A PROBABILISTIC APPROACH TO AUTONOMOUS PATH PLANNING FOR DIRECTIONAL MOBILE SENSORS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Min Zheng

August 2018

A PROBABILISTIC APPROACH TO AUTONOMOUS PATH PLANNING FOR DIRECTIONAL MOBILE SENSORS

Min Zheng

Cornell University 2018

Directional sensors, such as vision, infrared, ultrasound, and active acoustic sensors, are characterized by a preferred sensing direction, such that measurements are obtained only for a bounded subset of all possible aspect angles. By such approach, directional sensors can obtain information about the target's relative orientation, in addition to its distance. Common applications include cameras mounted on autonomous vehicles that may be used for urban surveillance or target recognition by means of on-board computer vision algorithms. One of the major challenges in planning the motion of directional mobile sensors is that an important target of interest may be occluded by the presence of obstacles in the sensor's line-of-sight. This thesis addresses this path-planning problem for an Unmanned Ground Vehicle (UGV) equipped with a vision sensor for the purpose of classifying multiple static targets in an obstacle-populated environment. An approach is developed for determining a UGV path that enables observations from all targets with known locations in minimum time. The approach guarantees that the UGV is able to classify every target previously localized, while avoiding collisions with obstacles and occlusions that prevent line-of-sight visibility. The approach consists of mapping targets into the UGV configuration space, thus obtaining C-targets, using complexity reduction techniques that take into account shadow regions caused by the presence of obstacles. An information roadmap method (IRM)

algorithm is used to build a connectivity graph from the C-target regions, and a solution with the least translation distance is obtained. Comprehensive simulations performed in MATLAB and Webots - a professional robot simulator that provides modules for sensors, robots, and their interactions with a 3-D virtual environment - demonstrate the effectiveness and performance improvement of the proposed approach when compared to existing methods, based on ”nearest neighbor” and classical traveling salesman problem (TSP) algorithms.

BIOGRAPHICAL SKETCH

Min Zheng is an M.S. student in the Laboratory for Intelligent Systems and Controls (LISC) at Cornell University. She received B.S. Degree in Mechanical Engineering at the University of Southern California in 2016. Her research interests are in the field of robot sensing and information-driven path planning.

ACKNOWLEDGEMENTS

I would first like to express my gratitude to my advisor, Dr. Silvia Ferrari, for her enormous help, consistent motivation and kind support during my graduate study at Cornell University. It was a great honor to finish this work under her advisement. I would like to thank Dr. Knepper for being my defense committee member and offering valuable suggestions on my research. I am also genuinely appreciative of Dr. Zhu and Dr. Fu in LISC, who have also provided tremendous help in my research and suggestions in my preparation for defense. Last but not the least, I would like to thank all the LISC members and my friends at Cornell University, who have given me constructive feedback and great support during the work. This work was funded by Office of Naval Research Grant ONR N000141310561.

CONTENTS

Biographical Sketch . . . iii
Acknowledgements . . . iv
Contents . . . v
List of Tables . . . vi
List of Figures . . . vii

1 Introduction and background **1**

2 Problem Formulation and Assumptions **6**

3 Methodology **14**
3.1 Minimizing Expected Entropy Reduction . . . 15
3.2 Directional C-targets . . . 18
3.2.1 C-target with obstacle avoidance . . . 20
3.3 Pruning of C-target regions . . . 24
3.4 Traversal of C-target with minimum translation . . . 32
3.4.1 C-target connectivity graph . . . 33
3.4.2 Edge weight for a complete connectivity graph . . . 35
3.5 Waypoint optimization of translational and rotational cost . . . 39

4 Simulations and results **41**
4.1 Simulation workspaces and sensor parameters . . . 41
4.2 Influence of important parameters in the planning approach . . . 46
4.3 Simulations in MATLAB and Webots® environment . . . 50
4.4 Performance comparisons with benchmark methods . . . 56

