at most L landmarks (training). For a witness set $\mathcal{W}_j$, the difference between the maximum coverage C_{jj} and the cross-coverage value C_{ij} obtained with the landmarks position x_i, expresses how good x_i is, compared to the optimal solution x_j:

$$\Delta C_{ij} = \frac{C_{jj} - C_{ij}}{C_{jj}} 100\% \tag{5.2}$$

The lower the value of ΔC_{ij} with $i \neq j$ the better the solution x_i having as reference the optimal solution. From now on ΔC_{ij} is referred to as the Cross-Coverage relative error. The average of this difference tends to zero when the number of poses increases. Figure 5.11 shows the behavior of $\overline{\Delta C}_{ij}$ for a fixed number of landmarks $L = 10$.

This behavior demonstrates that the larger the set of poses used in the optimization, the more robust the solution obtained is, as the coverage increases even for different sets of poses. The same conclusion can be driven from Figure 5.10, where it is shown a comparison between the maximum coverage and the average cross-coverage. It is possible to note that the two curves tend to converge with the number of poses. While the maximum coverage decreases because the number of visual markers is kept constant and it is more likely to obtain less coverage percentage for a larger number of poses. On the other hand, the cross-coverage slowly increases approximating the maximum coverage. This convergence reinforces that the landmark placement method tends to be more robust when considered a larger set of poses.

Despite the results demonstrating that the robustness increases with the number of poses, the question arises whether the number of poses is representative for a given polygonal environment. Aspects such as the number of vertices of the polygon, the ratio between the environment area and the SDA, the existence or not of holes in the polygon are important to answer this question.

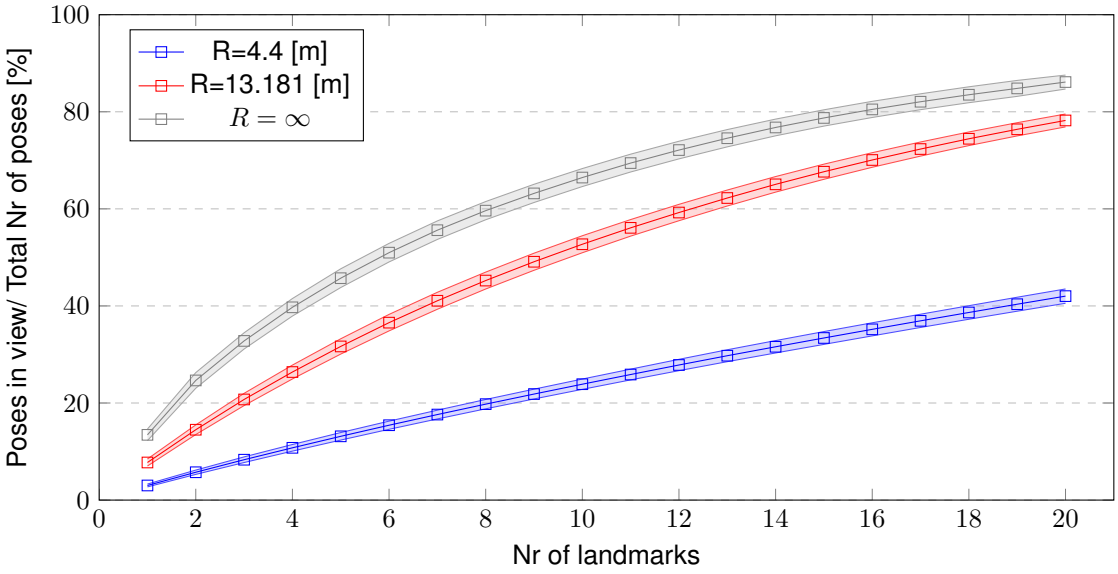


Figure 5.4: Room 2: Coverage graph per number of landmarks, considering 500 poses.

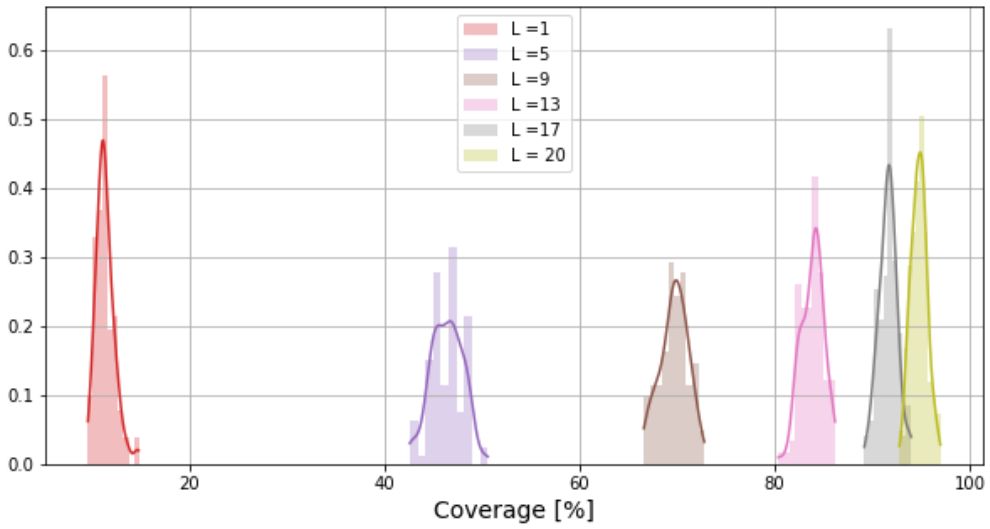


Figure 5.5: Room 1: Histograms of the Maximum Coverage percentage for different number of landmarks (L) and the respective estimated PDF obtained with *Kernel density estimation* method.

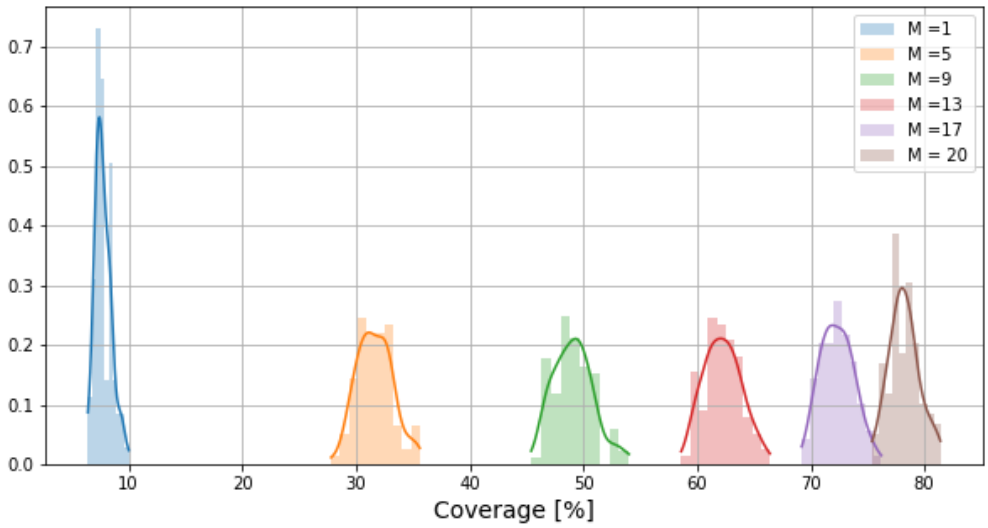


Figure 5.6: Room 2: Histograms of the Maximum Coverage percentage for different number of landmarks (L) and the respective estimated PDF obtained with *Kernel density estimation* method.

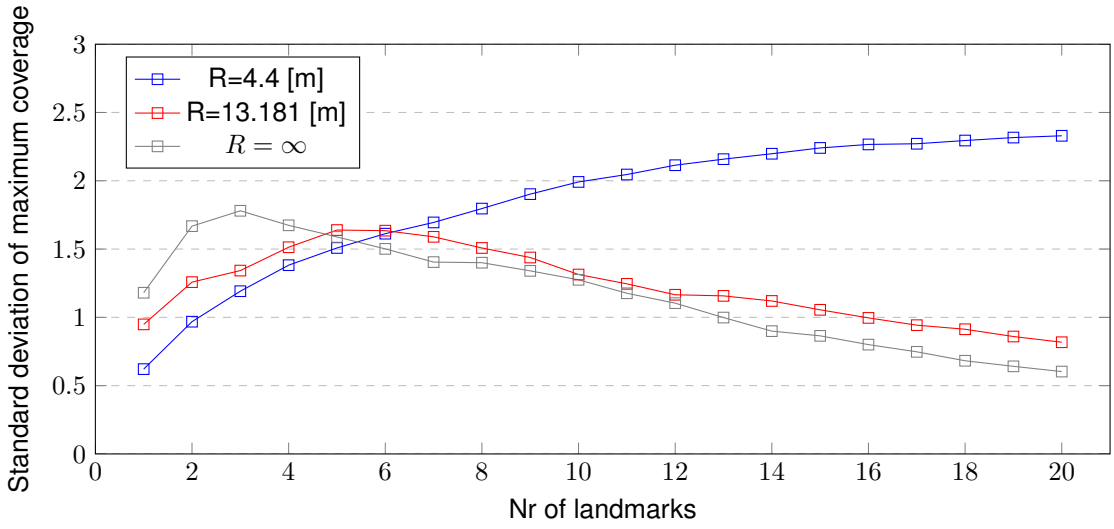


Figure 5.7: Room 1: Standard deviation of the Maximum Coverage per number of landmarks for 500 poses.

