

Optimization of Landmark Placement for Robot Navigation

Délio Bengaló¹, Alberto Vale² and Rodrigo Ventura³

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 the efficient and reliable performance of the application. This paper presents a fast and global optimization method for landmarks placement in a polygonal environment, considering sensors with limited detection area. The developed method considers landmarks of finite length. Firstly, a relaxed problem is solved by assuming pointwise landmarks and evolving to a realistic approach considering non-pointwise landmarks. The 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 optimal placement problem is formulated as an *Integer Linear Programming* (ILP) and a globally optimal solution is obtained through a standard ILP solver. When constrained to the same execution time budget, our method obtains better coverage than a set of meta-heuristic algorithms considered in previous works.

I. INTRODUCTION

The rising demand for applications such as mobile robotics and indoor positioning systems has led to an increasing interest in indoor localization. Over the last few years, a lot of research has been made to design more reliable solutions for the indoor localization problem [1], [2], [3]. Therefore, solutions have been provided, from radars, laser sensors, ultrasonic sonars, infrared sensors, radio frequency identification (RFID) devices to vision-based systems. In [4] a survey of indoor localization systems and technologies is presented.

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 [5]. 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. An optimal landmarks placement increases the operational capabilities, reduces the risks and costs of the autonomous system.

The focus of this paper is the subsequent *Optimization Problem*, which consists of determining the optimal placement of a set of landmarks for a given environment. It is proposed a novel method to optimize the landmarks placement 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. The method relies on a simulation of possible agent poses that are believed to be representative of the environment or a family of robot

trajectories, and it involves two major steps: a pre-processing phase where a visibility algorithm computes a set of potential locations for the landmarks and an optimization phase where the problem is formulated as an Integer Linear Programming (ILP) and finally is used a standard ILP solver to obtain the optimal solution.

The novelty of this paper is the formulation that allows globally optimal solutions based on an ILP, in contrast with meta-heuristic approaches from which drive sub-optimal solutions. The problem formulation is made possible by the pre-processing algorithms. In addition, the method developed in this paper deals with landmarks with a finite length which to the best of our knowledge is barely addressed in the literature.

This paper is organized as follows: In Section II, we present a survey on related works. Section III is devoted to the basic definitions and notations. Section IV analytically formulates the problem. In Section V, we explain the method used to optimize the landmarks placement with a focus in the pre-processing phase highlighting the algorithms for both pointwise and non-pointwise landmarks. Section VI presents the most relevant results of the computational experiments and a comparison of our method and a set of meta-heuristic algorithms. In Section VII, we discuss and interpret the results. In the last section, we present some concluding remarks and suggestions for future works.

II. RELATED WORKS

A typical approach to solve the landmarks placement problem (LPP) is the AGP which is a classical problem in Computational Geometry (CG) where the goal is to determine the number of guards that are sufficient to oversee the interior region of a polygon P [6], [7]. The AGP has been proved NP-hard [8]. However several efficient approximation algorithms are available [9], [10], [11], [12]. In general, these algorithms are having a common principle which is to reduce the AGP to an instance of the *Set Cover Problems*. In [13] an algorithm based on the AGP to compute the position of visual landmarks is proposed. The objective is to maximize the area in which the robot has clear line sight to at least one landmark. A *Simulated Annealing* algorithm is proposed to determine the optimal set of landmarks. In [14] the problem of colored landmarks is introduced. With an AGP approach its ensured a visible landmark from any point of the map, but never two of the same color. Its established bounds on the number of colors needed.

Indeed, the main drawback of the traditional AGP-based approach is that in real-world applications most of the sensors are not omnidirectional and have a finite range. Therefore, some approaches as the ones presented in [15] and [16], combines the AGP with assumptions that are more consistent with the capabilities of real-world sensors. In [15] the AGP with fading (AGPF) is considered, where a polygonal region is to be illuminated with light sources such that every point is illuminated with at least a global

¹Délio Bengaló Instituto Superior Tecnico, Universidade de Lisboa, Avenida Rovisco Pais 1, 1049-001, Lisboa, Portugal, delio.bengalo@tecnico.ulisboa.pt

threshold, the light intensity decreases over distance, and it is aimed to minimize the total energy consumption. Two versions of the AGPF are studied. One is based on discrete approximation where the guards can be placed on a finite set of predefined locations. The other version allows guards that can be positioned anywhere in the input polygon and are based on non-linear programming utilizing simplex-partitioning strategies. In [16] a discrete instance of the AGP is proposed to solve the general camera placement problem in polygons that represent buildings floor plans. 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. Since there are considered realistic configurations of the cameras such as limited field-of-view (FoV), finite depth-of-field (DoF), and also resolution constraints this method is more suitable for real-world applications.