5 Conclusion and Future Work **61**

Bibliography **63**

LIST OF TABLES

4.1 Performance comparison of the proposed method and three benchmark methods using map 1 58
4.2 Performance comparison of the proposed method and three benchmark methods using map 2 58
4.3 Performance comparison of the proposed method and three benchmark methods using map 3 58

LIST OF FIGURES

2.1 Onboard sensor with FOV $\mathcal{S}$ (green region) is mounted on the robotic platform $\mathcal{A}$. A target with coordinate $\mathbf{x}_T$ is not in the sensor's line-of-sight, since there is a point $\xi \in \mathcal{B}$ (marked by the red cross) that is in the same direction as $\mathbf{x}_T$ with respect to to the sensor's configuration, and is closer to the sensor than $\mathbf{x}_T$. 8

2.2 Example of the BN model used in this thesis. The feature discrete random variables are X_1 (shape), X_2 (color), and X_3 (texture), and the categorical random variable is Y, representing Treasure, Non-treasure. 10

3.1 The FOV of a sensor at configuration $\mathbf{q}$ is $\mathcal{S}(\mathbf{q})$, a subset of the configuration space. The FOV has a minimum and maximum range, r_{min} and r_{max}, and a maximum open angle α. In this example, although the target (red dot) with coordinate $\mathbf{x}_T$ is in the FOV, it is not in the line-of-sight of the sensor. Therefore, the robot current configuration does not belong to the C-target region of this target. 19

3.2 The robotic platform $\mathcal{A}$ is a rectangle with width w and length l, with geometry center at $\mathcal{O}_{\mathcal{A}}$. The disc (orange circle), $\mathcal{A}'$, is obtained by sweeping $\mathcal{A}$ around $\mathcal{O}_{\mathcal{A}}$ with all possible orientation. If at this coordinate, $\mathcal{A}'$ is collision-free with all the obstacles, such coordinate belongs to the 2-dimensional C-free $\mathcal{C}'_{free}$. 22

3.3 Demonstration of $\mathcal{C}_{target}$ in 3-D configuration space. The test workspace is the Webots® immersive environment with 30 targets. 23

3.4 A demonstration of the case where the number of waypoints needed is less than the number of C-target regions. To make measurements on four of the targets (red dot), a possible path is to visit waypoints sequence $\mathcal{P}_1 = \langle \mathbf{x}_a^1, \mathbf{x}_a^2, \mathbf{x}_a^3, \mathbf{x}_a^4 \rangle$ (blue triangles), another is to visit $\mathcal{P}_2 = \langle \mathbf{x}_b^1, \mathbf{x}_b^2 \rangle$ (orange triangles). The second path is more likely to have a lower cost. 25

3.5 The first step of C-target pruning approach. C-target regions $\mathcal{G}_a$, $\mathcal{G}_c$, $\mathcal{G}_e$ are marked as redundant and should be pruned. The remained regions are $\mathcal{G}_b$ and $\mathcal{G}_d$, by visiting which the three targets, $\mathcal{T}_1$,$\mathcal{T}_2$ and $\mathcal{T}_3$ can all be observed. 27

3.6 Two pairs of independent C-target regions, each contains two regions. The two pairs can observe the same set of targets. The pruning approach selects the pair that contains a larger number of coordinates. By visiting the selected regions, the four targets, $\mathcal{T}_1$,$\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$ can all be observed. 29

3.7 For the chained C-target regions, the pruning approach first selects the intersecting regions as described in case 1. Then, at the third step, the approach further checks the redundancy and discards unnecessary regions if possible. By visiting the selected regions, the four targets, $\mathcal{T}_1$,$\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$ can all be observed. . . 30
3.8 The effect of C-target pruning operation on the test workspace for satisficing experiments performed in Webots®. The operation greatly reduces the number of C-target regions needed to be traversed, allowing an improvement of planning efficiency. . . . 31
3.9 An example of the connectivity graph construction on the test workspace. A blue dot represent an assigned coordinate $\mathbf{r}_i$ for each C-target region $\mathcal{G}_i$. Point **a** is the start position. To estimate the distance cost between point **b** and **d**, the approach using existing nodes, such as using point **c**, may result in an inappropriate estimation. Instead, a path generated by PRM is used. Point **e** denotes the coordinate for a region whose geometry is not connected. Instead of its centroid, which does not belong to C'_{free}, an alternative point is sampled from the region. 38