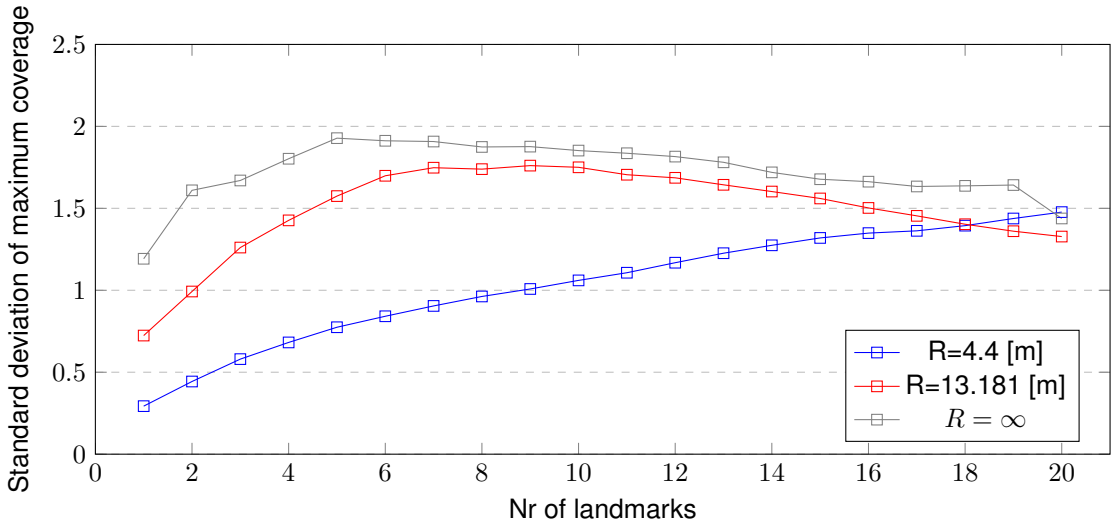
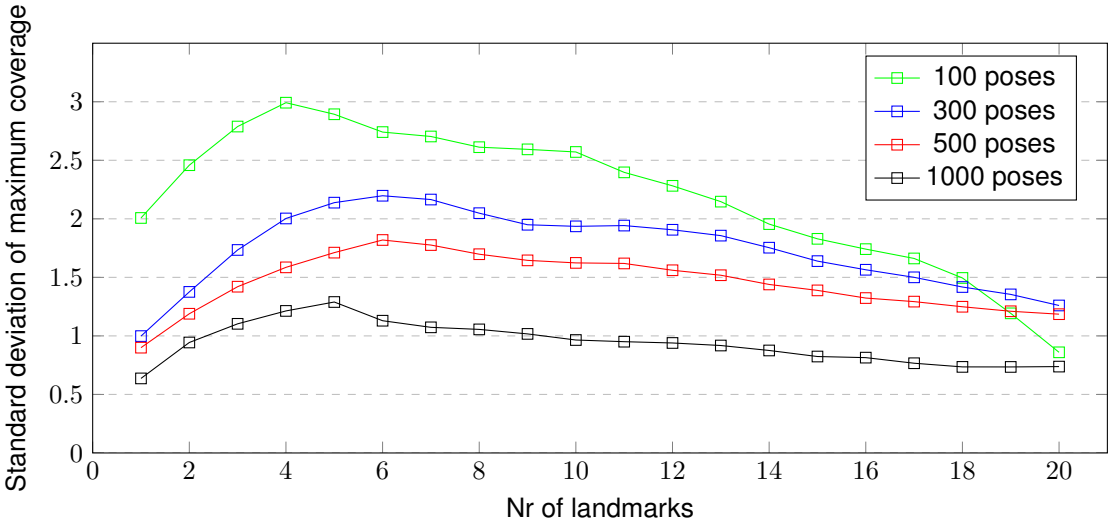
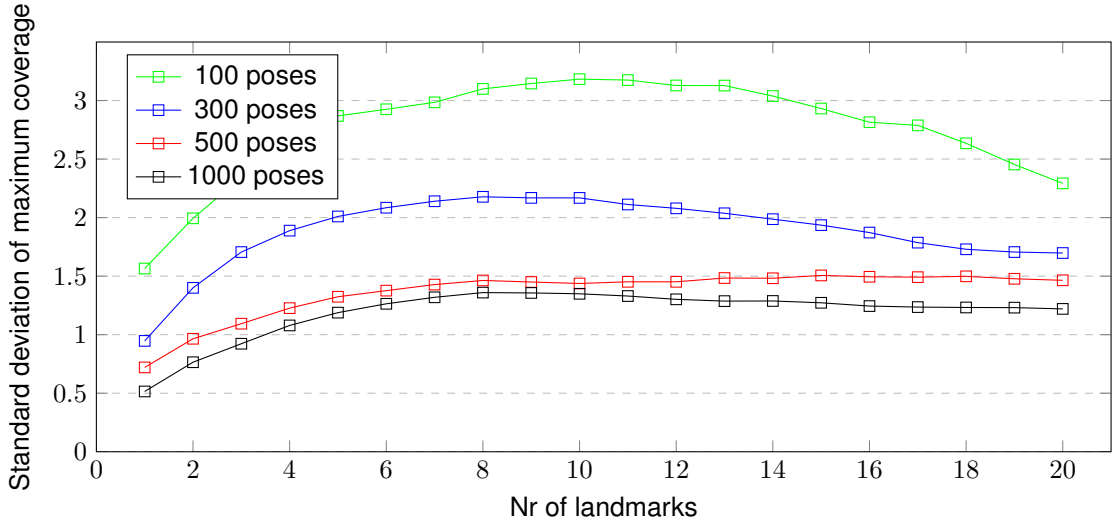


Figure 5.8: Room 2: Standard deviation of the Maximum Coverage per number of landmarks for 500 poses.

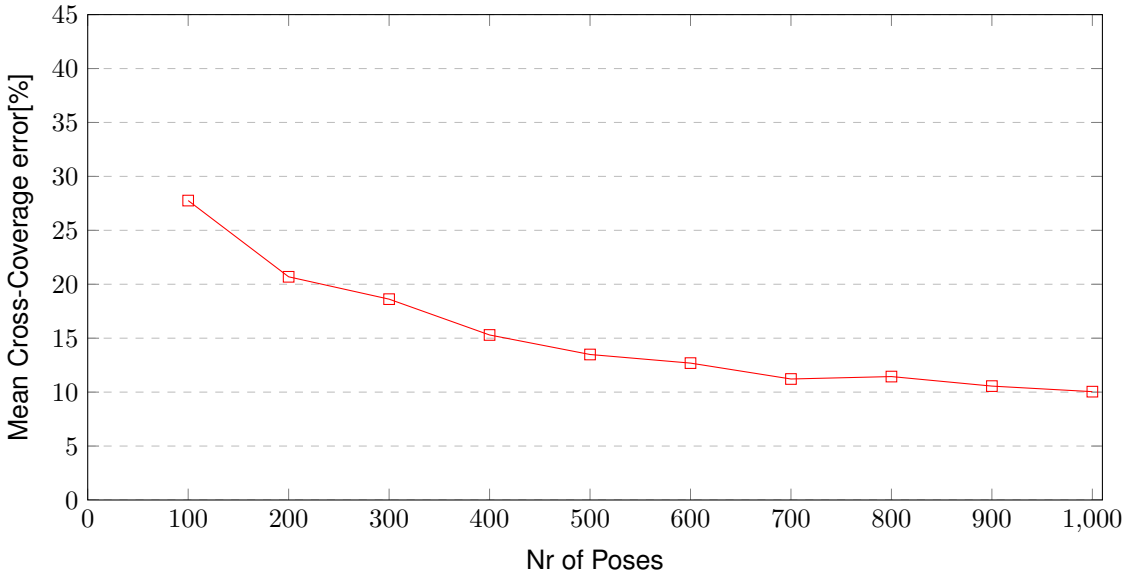


(a)

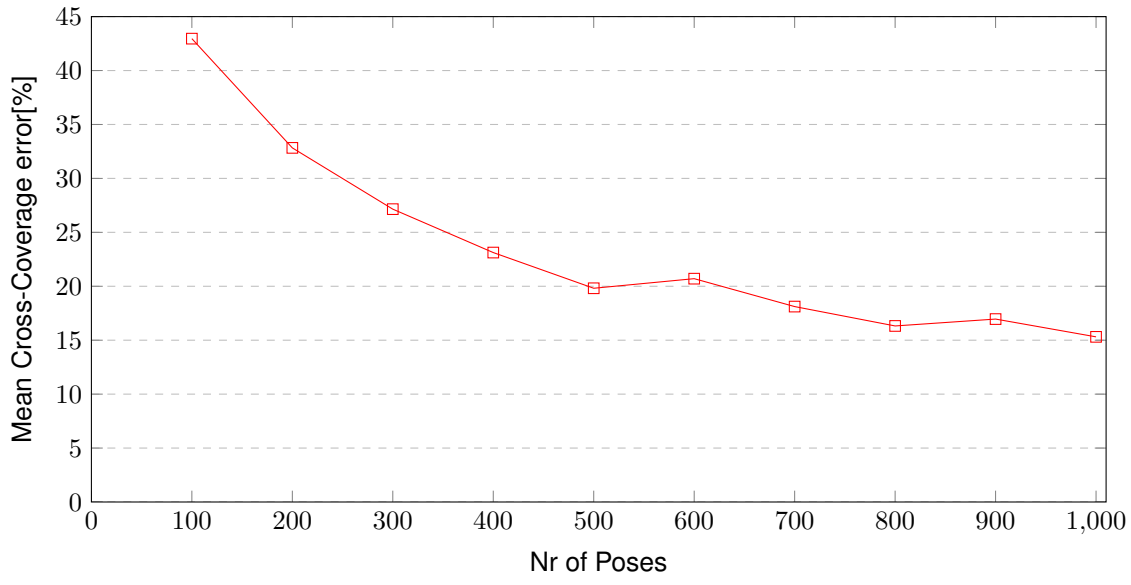


(b)

Figure 5.9: Standard deviation of the Maximum Coverage per number of landmarks with a sensor detection range of $R = 13.181m$ in Room 1 (a) and Room 2 (b).

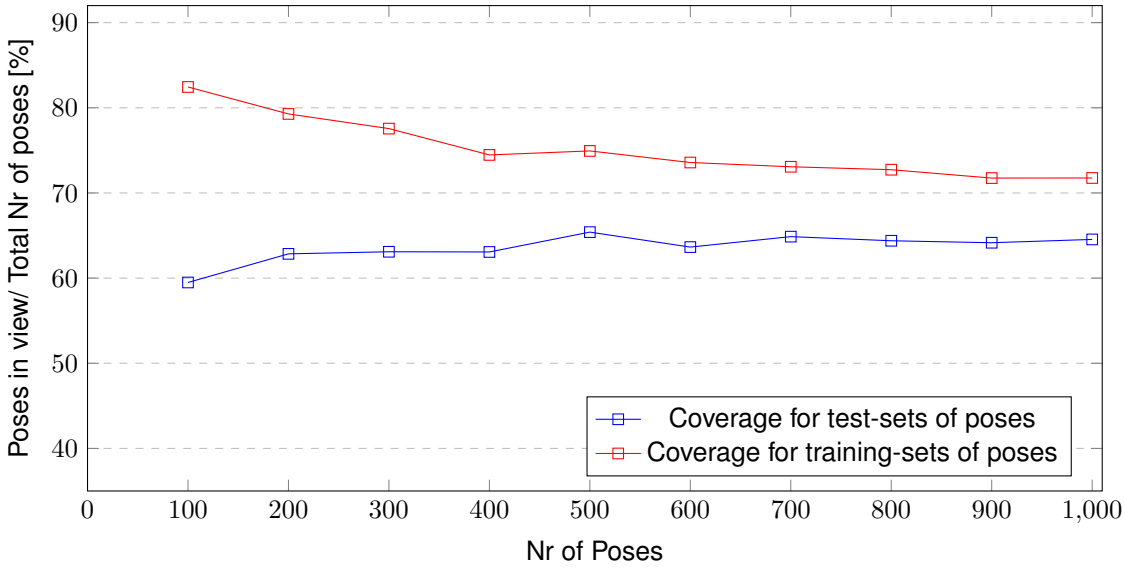


(a)

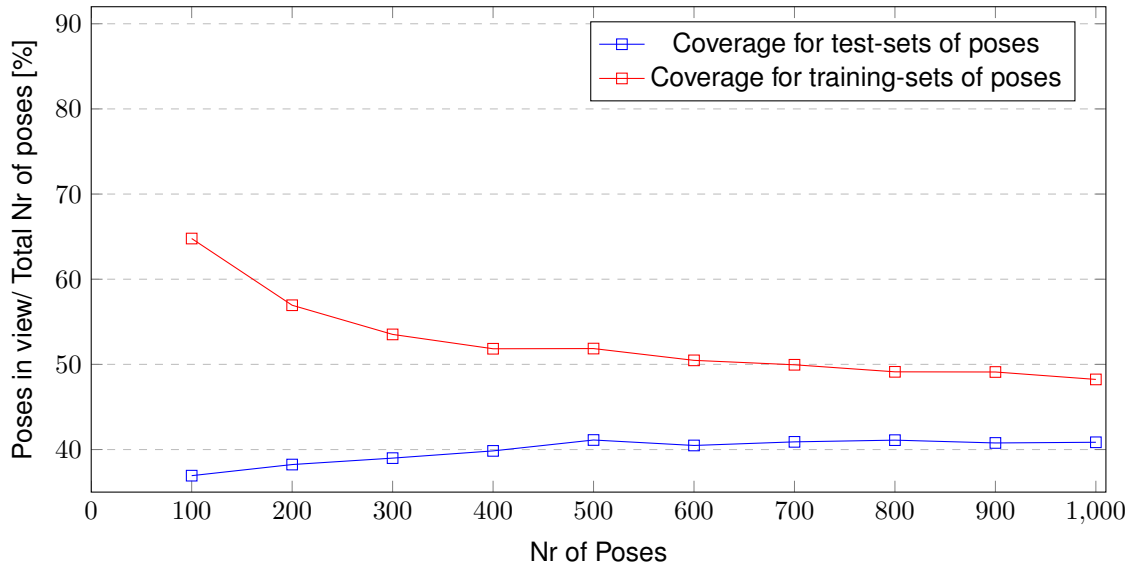


(b)

Figure 5.10: Cross-Coverage relative error per number of poses, obtained for 10 landmarks in Room 1 (a) and Room 2 (b).