In [17] the problem of landmarks placement for mobile robots that carry out predefined navigation tasks, is addressed. The goal is to find the minimum number of landmarks for which a bound is guaranteed on the maximum deviation from the desired trajectory. The optimal sensor placement with limited detection area in ideally unbounded rooms is addressed in [5]. 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 [17] 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.

In summary, the method developed in this paper is based on an ILP formulation that does not resort to a sub-optimal heuristic like for instance, simulated annealing, in addition, our method considers occlusion due to obstacles, sensors with limited range, and landmarks with finite length.

III. PRELIMINARIES

In Robotics it is usual to model a complex real-world environment as a simplified 2D representation, especially regarding Automated Guided Vehicles (AGVs), where the robot maintains contact with the floor; and UAV applications where the vehicle keeps its altitude approximately constant. Thus, in this paper, the map of the environment is modeled as a *polygon* that is allowed to have *holes* caused by obstacles e.g. columns, walls, furniture. Hence, every curve surface is approximated by a set of line segments.

A polygon P is a closed and path-connected space in a Euclidean plane, 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 [18] [7].

A polygon is classified as a *Simple polygon* if it is a simply connected space. Thus, the boundary of the polygon consists

of only one cycle of edges; and a *Polygon with holes* if it is multiply connected. Thus, the boundary consists of two or more cycles of edges.

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 $ext(P)$.

$$P = \partial P \cup int(P) \quad (1)$$

Let P be a polygon with holes. The set of holes in P is denoted by $H(P) = \{P_1, \dots, P_h\}$ 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 , often referred as *shell* of P .

The set of vertices of an arbitrary polygon P is denoted by $\mathcal{V}(P) = \{v_1, \dots, v_n\}$ and the set of edges will be denoted by $E(P) = \{e_1, \dots, e_n\}$ where n is the number of vertices and edges of P , if P is a simple polygon (the boundary consist of only one cycle of edges) then an edge $e_i = \overline{v_i v_{i+1}}$ with $i = 1, \dots, n$ 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} belongs to the same cycle of edges. Two vertices v_i and v_{i+1} belongs 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 :

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

in other words: $e \in E(P)$.

Throughout this paper, passive landmarks installed on the edges of a given environment are considered. Two possible representations for a landmark l are considered:

- A point representation $l = \{(x, y)\}$, where the landmark is referred as *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, where the set is a singleton in the case of pointwise landmarks. The pointwise representation is a simplification of the problem, which understanding is crucial to unfold the non-pointwise approach.

Assumption 1. The landmarks are restricted to the boundary of P . Thus for a landmark l exists an edge e of P such that l is a subset of e :

$$\exists e \in E(P) : l \subseteq e \quad (2)$$

The landmarks are assumed to be readable by means of a sensor (e.g. cameras or RFID readers) 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 $\langle W \rangle = (O^W, X^W, Y^W)$. In this paper, the region within which a sensor can detect a landmark is referred to as the Sensor Detection Area (SDA) and is denoted by s [5].

Definition III.1. The sensor detection area is a convex and connected planar region within which a sensor can detect a landmark.

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 [19], [20]. In the case of optical sensors, the FoV is a solid angle through which the detector can sense electromagnetic radiation. However, this paper considers only the horizontal direction, since the problem at hand is two-dimensional in the horizontal plane. These assumptions are suitable for 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 $r = R/\cos(\theta)$.