4.1 Map 1 applied in the simulations. It is relatively small scale, with 9 obstacles and 7 targets. 42
4.2 Map 2 has 25 obstacles and 30 targets. It is associated with two sets of targets of different prior feature knowledge. In the first set, 30 targets have non-zero EER. 43
4.3 Map 2 has 25 obstacles and 30 targets. It is associated with two sets of targets of different prior feature knowledge. In the second set, 15 out of 30 targets have non-zero EER, marked as red dots, while the targets with zero EER are marked by green dots. . . . 44
4.4 Map 3 is a large-scale maze. 27 targets are placed throughout the map. Some targets, such as 22, 23, 24 and 25 are placed near each other to form a "cluster". 45
4.5 The top view of the Webot® immersive environment along with the view of sensor camera. The workspace setup corresponds to the second map introduced above. 30 targets are placed in the environment, and they may subject to different initial observable feature set $\mathcal{F}_i$. 45
4.6 A demonstration of the possible effect of n_{PRM} on map 2. The plot on the left has a number of nodes far less than needed given the map complexity. Thus, the order of traversing the C-targets with the minimum cost may not be obtained. 47

4.7 A summary of the effect of n_{PRM} on map 2. The result is evaluated in terms of the translational distance, plotted in blue with a scale on the left axis. Their corresponding time consumptions of edge assignments via PRM is also plotted in red. The standard deviation obtained from the experiment is shown by the error bars. 48
4.8 Effect of w_t in terms of rotational angle and translational distance for map 1 and map 2. The standard deviation obtained from the experiment is shown by the error bars. 49
4.9 Effect of w_t demonstrated on map 2. The circles in each plot highlights the interesting effect of the weight on translation. To observe the two targets, the path on the right travels a significantly longer distance but with less rotation, since it has less weight on translation. 49
4.10 An example result for map 1 performed using MATLAB, with w_t = 1, $n_{PRM} = 200$. The robot saves translational distance significantly by rotating at certain waypoints to observe multiple targets. The total translation is 8.1 m and total rotation is 19.8 rad. 51
4.11 The optimal path for map 2, with 30 targets of non-zero EER, determined by C-target pruning with w_t = 1, $n_{PRM} = 1000$. The robot rotates at a waypoint in pruned C-target region to observe multiple targets if allowed. In this manner, the translation will be greatly reduced. The total translation is 30.0 m and total rotation is 45.1 rad. 52
4.12 Demonstrated in Webot® simulations, the mobile sensor navigates itself through a complex environment that is populated with multiple obstacles. The figure shows the process of the sensor obtaining measurements from a target under the table. 53
4.13 An example solution to map 2 that has 30 targets with non-zero EER, simulated in Webot®. With the help of C-target pruning operation, the mobile sensor reaches waypoints that may allow measurements on multiple targets. By adopting this approach, the translational cost is significantly reduced. 54
4.14 The optimal path for map 2, with 15 targets of non-zero EER, determined by C-target pruning with w_t = 1, $n_{PRM} = 1000$. The total translation is 21.6m. By determine a set of target with additional information gain, the translation cost is reduced by 28%. 55
4.15 The optimal path for map 3, determined by C-target pruning with w_t = 0.5, $n_{PRM} = 1200$. The robot rotates at a waypoint in pruned C-target region to observe multiple targets if allowed. Benefit from this behavior, the solution's translational cost is greatly reduced. The total translation is 76.6 m. 55