(a)



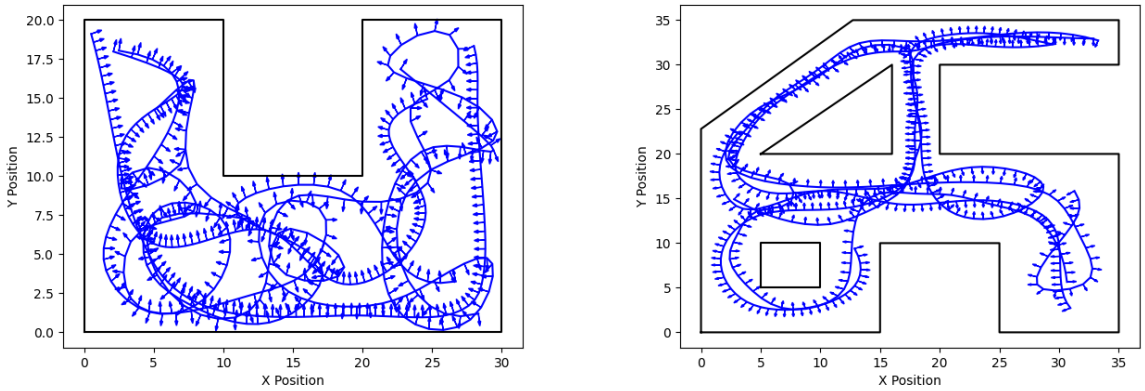
(b)

Figure 5.11: Comparison between the average coverage obtained in training and test condition for a fixed number of landmarks $L = 10$ in Room 1. (a) and Room 2 (b).

Trajectories

Aiming to understand the behavior of the method when the witness poses are not randomly generated, but with a plausible meaning, a similar analysis is now made for sets of poses obtained by sampling realistic trajectories.

Figures 5.12 and 5.13 present a set of predefined trajectories and their respective Pareto frontiers.



(a) Room 1 (b) Room 2

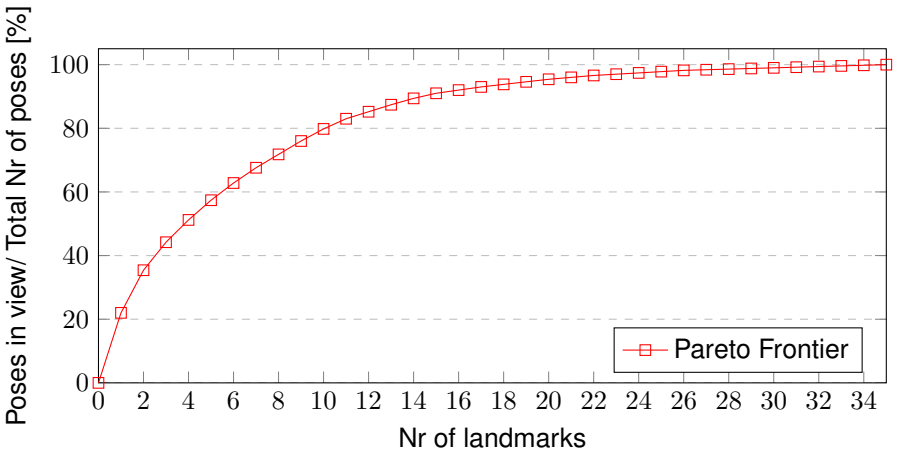
Figure 5.12: Considered Trajectories.

According to the results shown in figure 5.14 a set of landmarks optimized for random poses obtains better results at trajectories than a set of landmarks optimized for trajectories obtains when crossed with random witness sets. It is possible to conclude that a random poses-based solution is more robust than a trajectory-based solution obtaining better coverage of the polygon in general. Which is perfectly understandable once the random poses are uniformly scattered in the polygon. However, a trajectory-based optimization is more appropriated when one has a family of similar paths that for instance, must be traversed by a robot.

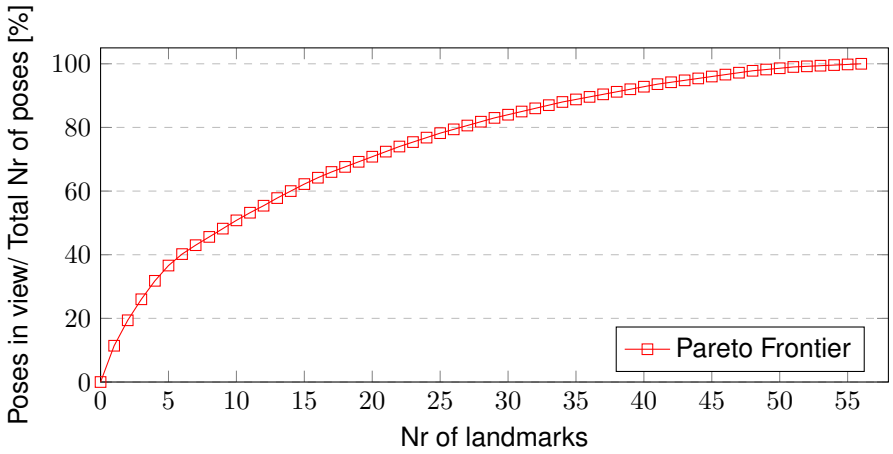
5.1 Pointwise vs non-Pointwise

As expected a non-pointwise obtains less coverage than a pointwise approach, as shown in figures 5.16 and 5.17. Figure 5.18 shows the mean processing time, for both Pointwise and non-Pointwise algorithms as a function of the number of poses. Similar to the theoretical calculations, the results suggest that the non-pointwise algorithm has a higher time complexity than the pointwise version.

The asymptotic behavior of the mean pre-processing time of the Pointwise algorithm suggests a polynomial time complexity as it was theoretically estimated in Chapter 4. The result also suggests that the non-Pointwise algorithm has a polynomial time complexity, however with higher order. Note that for smaller numbers of poses there are only slight differences in the mean pre-processing time of the two methods, this similarity is mainly due to the reduction of the distance between consecutive endpoints of the visible intervals in each edge of the polygon. To better understand this phenom, is important to remind the reader how the non-Pointwise algorithm computes the candidate intervals: a given portion of



(a)



(b)

Figure 5.13: Pareto Frontier for a set of poses obtained from predefined trajectories in Room 1 (a) and Room 2 (b).

an edge of P is considered to be a candidate interval if its length is greater or equal to the length of the landmark. Thus, to build a candidate interval the algorithm search in an ordered list for two endpoints such that the distance between them is greater than or equal to the length of the landmark. The smaller is the average distance between endpoints more often the algorithm needs to search nearly the entire list of endpoints.

Although the number of poses has a direct influence on the computation time, the most important factor is the number of raw intervals and even the number of pre-processed intervals, which indicates an output-sensitive behavior. Figure 5.19 highlights the correlation between the size of the output and the pre-processing time.

According to the experiments, the cardinality of the list of candidate intervals is proportional to the number of poses, which suggests a linear space complexity of both Pointwise and non-pointwise algorithms. It should be noted that the pointwise algorithm generates more candidate intervals. The number of candidate intervals translates to a larger ILP in the optimization phase (even with fewer constraints) requiring computational capacity. Figure 5.21 shows the time percentage that both approaches spent in each phase of the method.

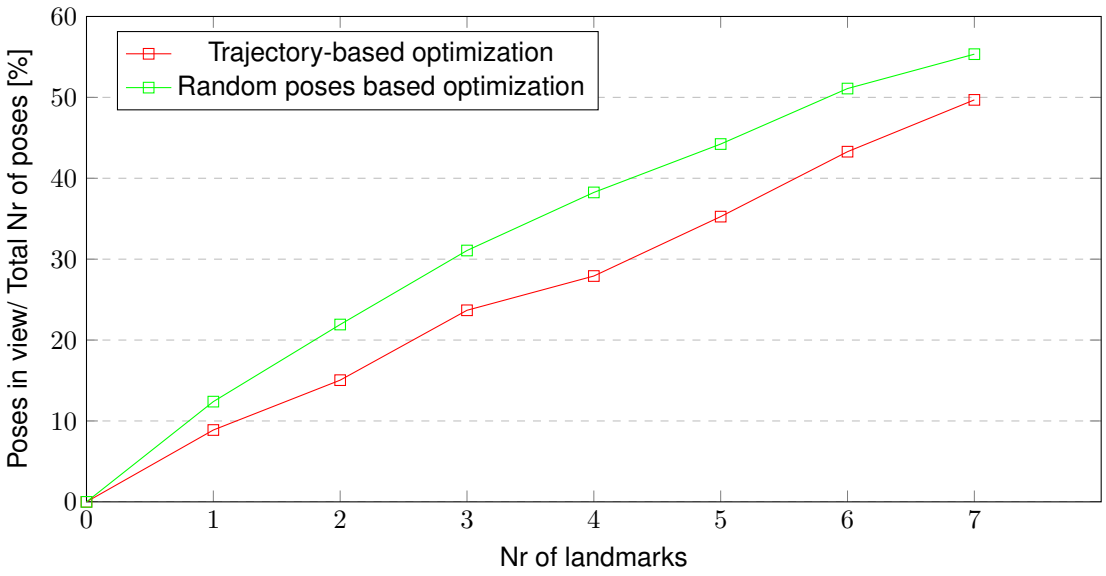


Figure 5.14: Room 1: Average coverage obtained in test conditions considering random poses and trajectories.

When considered solely the Visibility Intersection Algorithms, the pointwise algorithm is faster than the non-pointwise version, as it takes fewer computations. On the other hand, the Reading time is correlated with the size of the output of the Visibility Intersection Algorithms.

5.2 Comparison with meta-heuristic algorithms

Figure 5.22 presents the mean coverage of three meta-heuristics algorithms: Simulated Annealing; Particle Swarm (PS) and a Genetic Algorithm (GA). The results were obtained for benchmark sets of poses. Due to the complexity to integrate the non-Pointwise landmark approach with the straightforward meta-heuristic algorithms, a comparison of the non-Pointwise version of the four algorithms was not performed.

Based on the results of this analysis, the method developed in this thesis (VIA+ ILP) was shown to have better results than the meta-heuristic algorithms considered in the experiments. To achieve $\approx 80\%$ of coverage, the VIA + ILP method requires 8 landmarks. To achieve the same level of coverage the PS and GA require 12 landmarks while the SA requires at least 15. By searching a range of possible solution points, meta-heuristic methods generally can find ”good” solutions in a relatively short time frame when for instance compared with an exhaustive search algorithm, however, the optimality of the solutions is not guaranteed, particularly for problems with non-smooth objective function with several local optimal. This disadvantage is perceptible when for a greater number of landmarks, a lower coverage is obtained than that found for a certain lower number of landmarks, showing the non-optimality of the method.

From the meta-heuristic methods evaluated in the experiments, the GA is shown to have better results for the landmark placement problem, as for the same period of time it can find a set of landmarks that cover a higher percentage of the polygon. Figure 5.23 shows the mean processing time of the GA solver and the VIA + ILP method.

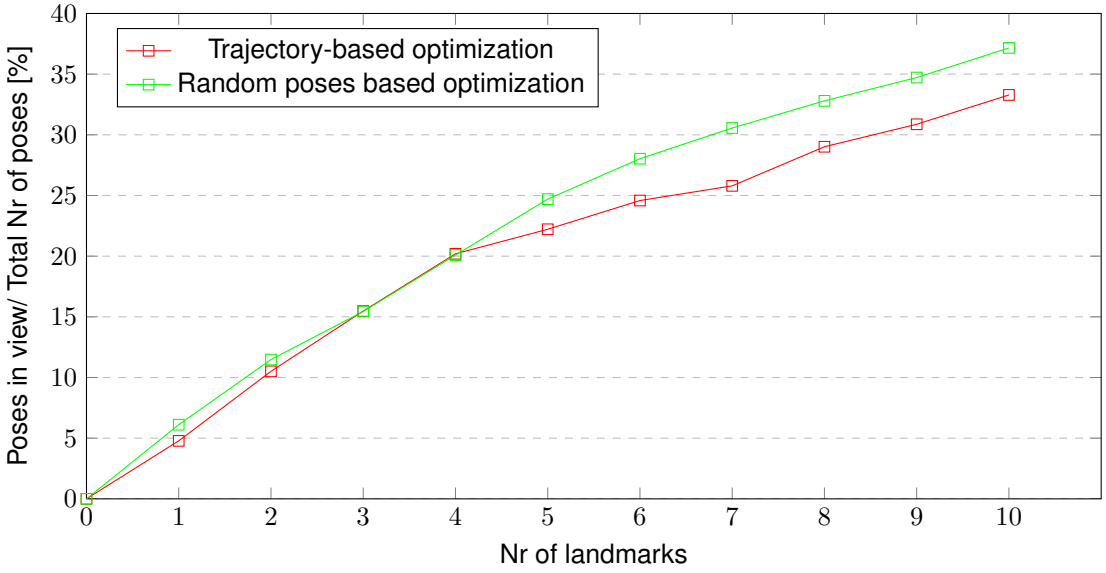


Figure 5.15: Room 2: Average coverage obtained in test conditions considering random poses and trajectories.