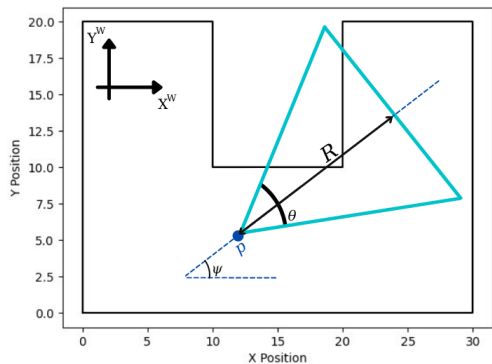


Fig. 1. Field-of-view and detection range of the sensors.

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:

$$0 < x \leq R \quad \wedge \quad \left| \frac{y}{x} \right| \leq \tan\left(\frac{\theta}{2}\right) \quad (3)$$

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

$$(\overline{qp} \subset P) \wedge (q \in s) \quad (4)$$

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 agent is a subset of the polygon; and (ii) q is inside the sensor detection area.

In Computational Geometry the portion of a polygon P , that is visible from a point p is called the *visibility polygon* of p and it is denoted by $V(p)$. However, only a subset of $V(p)$ satisfies the condition (4) which are the points of P that are inside the SDA s . From now on this intersection

region is referred to as the *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. If s is convex and connected then $v(p)$ is also star-shaped.

Definition III.2. 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 the *kernel* of P' .

Theorem III.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 2. A given landmark l is visible to a sensor located at the pose p with a SDA s if for every point $q \in l$ the condition (4) is satisfied.

IV. PROBLEM STATEMENT

This paper presents an ILP formulation to tackle 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 $\mathcal{W} = \{w_1, \dots, w_M\}$.

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

Definition IV.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 redundant¹ to place more than one landmark in the same candidate interval as they would be visible to the same set of poses.

Assumption 3. At most one landmark is placed at each candidate interval.

Consider $\mathcal{L} = \{l_1, \dots, l_N\}$, a set of candidate intervals where can be placed a landmark. The candidate interval l_j is associated with a nonempty 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:

$$\bigcup_{j=1}^N \mathcal{W}_j = \mathcal{W} \quad (5)$$

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

Definition IV.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 .

¹In this paper we assume that the number of landmarks is irrelevant to quality the localization.

This formulation resembles the *Set-Covering Problem* presented in [21] and [22]. The Set-Covering Problem is a classical combinatorial optimization problem, where given a ground set U and a collection of subsets $\mathcal{S} = \{S_1, \dots, S_N\}$ 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 :

$$\min |S'| \quad \text{s.t.} \quad \bigcup_{S_i \in S'} S_i = U \quad (6)$$

where $|S'|$ is the cardinality of the set S' .

In [22] 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 $S' \subseteq \mathcal{S}$ such that:

$$\max \left| \bigcup_{S_i \in S'} S_i \right| \quad \text{s.t.} \quad |S'| \leq k \quad (7)$$

Problems (6) and (7) 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 (6). From the conflicting objectives of maximizing coverage and minimizing the number of landmarks, arises a *multi-criteria optimization problem*, which 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 (7), transforming the maximum number of landmarks L in an objective function to be minimized.

Next, the mathematical formulation of the coverage problems (6) and (7) is presented. For sake of simplicity, it is first considered a pointwise and followed by an extension to a non-pointwise landmark.

1) *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 a $N \times 1$ 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 (6) is now formulated as:

$$\min \sum_{j=1}^N x_j \quad \text{s.t.} \quad \sum_{j=1}^N a_{ij} x_j \geq 1 \quad (8)$$

the M constraints in problem (8) 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 (7) in matrix notation, it is necessary to define y , a $M \times 1$ binary decision variable and its respective constraints:

$$\sum_{j=1}^N a_{ij} x_j \geq y_i \quad (9)$$

Hence, the number of poses covered by a given set of landmarks x is given by: $\sum_{i=1}^M y_i$.

Finally the problem (7) can be formulated in matrix notation as follows:

$$\max \sum_{i=1}^M y_i \quad \text{s.t.} \quad \begin{cases} \sum_{j=1}^N x_j \leq L \\ \sum_{j=1}^N a_{ij} x_j \geq y_i \end{cases} \quad (10)$$

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

2) *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 than 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.

Figure 2 illustrates a situation where exists a portion of the boundary of P (I_2) that 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 a landmark such that it is visible to both poses. Thus, 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 (10), 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 that 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.

Let us consider 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 $\|I_1 \cup I_2 \cup I_3\| \geq 2D_l$, which is a particular case of the following general result:

Theorem IV.1. Let I_a and I_b , be two overlapping candidate intervals, visible to pose a and b respectively, such that $\|I_a \cup I_b\| < 2D_l$. 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 .

Proof: If l_1 and l_2 are disjoint, i.e. $l_1 \cap l_2 = \emptyset$, then $\|I_a \setminus I_b\| \geq D_l$ and $\|I_b \setminus I_a\| \geq D_l$, thus the union of the two disjoint parts: $(I_a \setminus I_b) \cup (I_b \setminus I_a)$, has a length equal or greater than $2D_l$, which is a contradiction.

Since, for practical reasons, there can be no overlapping landmarks. From a pair of overlapping it is possible to select only one interval 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 which has a mutually exclusive relation with at least another interval in the graph; \mathcal{A} are the set of *edges*² of the graph, an edge of the graph (u, v) means that the candidate intervals with index u and v are mutually exclusive.

²Not to be confused with edges of a polygon

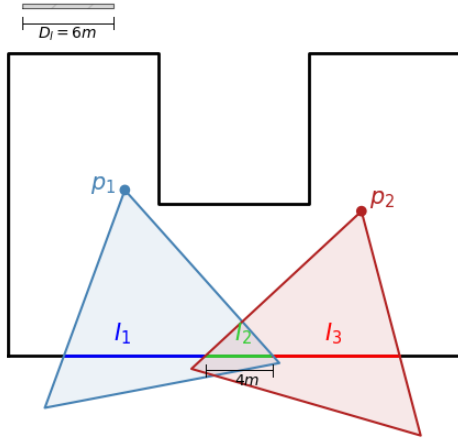


Fig. 2. The Intervals I_1 and I_3 are visible only for the poses p_1 and p_2 respectively. On the other hand, I_2 is visible from both poses, however, the length D_l of a landmark is greater than $\|I_2\|$.

Definition IV.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 (8) and (10) for each pair of mutually exclusive intervals. Thus, for a non-pointwise landmarks, the minimum set-cover problem is now formulated as follows:

$$\min \sum_{j=1}^N x_j \quad \text{subject to} \quad \begin{cases} \sum_{j=1}^N a_{ij} x_j \geq 1 \\ x_v + x_u \leq 1 \quad \forall (u, v) \in \mathcal{A} \end{cases} \quad (11)$$

and the maximum-coverage problem is formulated as follows:

$$\max \sum_{i=1}^M y_i \quad \text{s.t. subject to} \quad \begin{cases} \sum_{j=1}^N x_j \leq L \\ \sum_{j=1}^N a_{ij} x_j \geq y_i \\ x_v + x_u \leq 1 \quad \forall (u, v) \in \mathcal{A} \end{cases} \quad (12)$$

V. METHODOLOGY

The optimal LPP is usually intractable in the continuous domain. Thus, this paper presents a discrete formulation of the continuous problem. The formulation consists of considering a witnesses set $\mathcal{W} = \{w_1, \dots, w_M\}$, from which is computed a finite set $\mathcal{L} = \{l_1, \dots, l_N\}$ of candidate intervals where can be placed a landmark.

Next, we describe the two phases that constitute our method:

A. Pre-Processing

The Pre-Processing phase takes as input the polygon P and the witness set \mathcal{W} . And has as output a set $\mathcal{L} = \{l_1, \dots, l_N\}$ of candidate intervals, each one associated to a potential landmark; the set of subsets $\{\mathcal{W}_1, \dots, \mathcal{W}_N\} \subset 2^{\mathcal{W}}$, 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. From now on, this algorithm is referred to as *Visibility Intersection Algorithm* (VIA).