The results of this section reinforce the choice of an ILP approach instead of a straightforward optimization with meta-heuristic algorithms or even exhaustive search algorithms. It is clear an exponential asymptotic behavior of the GA. For the VIA + ILP, the number of landmarks does not even affect the optimization time which only depends on the number of variables and constraints of the ILP.

5.3 Coverage and redundancy assessment

Figure 4.2 presents the results of the coverage assessment performed in both polygons. One of the main goals of this thesis is to study a possible trade-off between the number of landmarks and the coverage. The problem formulation does not directly consider the installation cost of the landmarks within the environment. However, the installation cost can be interpreted as proportional to the number of landmarks. For a trade-off analysis, it is suggested a multi-criteria optimization problem formulation, as presented in equation 3.8, where it is aimed to simultaneously minimize and maximize the number of landmarks and the coverage respectively. Since the optimal landmarks placement strongly depends on a sample set of poses, a statistical analysis is crucial. The following analysis is made based on the results presented in figures 5.3 and 5.4.

To achieve a trade-off, for instance, one can define a minimum required coverage percentage and a threshold value for the gain. Thus the first number of landmarks that have an average gain below the threshold value and an average coverage above the required coverage is selected to achieve a trade-off between the landmarks installation cost and the percentage of covered area.

Figures 5.24 and 5.25 show the gain, $G(L)$ that is obtained as a function of the number of landmarks in both environments. For the experimental results let us consider a *Gain threshold value* $g_t = 2.5\%$ and a minimum required coverage $C_{Req} = 75\%$. Note that for $R = 4.4[m]$ it is not possible to satisfy the minimum required coverage of 75% with $M \leq 20$, thus for experimental purposes it is selected $M = 20$

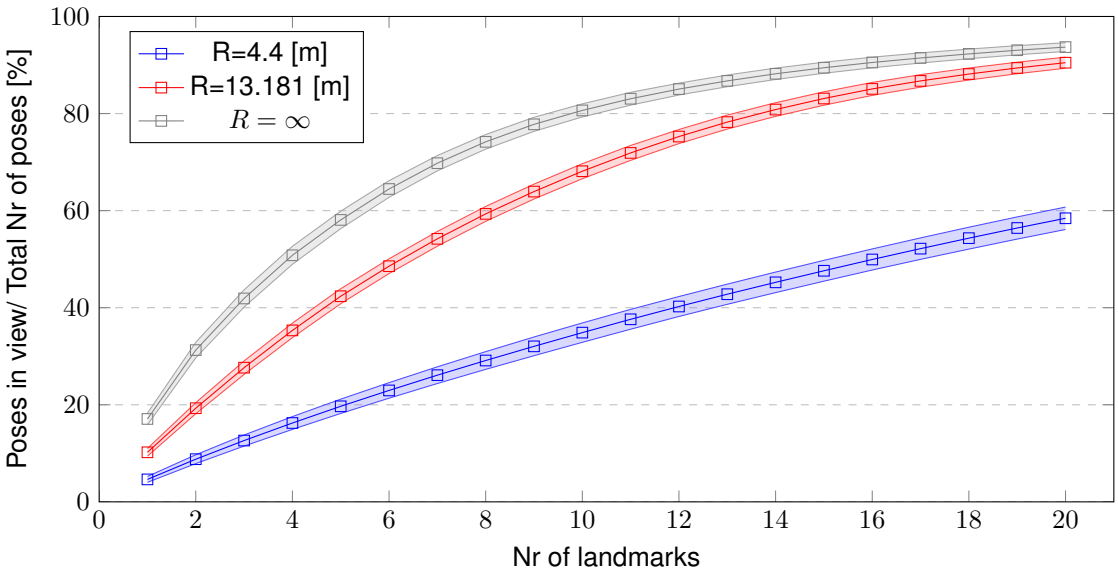


Figure 5.16: Coverage per number of landmarks in Room 1 for different maximum detection range and a non-pointwise landmark with length $D_l = 1[m]$.

landmarks.

		M	Gain [%]	Average Coverage [%]
Room 1	$R = 4.4[m]$	-	-	-
	$R = 13.181[m]$	14	2.46	86.34
	$R = \infty$	11	2.34	87.09
Room 2	$R = 4.4[m]$	-	-	-
	$R = 13.181[m]$	19	1.95	76.39
	$R = \infty$	14	2.22	76.77

Table 5.1: Selection of M for each maximum range of detection. Based on the results of figures 5.3 and 5.24 is selected the smaller M for which is obtained an average coverage above 75% and an average gain below 2.5%.

Since the objective function does not address redundancy, one can have an insight into redundancy by considering the percentage of witnesses poses covered by more than one landmark. Figure 5.26 shows that redundancy tends to increase with the coverage and the number of landmarks.

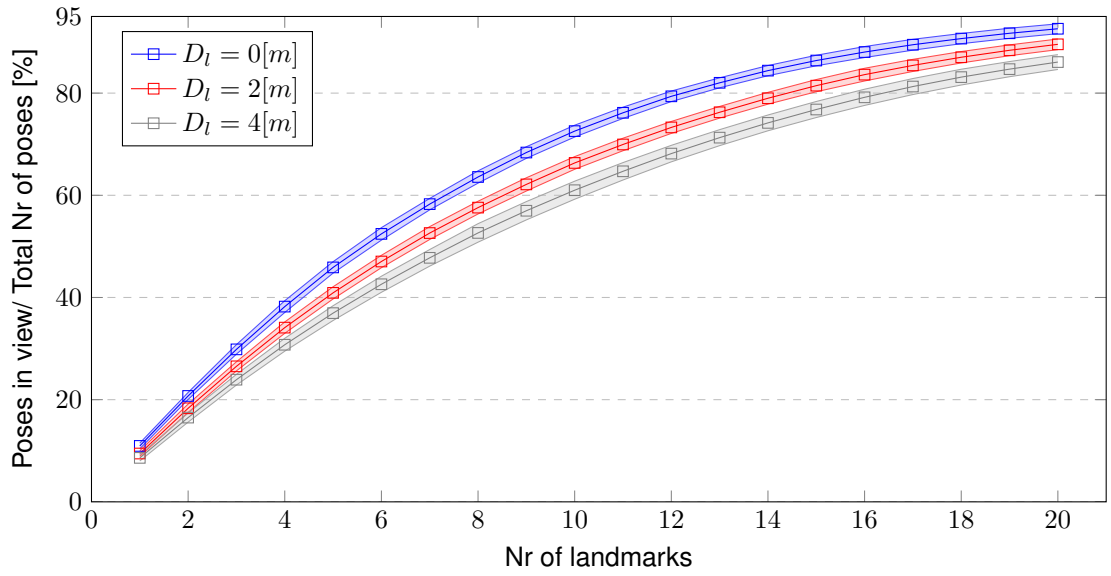


Figure 5.17: Room 1: Maximum Coverage for different landmark lengths, obtained for 500 poses with $R = 13.181m$.

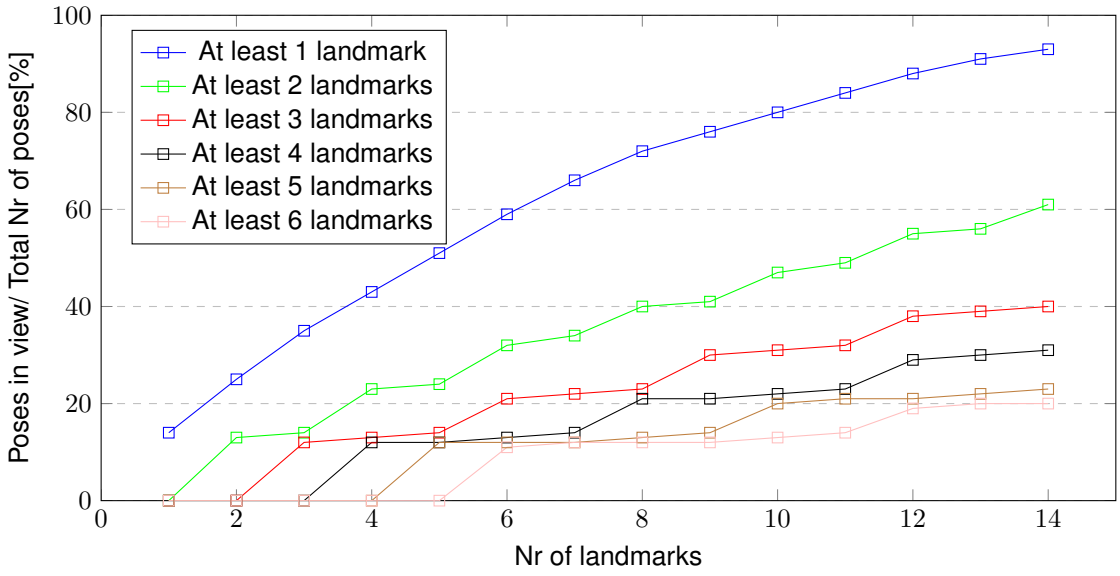


Figure 5.26: Room 1: Coverage for more than one landmark, considering 500 poses with $R = 13.181m$.

5.4 Discussion

The main objective of this thesis is to present a global optimization method for the landmarks placement problem in a given environment. However, numerical and simulations results are not sufficient to prove the optimality of the method, one can have significant insights and indicators that reinforce the theory on the optimality of the method. Especially the results presented in figures 4.3 and 4.4 where the same method developed in this thesis to optimize the landmarks placement is used as an approximation method of the AGP. For the two polygons considered, the method respects the theoretical bound on the number of guards that are sufficient to cover both simple and polygon with holes. Another important

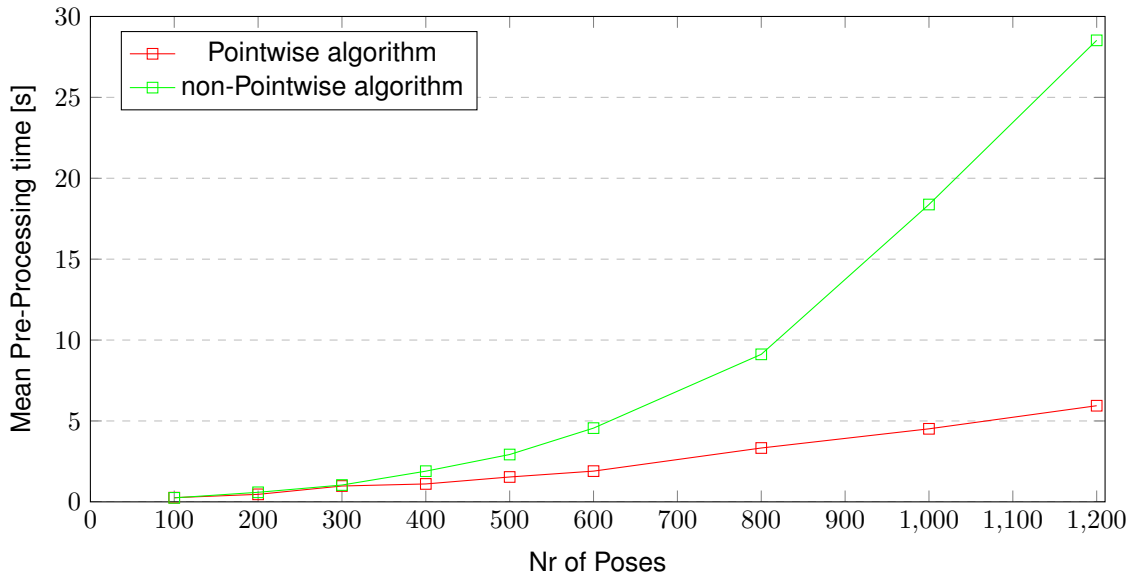


Figure 5.18: Room 1: Pre-Processing time as a function of the number of poses. In the experiments have been considered 10 landmarks.

result is the shape of the Pareto frontier and curve of coverage as a function of the number of landmarks in general, in our method (VIA + ILP), which is monotonically increasing. For a greater number of landmarks is always obtained a greater coverage percentage as one can see in figures 5.2 5.3 and 5.4. For instance, the heuristics algorithms considered in this thesis do not show the same monotone behavior, which is a clear indicator of non-optimality.

The training/test results display that as greater the cardinality of the witness set, the greater the robustness of the method, especially, for a random set of poses. Since the landmarks placement optimization based on a random witness set is only plausible under the assumption that the witness set is representative of the polygon or of a certain region of interest that is aimed to be covered. The coverage obtained in training and testing sets of poses tend to converge as the number of poses increases. In fact, the number of poses is crucial for the optimization problem. Not only it can penalize the computational time if arbitrary large, but also can affect the robustness of the solution if it is considered a small number of poses. Also, the standard deviation of the maximum coverage of a random witness set decreases with the number of poses.

The results provide important information to address a trade-off between the number of landmarks and the coverage. Especially, the analysis of the Pareto frontier's shape and the behavior of the incremental gain (see figures 5.24 and 5.25). However, it is important to have in mind that several factors may affect the shape of the curve including the number of poses, the area of the polygon, the field-of-view and the detection range. And the selection of the number of landmarks that better suit a trade-off with the coverage must concern all these factors.

The implemented method shows to be faster than the heuristic algorithms, especially greater numbers of landmarks. However, the optimization is performed offline, which makes the time complexity difference not as relevant as the difference in coverage percentage. Regarding coverage, our method always shows better results, for a certain number of landmarks, our method cover approximately 20%

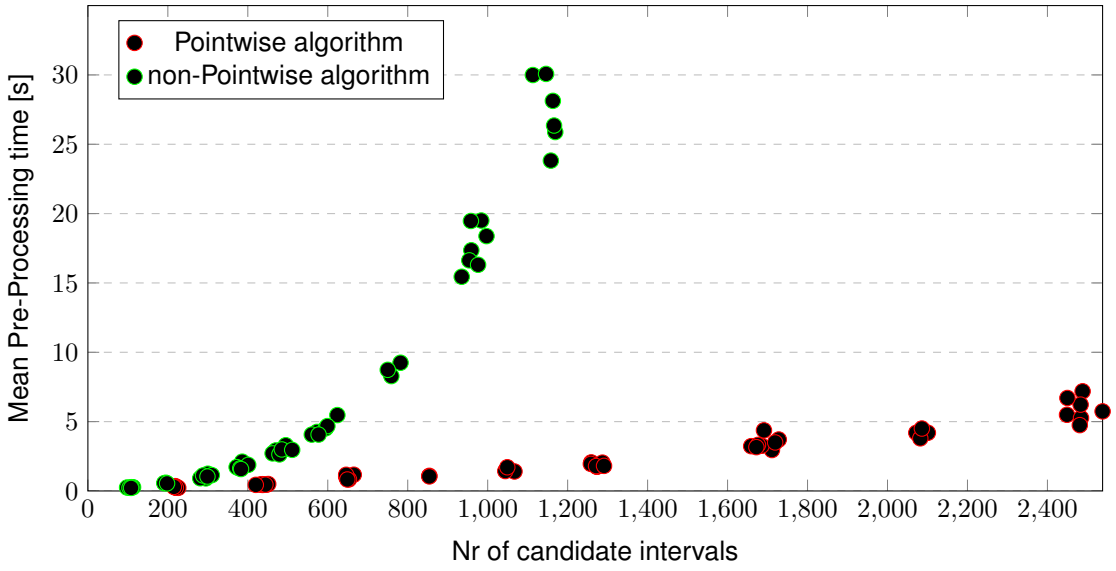


Figure 5.19: Room 1: Scatter plot of mean processing time per number of candidate intervals, have been considered 10 landmarks.

and 8% more poses than the best meta-heuristic algorithm in Room 1 and Room 2 respectively.

As expected, pointwise landmarks can cover a higher percentage of poses when compared with non-pointwise landmarks. this difference is explained by constraints on the length of the candidate intervals since candidate intervals with lengths shorter than the landmark are eliminated in the pre-processing phase. In terms of time complexity, the results show a higher complexity of the non-pointwise algorithm which is aligned with the theoretical results.

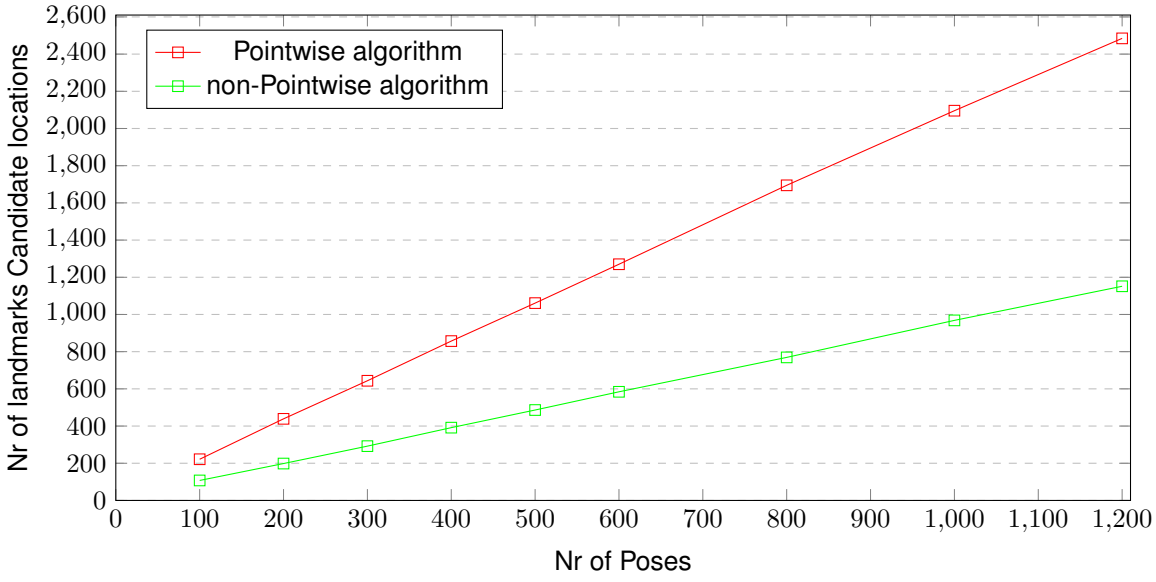


Figure 5.20: Room 1: Relation between the number of poses and the number of candidate intervals.

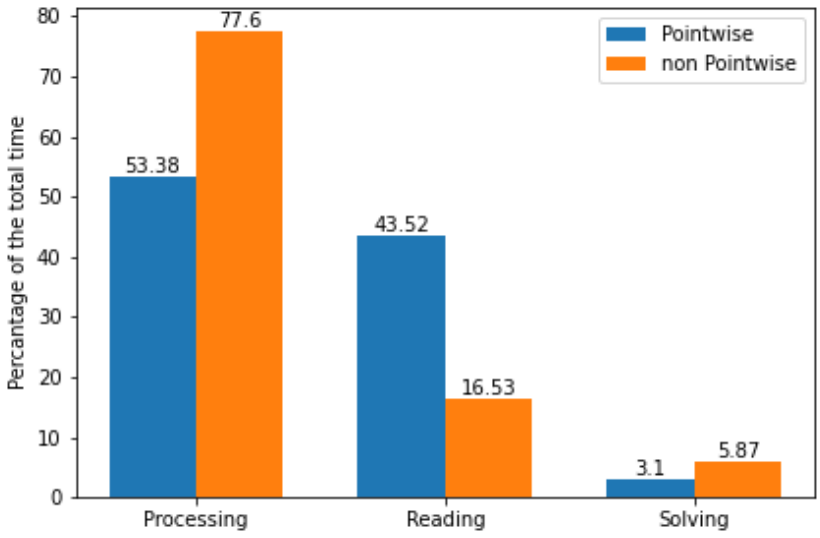
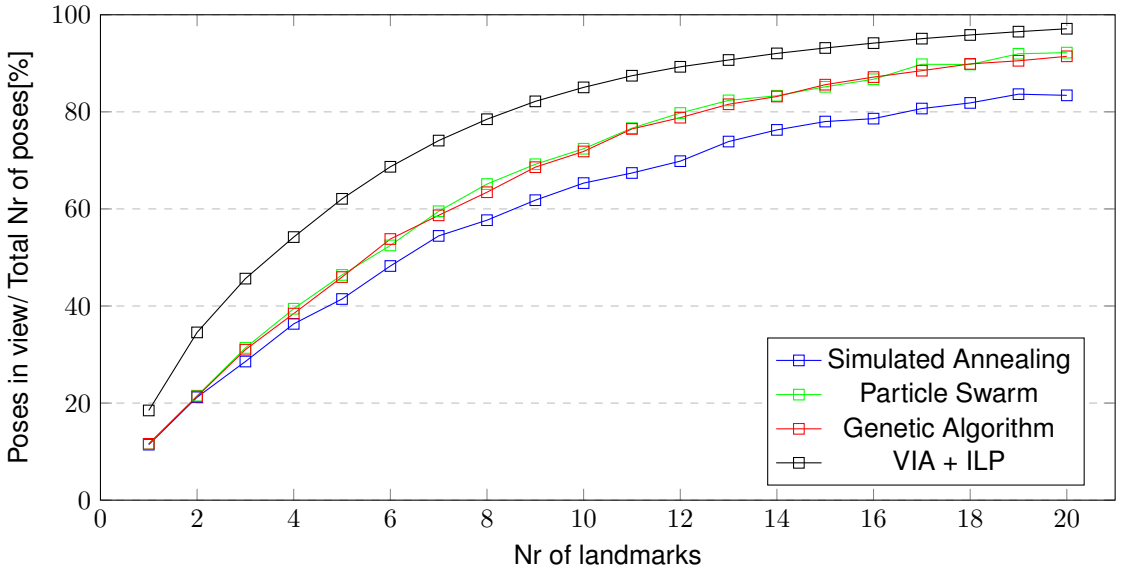
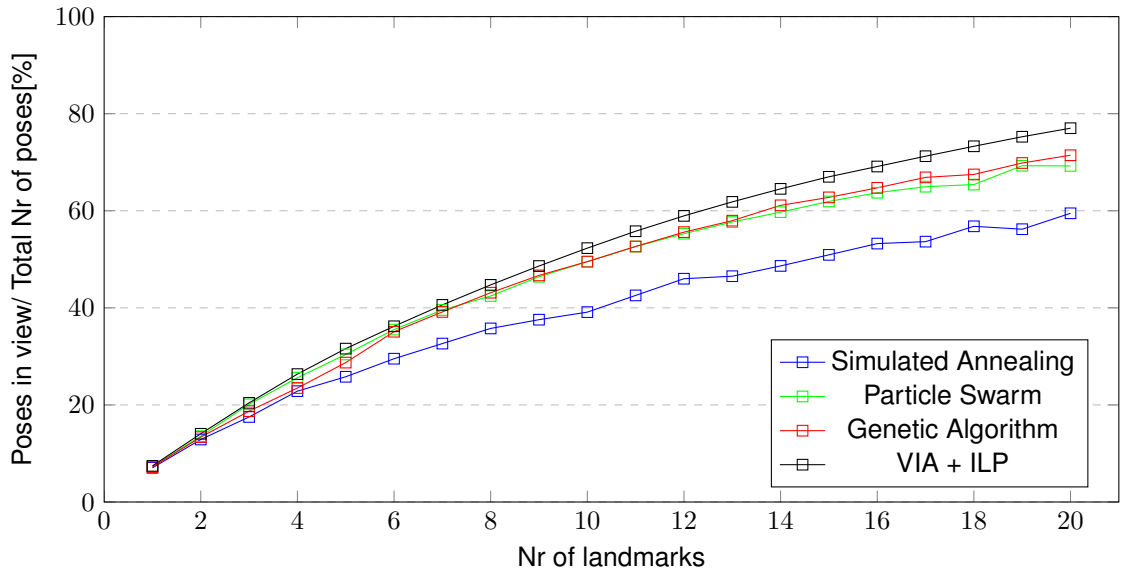


Figure 5.21: The contribution percentage of the pre-processing phase (implemented by the VIA) and the optimization phase for the total elapsed time. The Reading is the process of loading the parameters to build the ILP, the reading time is correlated with the size of the input of the solver. The optimization time is the sum of the parameters reading time and the solving time. The results have been obtained from 100 runs, where each run considers 500 random poses with a detection range $R = 13.181m$.