Algorithm 1 Candidate landmarks location Algorithm

- 1: Input : Polygon P with n vertices, witness set $\mathcal{W} = \{w_1, \dots, w_M\}$.
 - 2: Output : A list of candidate locations for the landmarks $\mathcal{L} = \{l_1, \dots, l_N\}$; $\{\mathcal{W}_1, \dots, \mathcal{W}_N\}$ and the constraint graph $\mathcal{G}(\mathcal{N}, \mathcal{A})$ of mutual exclusive intervals
 - 3: $\mathcal{E} = \emptyset$
 - 4: **for** $i = 1, \dots, M$ **do**
 - 5: Compute visibility polygon $V(w_i)$
 - 6: Compute $v(w_i) = V(w_i) \cap P$
 - 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** $k = 1, \dots, n$ **do**
 - 17: $\mathcal{L}_k = \text{VisIntersect}(k, \mathcal{E})$
 - 18: $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{L}_k$
 - 19: **end for**
-

First step is identify the edges of the limited-visibility polygon $v(w_i)$ that are also a subset of an edge of P for all $w_i \in \mathcal{W}$. 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 . A candidate interval is identified by a 3-tuple, $\varepsilon = (k, i, u)$ where $k = 1, \dots, n$, is the index of the edge e_k of P that contains the candidate interval, the second element is the index i of the pose w_i ; and 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} .

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 .

Following, we present the pseudo-code of two versions of the `VisIntersect` function for both pointwise and non-pointwise landmarks.

1) *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.

The fundamental principle of Algorithm (2) 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 all the 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 w_i . The intersection query is performed in $\mathcal{O}(M \log M)$ time by using an *interval tree* [23].

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

```

1: Input: Edge  $e \in E(P)$  and set  $\mathcal{E}$  of detectable
   portions of  $\partial P$ 
2: Output: List of potential landmarks in the edge  $e_k$ :
    $\mathcal{L}_k = \{l_1, \dots, l_{N'}\}$ 
3:  $\mathcal{I} = \emptyset$ 
4: for every candidate interval  $c$  indexed by  $\varepsilon = (k, i, u) \in \mathcal{E}$  do
5:   Normalize  $c$  to an interval  $\iota \subseteq [0, 1]$  such that the
   edge  $e_k$  goes from 0 to 1
6:    $\mathcal{I} \leftarrow \mathcal{I} \cup \{\iota\}$ 
7: end for
8: Build a balanced binary Interval Tree  $T(\mathcal{I})$  according to
   the value of the left endpoint of each interval
9: Construct an ordered set  $\mathcal{F}$  of the endpoints from the
   intervals in  $\mathcal{I}$ 
10:  $I = \emptyset$ 
11: for  $z = 1, \dots, |\mathcal{F}|$  do
12:   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  $F_z$ 
13:   if  $\mathcal{W}_z \neq \emptyset$  then
14:      $I \leftarrow I \cup \{(k, \mathcal{W}_z, z)\}$ 
15:   end if
16: end for
17: Transform the intervals in  $I$  to line segments (candidate
   intervals in the edge) and construct the set of potential
   landmarks  $\mathcal{L}_k$ 

```

2) *Algorithm for non-Pointwise landmarks*: Due to assumption (2) only candidate intervals with length greater than or equal to the length of a landmark are considered.

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 terms of the algorithm, this difference is implemented by replacing the function `VisIntersect` to `VisIntersectNP` in the VIA. The pseudo-code of `VisIntersectNP` is presented in algorithm (3).

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 than the normalized landmark length d_l .

VI. RESULTS

This section presents the results of a set of tests performed on the developed method implementation. During the exper-