(a)



(b)

Figure 5.22: Mean Coverage comparison with meta-heuristics algorithms for Room 1 (a) and Room 2 (b).

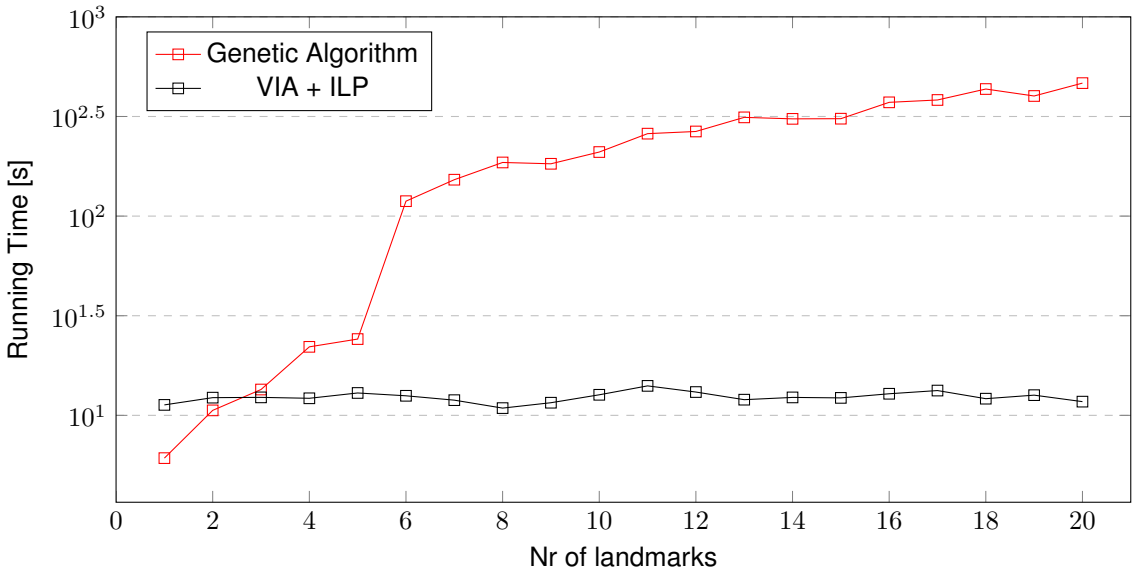


Figure 5.23: Average elapsed time comparison with the Genetic Algorithm for 500 poses with $R = 13.181m$ in Room 1.

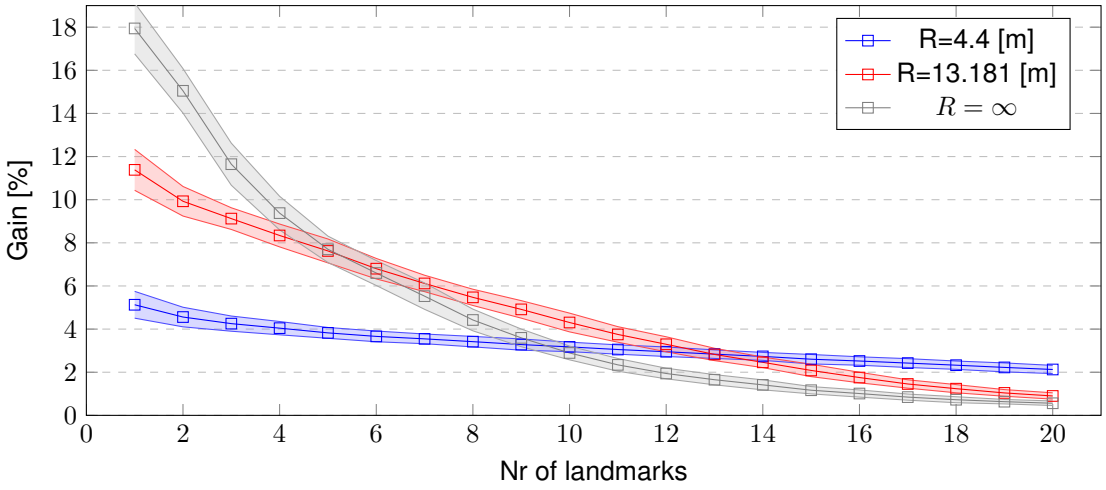


Figure 5.24: Room 1: Difference of coverage obtained for L and $L + 1$.

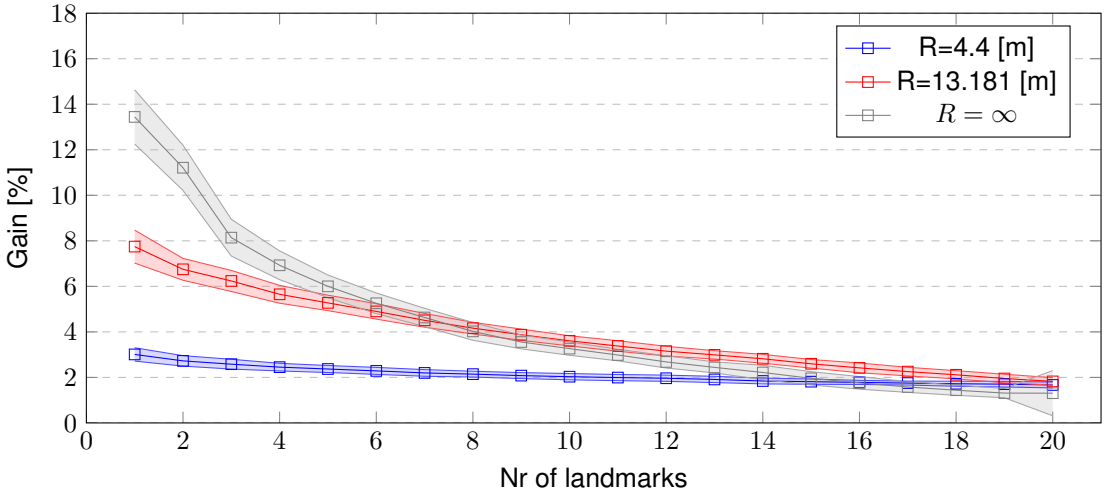


Figure 5.25: Room 2: Difference of coverage obtained for L and $L + 1$.

Chapter 6

Conclusions

This thesis presents an optimization method for the placement of robot navigation landmarks. Firstly, it is solved a relaxed problem assuming pointwise landmarks and evolving to a realistic approach considering non-pointwise landmarks. The method relies on a simulation of possible robot poses that are believed to be representative of the environment. The designed method consists of a pre-processing phase and an optimization phase. The pre-processing phase is implemented by pointwise and non-pointwise polynomial-time algorithms that compute a set of candidate intervals where a landmark can be placed. Then, the problem is formulated as an ILP and a standard ILP solver is selected to optimize the landmarks placement in the environment. It is studied a trade-off between the number (or cost of installation) of landmarks and the coverage required since there is an interest in minimizing the number of landmarks due to the system cost and computation capability that is necessary to process a large number of landmarks. In addition, the problem was formulated as a Multi-Criteria optimization problem to discuss criteria for a trade-off between the number of landmarks installed and the coverage, based on a Pareto frontier analysis. The implemented method was compared with meta-heuristics solvers including Particle Swarm, Simulated Annealing, and Genetic Algorithm.

An ILP solver guarantees global optimality in contrast with metaheuristic algorithms. Moreover, this significantly reduces the optimization time, since an ILP solver with an appropriate formulation can yield solutions in much less time than the worst-case scenario. It is important to mention that the obtained solution is optimal for the selected set of poses, which is an approximation of the continuous coverage problem similar to the NP-hard *Art Gallery Problem*, with the additions of a limited range and field-of-view for the sensor.

Determining the ideal pose set is one of the primal challenges of this approach, which on its own is an interesting topic for future works. The difference between in coverage obtained in training and testing conditions tends to decrease with the number of poses, which reinforces that the landmark placement method tends to be more robust when considered a larger set of poses.

Considering an infinite range and unlimited field-of-view for the sensors reduces the landmarks placement problem to the AGP. Under these conditions, in several simulations performed, the method always yields solutions with fewer guards (landmarks) than the theoretical upper bounds established by the *Art*

Gallery Theorems. This experiment was used as a validation of the method.

According to the results, a non-pointwise approach yields slightly less coverage than the relaxed pointwise version, as expected. However, the complexity of the pre-processing algorithm has greater order than a non-pointwise approach.

Finally, one can conclude that the presented method can be used as an optimization tool in applications including indoor localization and indoor advertising. In the latter application, the non-pointwise landmarks approach presented can offer an insight into the optimal placement of the advertising devices to reach as many people as possible.

6.1 Future Work

Many aspects of this thesis remain as open issues, thus, there is much room for future improvements, such as determining the ideal set of poses according to the environment and features of the sensor systems, improving the time complexity of the pre-processing algorithms especially the non-pointwise version and a more detailed study on the advantages of considering a non-pointwise landmark for applications other than robot localization.

Future works should include aspects like uncertainty on the identification of the landmarks and redundancy. In a landmarks-based localization method, it is important to ensure that a detected landmark is distinguishable from other landmarks. The presented method, for instance, does not consider a separation distance between two landmarks which may lead to data association errors, such problems could be addressed by including more features of the landmarks such as colors, shape, probability of a landmark identification error, and among others aspects. Although redundancy can be addressed with the presented method, it is important to directly include redundancy in the objective function. The results presented in this thesis were obtained through simulation. Hence, future works need to include practical experiments, taking into consideration aspects like the sensor uncertainty, error on the landmarks localization

Many other interesting possibilities have not been addressed in this thesis. One that stands out is the possibility to consider moving obstacles. Giving that there is traffic in a typical robot work-space whether it is people or other robots, a dynamic environment would be a better model than a static approach that is considered in this thesis. Another possibility is to deal with a 3D landmarks placement problem, which would bring new challenges especially to the pre-processing algorithm and in this case, more sophisticated computer-graphic methods would be required to deal with 3D obstacles and compute the intersections zones in the walls.

Our method represents a significant advance compared to the work of Cordeiro [11] in terms of optimality and complexity. Although the problem is far from being considered closed, this project together with Cordeiro's work leaves strong foundations for a new and practical approach to address the problem of placing landmarks that can be used in applications of great scientific and economic potential.

Bibliography

[1] A. Al-Kaff, D. Martín, F. García, A. de la Escalera, and J. María Armingol. Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*, 92: 447 – 463, 2018. ISSN 0957-4174.

[2] N. Metni and T. Hamel. A uav for bridge inspection: Visual servoing control law with orientation limits. *Automation in Construction*, 17(1):3 – 10, 2007. ISSN 0926-5805.

[3] M. Burri, J. Nikolic, C. Hürzeler, G. Caprari, and R. Siegwart. Aerial service robots for visual inspection of thermal power plant boiler systems. In *2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, pages 70–75, Sep. 2012.

[4] M. Stokkeland, K. Klausen, and T. A. Johansen. Autonomous visual navigation of unmanned aerial vehicle for wind turbine inspection. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 998–1007, June 2015.

[5] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme. Robust localization using relative and absolute position estimates. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*, volume 2, pages 1134–1140 vol.2, Oct 1999.

[6] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *J. Field Robotics*, 14:231–249, 1997.

[7] G. Dedes and A. G. Dempster. Indoor gps positioning - challenges and opportunities. In *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.*, volume 1, pages 412–415, Sep. 2005.

[8] Z. Song, G. Jiang, and C. Huang. A survey on indoor positioning technologies. In Q. Zhou, editor, *Theoretical and Mathematical Foundations of Computer Science*, pages 198–206, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-24999-0.