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  $\partial P$ 
2: Output: List of potential landmarks in the edge  $e_k$ :
    $\mathcal{L}_k = \{l_1, \dots, l_W\}$  and  $S$  a list of pairs of mutually
   exclusive intervals
3: if  $D_l \leq \text{length}(e_k)$  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  $D_l = \text{length}(e_k)$ 
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  $\mathcal{W}_z \neq \emptyset$  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 (can-
   didate intervals in the edge) and construct the set of
   potential landmarks  $\mathcal{L}_k$ 
23: end if

```

iments, it is solved instances of problems 10 and 12. We test the method for different witness sets in two different polygons presented in Figure 3. For the experiments, have been considered two sources of witness sets: random poses generated with uniform distribution; sample poses from predefined trajectories in the polygons. The most relevant results of our experiments are presented in Figures 4 and 5.

The method is trained and then tested in different witness sets. The procedure consists of optimizing the landmarks placement for a given set of poses (training) and evaluating the performance of the obtained solution in different sets of poses (testing), the results of this experiment are presented in Figure 8. In addition, we present a comparative analysis between our method and a set of meta-heuristics algorithms used to solve the LPP, namely: particle swarm (PS), genetic algorithm (GA), and simulated annealing (SA). For this purpose, the VIA 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.

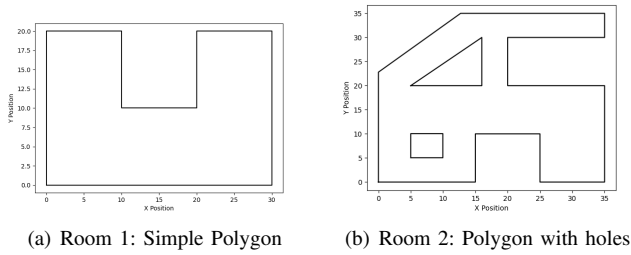


Fig. 3. Polygons were used in the experimental results.

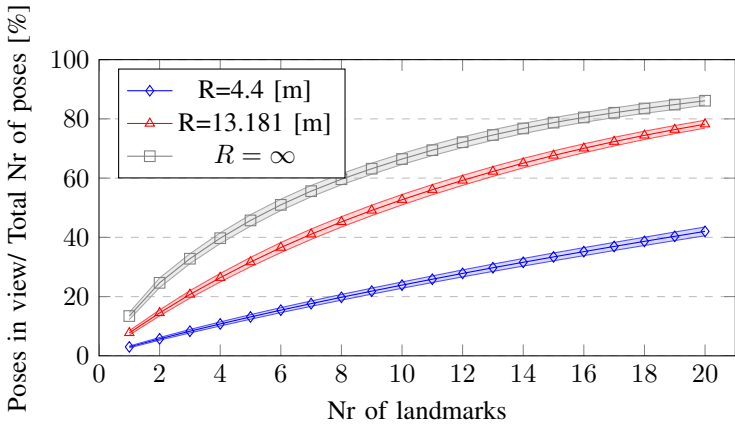


Fig. 4. Room 2: Coverage graph per number of landmarks.

VII. RESULT DISCUSSION

According to the results shown in Figures 6 and 7, our method has better performance than the meta-heuristic algorithms, for the same period of time and a certain number of landmarks, our method cover approximately 20% and 8% more poses than the best meta-heuristic algorithm in Room 1 and Room 2 respectively. As can be seen in Figures 4, 6 and 7, for our method (VIA+ILP) the shape of the curve of coverage as a function of the number of landmarks, is monotonically increasing, since for a greater number of landmarks it is always obtained a greater coverage percentage. For instance, the heuristics algorithms considered in this paper do not show the same monotone behavior, which is a clear indicator of non-optimality.

The training/test results presented in highlight 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 random witness set is only plausible under the assumption that the witness set is representative of the polygon or a certain region of interest. 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

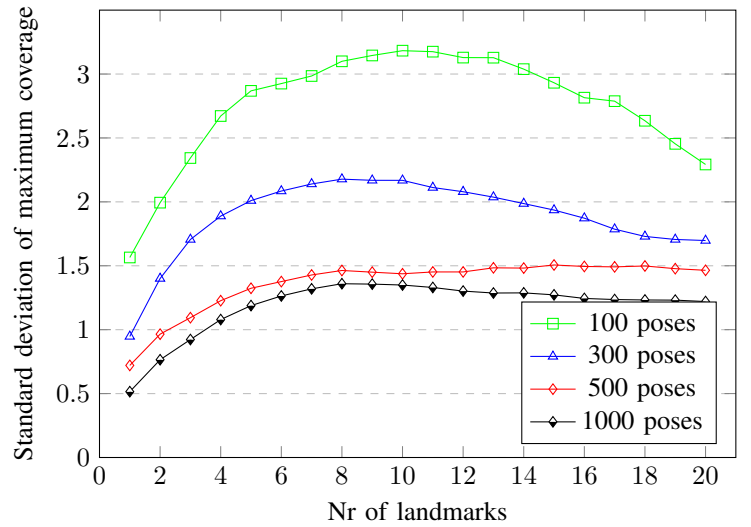


Fig. 5. Standard deviation of the Maximum Coverage per number of landmarks with a sensor detection range of $R = 13.181 m$ in Room 2.

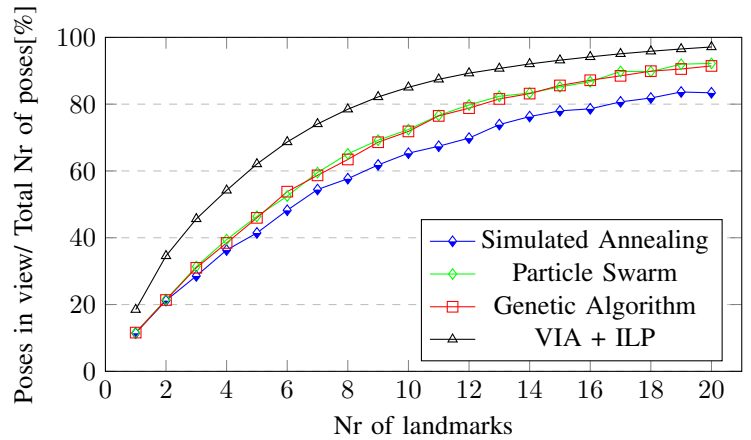


Fig. 6. Mean Coverage comparison with meta-heuristics algorithms for Room 1.

coverage of a random witness set decreases with the number of poses.

VIII. CONCLUSIONS

This paper presents an optimization method for the placement of robot navigation landmarks. It is considered both pointwise and non-pointwise landmarks. The method relies on a simulation of possible robot poses that are believed to be representative of the environment or a family of robot trajectories. The procedure consists of computing a set of candidate intervals where a landmark can be placed through polynomial-time algorithms; formulating and solving the problem as an ILP. An ILP solver guarantees global optimality in contrast with meta-heuristic algorithms. 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*.

The difference in coverage obtained in training and testing

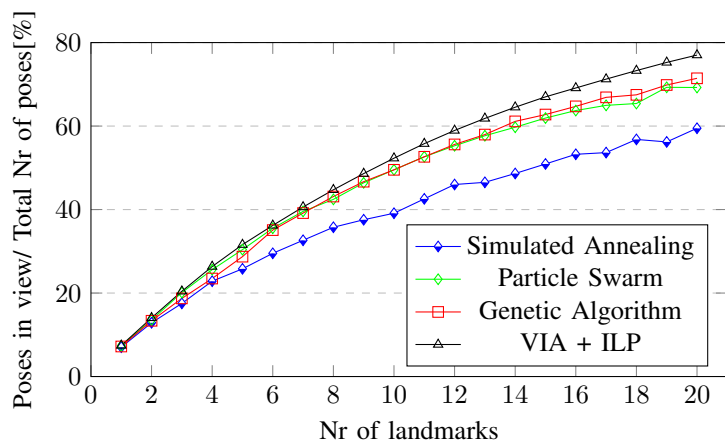


Fig. 7. Mean Coverage comparison with meta-heuristics algorithms for Room 2 (b).

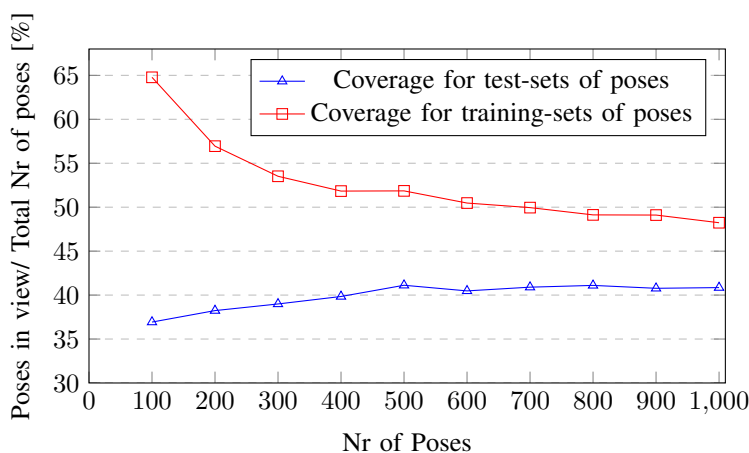


Fig. 8. Comparison between the average coverage obtained in training and test condition for a fixed number of landmarks $L = 10$.

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.

Future works should include aspects like uncertainty on the identification of the landmarks and redundancy. In addition, a study on determining the ideal set of poses according to the environment and features of the sensor system is needed to complement our approach

ACKNOWLEDGMENT

Laboratories Institute for Plasmas and Nuclear Fusion (IPFN) and LARSyS (ISR) received financial support from FCT through projects UIDB/50010/2020 and UIDP/50010/2020 and UIDB/50009/2020, respectively.

REFERENCES

- [1] 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)*, Jan 2019, pp. 7–11.
- [2] 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)*, Sep. 2016, pp. 655–659.

- [3] 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)*, 2017, pp. 1–6.
- [4] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.
- [5] 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)*, Oct 2016, pp. 1–8.
- [6] J. O'Rourke, "Galleries need fewer mobile guards: A variation on chvatal's theorem," *Geometriae Dedicata*, vol. 14, pp. 273–283, 1983.