[9] F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21(3):2568–2599, 2019.

[10] P. Nazemzadeh, D. Fontanelli, and D. Macii. Optimal placement of landmarks for indoor localization

using sensors with a limited range. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Oct 2016.

[11] J. R. Cordeiro. Optimal visual markers placement for marker-based localization of uavs in indoor environments. Master's thesis, Universidade de Lisboa - Instituto Superior Técnico, 2019.

[12] W. Sutherland. *Introduction to Metric and Topological Spaces*. Open university set book. Clarendon Press, 1975. ISBN 9780198531616.

[13] S. K. Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, 2007.

[14] O. J. *Art Gallery Theorems and Algorithms*. International series of monographs on computer science. Oxford University Press, 1987. ISBN 9780195039658.

[15] J. O'Rourke. Galleries need fewer mobile guards: A variation on chvátal's theorem. *Geometriae Dedicata*, 14:273–283, 1983.

[16] T. C. Shermer. Recent results in art galleries (geometry). *Proceedings of the IEEE*, 80(9):1384–1399, Sep. 1992. ISSN 1558-2256.

[17] B. Joe and R. Simpson. Corrections to lee's visibility polygon algorithm. *BIT Numerical Mathematics*, 27:458–473, 1987.

[18] T. Asano. An efficient algorithm for finding the visibility polygon for a polygonal region with holes. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 68: 557–559, 1985.

[19] T. Asano, T. Asano, J. Hershberger, L. Guibas, and H. I. and. Visibility of disjoint polygons. *Algorithmica*, 1:49–63, 1986.

[20] P. J. Heffernan and J. S. B. Mitchell. An optimal algorithm for computing visibility in the plane. In F. Dehne, J.-R. Sack, and N. Santoro, editors, *Algorithms and Data Structures*, pages 437–448, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg. ISBN 978-3-540-47566-8.

[21] U. M. Erdem and S. Sclaroff. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Computer Vision and Image Understanding*, 103(3):156 – 169, 2006. ISSN 1077-3142. Special issue on Omnidirectional Vision and Camera Networks.

[22] S. Fisk. A short proof of chvátal's watchman theorem. *Journal of Combinatorial Theory, Series B*, 24(3):374, 1978. ISSN 0095-8956.

[23] J. Kahn, M. Klawe, and D. Kleitman. Traditional galleries require fewer watchmen. *SIAM Journal on Algebraic Discrete Methods*, 4(2):194–206, 1983.

[24] D. Avis and G. Toussaint. An efficient algorithm for decomposing a polygon into star-shaped polygons. *Pattern Recognition*, 13(6):395 – 398, 1981. ISSN 0031-3203.

[25] A. Kröller, T. Baumgartner, S. Fekete, and C. Schmidt. Exact solutions and bounds for general art gallery problems. *ACM J. Exp. Algorithmics*, 17, 2012.

[26] J. O'Rourke and K. Supowit. Some np-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29(2):181–190, March 1983. ISSN 1557-9654.

[27] D. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, March 1986. ISSN 1557-9654.

[28] R. Bar-Yehuda and S. Even. A linear-time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2(2):198 – 203, 1981. ISSN 0196-6774.

[29] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256 – 278, 1974. ISSN 0022-0000.

[30] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. ISSN 0364765X, 15265471.

[31] S. K. Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718 – 722, 2010. ISSN 0166-218X.

[32] E. Krohn and B. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66:564–594, 07 2013.

[33] A. Efrat and S. Har-Peled. Guarding galleries and terrains. *Information Processing Letters*, 100(6): 238 – 245, 2006. ISSN 0020-0190.

[34] K. Clarkson and K. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete and Computational Geometry*, 37, 02 2005.

[35] S. Eidenbenz. Approximation algorithms for terrain guarding. *Information Processing Letters*, 82 (2):99 – 105, 2002. ISSN 0020-0190.

[36] D. Tozoni, P. De Rezende, and C. Souza. The quest for optimal solutions for the art gallery problem: A practical iterative algorithm. volume 7933, pages 320–336, 06 2013. ISBN 978-3-642-38526-1.

[37] M. C. Couto, P. J. de Rezende, and C. C. de Souza. An exact algorithm for minimizing vertex guards on art galleries. *International Transactions in Operational Research*, 18(4):425–448, 2011.

[38] T. Baumgartner, S. Fekete, A. Kröller, and C. Schmidt. Exact solutions and bounds for general art gallery problems. volume 17, pages 11–22, 01 2010. ISBN 978-0-89871-931-4.

[39] A. Bottino and A. Laurentini. A nearly optimal algorithm for covering the interior of an art gallery. *Pattern Recognition*, 44(5):1048–1056, 2011. ISSN 0031-3203.

[40] Y. Amit, J. Mitchell, and E. Packer. Locating guards for visibility coverage of polygons. *Int. J. Comput. Geometry Appl.*, 20:601–630, 01 2010.

[41] S. Fekete, S. Friedrichs, A. Kröller, and C. Schmidt. Facets for art gallery problems. volume 73, 08 2013. ISBN 978-3-642-38767-8.

[42] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 2^{nd} edition, 2006. ISBN:978-0387303031.

[43] L. Wolsey and G. Nemhauser. *Integer and Combinatorial Optimization*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2014. ISBN 9781118626863.

[44] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998.

[45] H. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes. Field and service applications - an autonomous straddle carrier for movement of shipping containers - from research to operational autonomous systems. *IEEE Robotics Automation Magazine*, 14(3):14–23, 2007.

[46] H. Kim, J. W. Kim, and B. Jang. Indoor positioning system using sensor and crowdsourcing landmark map update. In *2019 International Conference on Green and Human Information Technology (ICGHIT)*, pages 7–11, Jan 2019.

[47] K. Guan, L. Ma, X. Tan, and S. Guo. Vision-based indoor localization approach based on surf and landmark. In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 655–659, Sep. 2016.

[48] X. Hou and T. Arslan. Monte carlo localization algorithm for indoor positioning using bluetooth low energy devices. In *2017 International Conference on Localization and GNSS (ICL-GNSS)*, pages 1–6, 2017.

[49] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.

[50] D. Macii, A. Colombo, P. Pivato, and D. Fontanelli. A data fusion technique for wireless ranging performance improvement. *IEEE Transactions on Instrumentation and Measurement*, 62(1):27–37, 2013.

[51] P. Nazemzadeh, D. Fontanelli, D. Macii, T. Rizano, and L. Palopoli. Design and performance analysis of an indoor position tracking technique for smart rollators. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–10, 2013.

[52] A. A. N. Shirehjini, A. Yassine, and S. Shirmohammadi. An rfid-based position and orientation measurement system for mobile objects in intelligent environments. *IEEE Transactions on Instrumentation and Measurement*, 61(6):1664–1675, 2012.

[53] R. Mautz and S. Tilch. Survey of optical indoor positioning systems. In *2011 International Conference on Indoor Positioning and Indoor Navigation*, pages 1–7, 2011.

[54] M. Beinhofer, J. Müller, and W. Burgard. Effective landmark placement for accurate and reliable mobile robot navigation. *Robotics and Autonomous Systems*, 61(10):1060–1069, 2013. ISSN 0921-8890. Selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011).

[55] J. Salas and J. L. Gordillo. Placing artificial visual landmarks in a mobile robot workspace. In H. Coelho, editor, *Progress in Artificial Intelligence — IBERAMIA 98*, pages 274–282, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-49795-0.

[56] P. Bessiere, J.-M. Ahuactzin, E.-G. Talbi, and E. Mazer. The "ariadne's clew" algorithm: global planning with local methods. In *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '93)*, volume 2, pages 1373–1380 vol.2, 1993.

[57] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, page 816–822. AAAI Press, 1992. ISBN 0262510634.

[58] L. Erickson and S. LaValle. An art gallery approach to ensuring that landmarks are distinguishable. 06 2011.

[59] K. Sundar, S. Srinivasan, S. Misra, S. Rathinam, and R. Sharma. Landmark placement for localization in a gps-denied environment. In *2018 Annual American Control Conference (ACC)*, pages 2769–2775, 2018.

[60] T. Rupp and P. Levi. Optimized landmark arrangement for absolute localization-a practical approach. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 1, pages 448–453 vol.1, 2000.

[61] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47: 2280–2292, 06 2014.

[62] R. Paschotta. *Encyclopedia of Laser Physics and Technology*. Wiley, 2008. ISBN 9783527408283.

[63] J. Greivenkamp. *Field Guide to Geometrical Optics*. Field Guide Series. Society of Photo Optical, 2004. ISBN 9780819452948.

[64] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In K. Jansen, S. Khanna, J. D. P. Rolim, and D. Ron, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 72–83, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-27821-4.

[65] N. Thakoor, V. Devarajan, and J. Gao. Computation complexity of branch-and-bound model selection. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1895–1900, Sep. 2009.

[66] J. Nievergelt and F. P. Preparata. Plane-sweep algorithms for intersecting geometric figures. *Commun. ACM*, 25(10):739–747, Oct. 1982. ISSN 0001-0782.

[67] T. Koch. *Rapid Mathematical Prototyping*. PhD thesis, Technische Universität Berlin, 2004.

[68] K. J. Obermeyer and Contributors. VisiLibity: A c++ library for visibility computations in planar polygonal environments. `http://www.VisiLibity.org`, 2008. R-1.

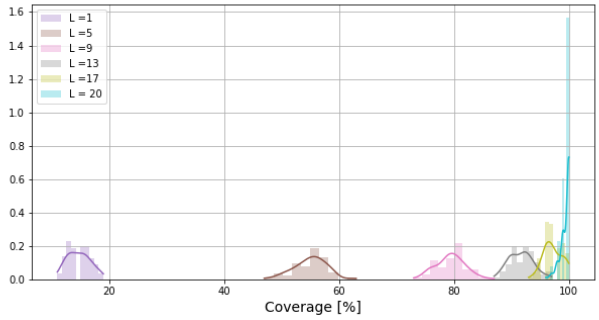
[69] J. Schmidt. Interval stabbing problems in small integer ranges. volume 5878, pages 163–172, 12 2009. ISBN 978-3-642-10630-9.

[70] H. Imai and T. Asano. Dynamic orthogonal segment intersection search. *Journal of Algorithms*, 8 (1):1–18, 1987. ISSN 0196-6774.

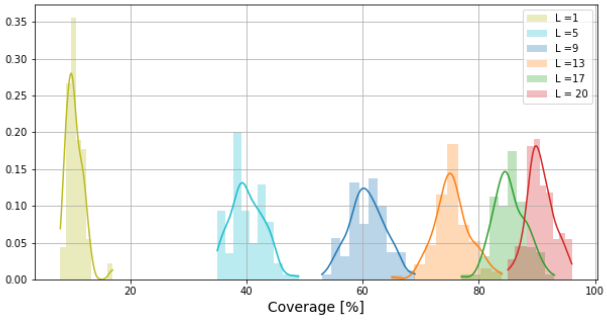
[71] H. Edelsbrunner. *Dynamic Data Structures for Orthogonal Intersection Queries*. Forschungsberichte / Inst. f. Informationsverarbeitung. Inst. f. Informationsverarbeitung, TU Graz, 1980.

Appendix A

Figures and Tables

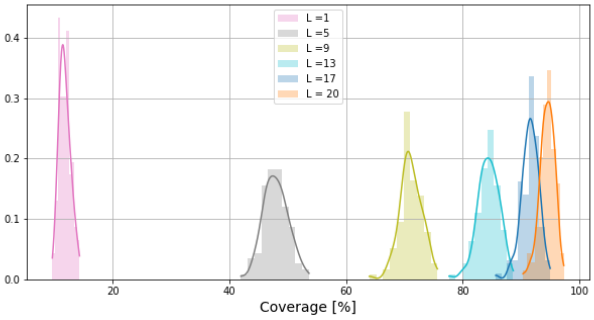


(a)

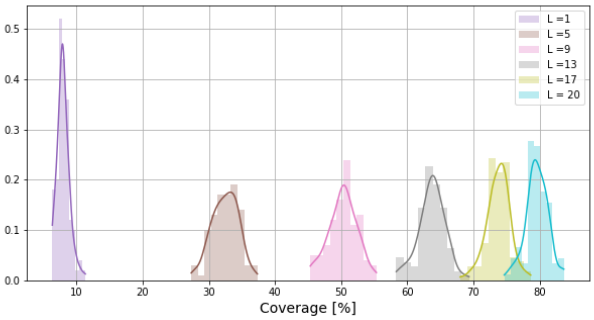


(b)

Figure A.1: Histograms of the Maximum Coverage percentage for different numbers of landmarks and the respective estimated PDF for 100 poses in Room 1 (a) and Room 2 (b).