- [7] O. J., *Art Gallery Theorems and Algorithms*, ser. International series of monographs on computer science. Oxford University Press, 1987.
- [8] D. Lee and A. Lin, "Computational complexity of art gallery problems," *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 276–282, March 1986.
- [9] S. K. Ghosh, "Approximation algorithms for art gallery problems in polygons," *Discrete Applied Mathematics*, vol. 158, no. 6, pp. 718 – 722, 2010.
- [10] A. Kröller, T. Baumgartner, S. Fekete, and C. Schmidt, "Exact solutions and bounds for general art gallery problems," *ACM J. Exp. Algorithmics*, vol. 17, 2012.
- [11] R. Bar-Yehuda and S. Even, "A linear-time approximation algorithm for the weighted vertex cover problem," *Journal of Algorithms*, vol. 2, no. 2, pp. 198 – 203, 1981.
- [12] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.
- [13] J. Salas and J. L. Gordillo, "Placing artificial visual landmarks in a mobile robot workspace," in *Progress in Artificial Intelligence — IBERAMIA 98*, H. Coelho, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 274–282.
- [14] L. Erickson and S. LaValle, "An art gallery approach to ensuring that landmarks are distinguishable," 06 2011.
- [15] M. Ernestus, S. Friedrichs, M. Hemmer, J. Kokemüller, A. Kröller, M. Moeni, and C. Schmidt, "Algorithms for art gallery illumination," *Journal of Global Optimization*, vol. 68, 05 2017.
- [16] U. M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 156 – 169, 2006, special issue on Omnidirectional Vision and Camera Networks.
- [17] M. Beinhofer, J. Müller, and W. Burgard, "Effective landmark placement for accurate and reliable mobile robot navigation," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1060–1069, 2013, selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011).
- [18] S. K. Ghosh, *Visibility Algorithms in the Plane*. Cambridge University Press, 2007.
- [19] R. Paschotta, *Encyclopedia of Laser Physics and Technology*. Wiley, 2008.
- [20] J. Greivenkamp, *Field Guide to Geometrical Optics*, ser. Field Guide Series. Society of Photo Optical, 2004.
- [21] L. Wolsey and G. Nemhauser, *Integer and Combinatorial Optimization*, ser. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2014.
- [22] C. Chekuri and A. Kumar, "Maximum coverage problem with group budget constraints and applications," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, K. Jansen, S. Khanna, J. D. P. Rolim, and D. Ron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 72–83.
- [23] J. Schmidt, "Interval stabbing problems in small integer ranges," vol. 5878, 12 2009, pp. 163–172.