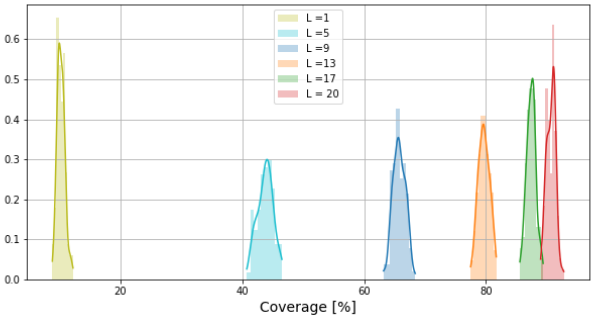


(a)

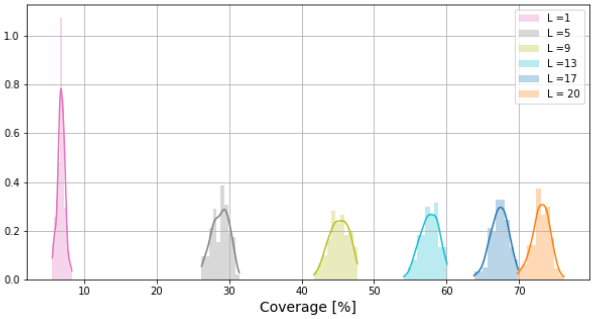


(b)

Figure A.2: Histograms of the Maximum Coverage percentage for different numbers of landmarks and the respective estimated PDF for 300 poses in Room 1 (a) and Room 2 (b).



(a)



(b)

Figure A.3: Histograms of the Maximum Coverage percentage for different numbers of landmarks and the respective estimated PDF for 1000 poses in Room 1 (a) and Room 2 (b).

	L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10	L=11	L=12	L=13	L=14	L=15	L=16	L=17	L=18	L=19	L=20
mean	5.1	9.7	13.9	18.0	21.8	25.5	29.0	32.4	35.7	38.9	41.9	44.9	47.7	50.5	53.1	55.6	58.0	60.3	62.6	64.7
std	0.6	1.0	1.2	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.0	2.1	2.2	2.2	2.2	2.3	2.3	2.3	2.3	2.3
min	3.9	7.4	10.9	14.4	17.9	21.3	24.7	27.8	30.8	33.8	36.9	39.6	42.4	44.7	47.1	49.4	51.8	54.1	56.5	58.4
25%	4.7	9.0	13.2	17.2	20.9	24.5	27.9	31.3	34.4	37.5	40.5	43.5	46.4	49.2	51.9	54.4	56.8	59.0	61.1	63.1
50%	5.0	9.6	13.9	17.8	21.6	25.3	28.8	32.3	35.6	38.7	41.9	44.9	47.7	50.4	52.9	55.3	57.7	60.1	62.2	64.5
75%	5.5	10.2	14.5	18.6	22.5	26.3	30.0	33.4	36.6	39.8	42.9	46.0	48.8	51.6	54.2	56.8	59.2	61.7	64.0	66.1
max	7.0	12.5	16.8	21.2	25.4	29.2	33.1	36.9	40.7	44.1	47.5	51.0	54.4	57.3	60.2	62.7	65.1	67.2	69.3	71.4

Table A.1: Coverage numerical results obtained in Room 1 with $R = 4.4m.$ for 500 poses. In this table, L is the number of landmarks.

	L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10	L=11	L=12	L=13	L=14	L=15	L=16	L=17	L=18	L=19	L=20
mean	11.4	21.3	30.4	38.8	46.4	53.2	59.3	64.8	69.7	74.0	77.8	81.1	83.9	86.3	88.4	90.2	91.6	92.9	93.9	94.8
std	0.9	1.3	1.3	1.5	1.6	1.6	1.6	1.5	1.4	1.3	1.2	1.2	1.2	1.1	1.1	1.0	0.9	0.9	0.9	0.8
min	9.6	18.6	27.0	35.2	42.6	49.6	55.8	61.6	66.6	71.0	74.6	77.8	80.4	83.0	85.6	87.6	89.2	90.6	91.8	92.8
25%	10.8	20.6	29.4	37.6	45.2	52.2	58.4	63.8	68.8	73.3	77.0	80.2	83.1	85.6	87.8	89.6	91.0	92.2	93.4	94.2
50%	11.2	21.2	30.4	38.8	46.6	53.2	59.2	64.8	69.8	74.0	77.8	81.2	84.0	86.4	88.4	90.2	91.6	93.0	94.0	94.8
75%	11.8	22.0	31.2	40.0	47.5	54.4	60.5	66.0	70.7	75.0	78.8	81.8	84.6	87.0	89.0	90.8	92.2	93.4	94.6	95.4
max	14.8	24.8	33.4	41.6	50.6	58.0	64.0	68.6	72.8	76.6	80.0	83.4	86.2	88.6	90.4	92.6	94.0	95.6	96.4	97.0

Table A.2: Coverage numerical results obtained in Room 1 with $R = 13.181m$ for 500 poses.

	L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10	L=11	L=12	L=13	L=14	L=15	L=16	L=17	L=18	L=19	L=20
mean	17.9	33.0	44.6	54.0	61.7	68.3	73.8	78.3	81.9	84.7	87.1	89.0	90.7	92.1	93.3	94.3	95.1	95.8	96.5	97.0
std	1.2	1.7	1.8	1.7	1.6	1.5	1.4	1.4	1.3	1.3	1.2	1.1	1.0	0.9	0.9	0.8	0.7	0.7	0.6	0.6
min	15.8	29.4	40.8	49.6	56.6	63.4	70.0	74.0	78.0	81.0	84.0	86.2	88.2	89.8	91.0	92.0	93.0	93.8	94.6	95.2
25%	17.0	31.8	43.6	53.0	60.6	67.3	72.8	77.4	81.2	84.1	86.4	88.4	90.2	91.6	92.8	93.8	94.6	95.4	96.0	96.6
50%	18.0	33.2	44.6	54.0	61.6	68.4	73.8	78.2	82.0	84.8	87.2	89.2	90.8	92.2	93.4	94.4	95.2	96.0	96.6	97.2
75%	18.6	34.1	45.8	55.2	62.8	69.4	74.8	79.2	82.6	85.6	87.8	89.8	91.4	92.8	94.0	94.8	95.6	96.4	97.0	97.4
max	22.0	37.4	48.6	58.6	66.2	71.4	76.8	81.6	85.0	87.4	89.6	91.6	93.0	94.2	95.2	96.2	97.0	97.6	98.2	98.6

Table A.3: Coverage numerical results obtained in Room 1 with $R = \infty$ for 500 poses.

	L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10	L=11	L=12	L=13	L=14	L=15	L=16	L=17	L=18	L=19	L=20
count	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
mean	3.0	5.7	8.3	10.8	13.1	15.4	17.6	19.8	21.8	23.9	25.9	27.8	29.7	31.6	33.4	35.2	36.9	38.7	40.4	42.0
std	0.3	0.4	0.6	0.7	0.8	0.8	0.9	1.0	1.0	1.1	1.1	1.2	1.2	1.3	1.3	1.3	1.4	1.4	1.4	1.5
min	2.5	4.7	7.0	9.2	11.5	13.6	15.8	17.9	19.8	21.8	23.7	25.7	27.4	29.2	31.0	32.7	34.4	36.0	37.7	39.4
25%	2.8	5.4	7.9	10.2	12.5	14.8	17.0	19.1	21.1	23.1	25.1	27.0	28.8	30.6	32.4	34.1	35.9	37.6	39.2	40.9
50%	3.0	5.7	8.3	10.8	13.1	15.4	17.6	19.7	21.8	23.9	26.0	27.9	29.8	31.6	33.3	35.2	36.9	38.6	40.4	42.0
75%	3.2	6.0	8.7	11.2	13.7	16.0	18.2	20.4	22.5	24.6	26.7	28.7	30.8	32.6	34.4	36.2	38.0	39.7	41.5	43.2
max	3.8	6.8	9.8	12.5	15.2	17.6	19.6	21.9	24.1	26.4	28.4	30.4	32.4	34.4	36.4	38.1	39.8	41.6	43.4	45.2

Table A.4: Coverage numerical results obtained in Room 2 with $R = 4.4m.$ for 500 poses. In this table, L is the number of landmarks.

	L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10	L=11	L=12	L=13	L=14	L=15	L=16	L=17	L=18	L=19	L=20
count	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
mean	7.7	14.5	20.7	26.4	31.7	36.6	41.1	45.2	49.1	52.7	56.1	59.2	62.2	65.1	67.6	70.1	72.3	74.4	76.4	78.2
std	0.7	1.0	1.3	1.4	1.6	1.7	1.7	1.7	1.8	1.8	1.7	1.7	1.6	1.6	1.6	1.5	1.5	1.4	1.4	1.3
min	6.4	12.2	17.8	23.0	27.8	32.6	37.2	41.8	45.4	48.8	52.2	55.4	58.6	61.6	64.2	66.6	69.2	71.4	73.6	75.4
25%	7.2	13.8	19.7	25.2	30.6	35.3	39.8	44.0	47.8	51.4	54.8	58.0	61.0	63.8	66.5	69.0	71.2	73.4	75.4	77.4
50%	7.6	14.4	20.8	26.4	31.6	36.4	40.8	45.2	49.2	52.8	56.0	59.2	62.2	65.0	67.6	70.0	72.2	74.4	76.2	78.0
75%	8.2	15.4	21.7	27.4	32.8	37.7	42.2	46.2	50.2	53.8	57.2	60.4	63.4	66.2	68.7	71.2	73.3	75.2	77.2	79.0
max	10.0	16.8	23.6	29.8	35.6	40.8	45.6	49.8	54.0	57.6	60.8	63.8	66.4	69.0	71.8	74.0	76.2	78.2	79.8	81.4

Table A.5: Coverage numerical results obtained in Room 2 with $R = 13.181m$ for 500 poses.

	L=1	L=2	L=3	L=4	L=5	L=6	L=7	L=8	L=9	L=10	L=11	L=12	L=13	L=14	L=15	L=16	L=17	L=18	L=19	L=20
count	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0	99.0
mean	13.4	24.7	32.8	39.7	45.7	51.0	55.6	59.6	63.2	66.4	69.4	72.1	74.6	76.8	78.7	80.5	82.1	83.5	84.8	86.1
std	1.2	1.6	1.7	1.8	1.9	1.9	1.9	1.9	1.9	1.9	1.8	1.8	1.8	1.7	1.7	1.7	1.6	1.6	1.6	1.4
min	11.2	21.6	29.2	35.0	40.4	45.6	50.0	54.4	58.0	61.2	64.0	66.6	68.8	71.0	73.2	75.2	77.0	77.4	77.4	81.8
25%	12.6	23.4	31.6	38.5	44.4	49.7	54.4	58.4	62.0	65.3	68.4	71.0	73.6	75.8	77.8	79.6	81.2	82.5	83.8	85.1
50%	13.4	24.6	32.6	39.6	45.6	50.8	55.4	59.4	63.0	66.2	69.2	72.0	74.2	76.6	78.6	80.4	82.0	83.6	85.0	86.4
75%	14.2	25.9	33.8	40.8	46.8	52.2	56.9	60.9	64.4	67.6	70.6	73.4	75.8	78.0	80.0	81.8	83.2	84.6	85.8	87.0
max	17.2	28.6	37.2	44.6	50.8	55.8	60.8	64.6	68.2	71.4	74.8	78.0	80.4	82.4	84.4	86.0	87.4	88.6	89.8	90.8

Table A.6: Coverage numerical results obtained in Room 2 with $R = \infty$ for 500 poses.