4.16 The optimal path determined by cell decomposition and integer programming (IP) for map 2, with 30 targets of non-zero EER, and the optimization assumption that translation dominates. The total translation is 39.2 m and total rotation is 58.5 rad, while with similar assumptions, the translational distance of the proposed method (CTP) is 30.2 m, and rotational angle is 52.5 rad. 59
4.17 The optimal path for map 3, with $n_{PRM} = 1200$, determined by Heuristic TSP via 2-opt. Without benefiting from the sensor's FOV, the robot needs to visit at the exact location of every target. The total translation is 87.8 m. While with similar assumptions, the translational distance of the proposed method (CTP) is 77.6 m. 60

CHAPTER 1

INTRODUCTION AND BACKGROUND

Sensor planning is to develop a strategy to gather measurements for a sensing objective, such as target classification [1]. When solving this problem with mobile sensors mounted on robotic platforms, planning the sensor path is one of the essential tasks [1, 2]. Existing works, such as [3–6], have provided abundant solutions to robot navigation in dynamic or unknown environments with obstacle avoidance. In those studies, the onboard sensors are mostly used for assisting the navigation, while this thesis focuses on utilizing the sensors to support a sensing objective.

This thesis addresses the problem of path planning of an Unmanned Ground Vehicle (UGV) with an onboard directional sensor to classify multiple fixed targets in an obstacle-populated environment in minimum time. In this problem, the sensor serves as an information gathering agent along the path traveled by the robotic platform. This problem is also known as a variant of the treasure hunt problem, a coupled problem of robotic sensor motion and measurement planning [7]. It is highly relevant to many modern sensors and advanced surveillance systems, with potential applications including real-world landmine detection [8], wildlife tracking for biology research [9] and environmental monitoring [10]. In addition, it provides a benchmark optimization solution to active satisficing experiments, where human or animal subjects seek for targets and complete classification tasks in immersive environments, sometimes under environmental pressure such as time pressure [11, 12]. The term satisficing refers to the approach of stopping the search as soon as a solution that meets the expectation is achieved [13].

The satisficing studies are performed to understand and approximate the high-paced real-world decision making process with uncertain information and limited resources [11]. The solution to the proposed problem in this thesis is able to provide a comparison with human navigation and decision making strategies, and thus a better understanding in such field.

Directional sensors, such as vision, infrared, ultrasound, and active acoustic sensors, only allow measurements for a bounded subset of all possible aspect angles. For such sensors, information about a target's relative orientation can be obtained, in addition to its distance. A major challenge in planning the motion of directional mobile sensors is that an important target of interest may be occluded by the presence of obstacles in the sensor's line-of-sight. The planning problem thus requires knowledge of the targets' geometry and position and sensor's field-of-view (FOV) [14]. Therefore, the UGV path solution to the proposed problem must enable measurements of every target previously localized in the minimum time, while avoiding collisions with obstacles and occlusions that impair line-of-sight visibility.

The following discussion will examine the methods presented in previous literature and their applicability to the proposed problem. Solutions to coverage path planning allow observations from every necessary target. For instance, approaches of complete coverage and random coverage introduced in [2, 15] allow the robotic sensor to pass over all possible points of a field. Although they demonstrate capability in tasks like floor cleaning, demining and painting [2,15, 16], these methods do not utilize priorly known workspace information, so that the result paths are often cost inefficient. Alternatively, several approaches to the information-driven sensing problem are developed, where the information

of the workspace and sensor's FOV geometry are well represented. Information roadmap method (IRM) introduced in [1] combines probabilistic roadmap method (PRM) and information-driven sensor planning, aided by information theoretic functions that allow estimation on the measurements' value before deploying the sensor. With a hybrid sampling strategy, the approach is able to generate an efficient path with obstacle avoidance in a workspace that has both wide-open regions and narrow passages. Information potential method (IPM) in [17] generates a potential navigation function from conditional mutual information to maximize the information value gain along the sensor path while escaping local minima using a local roadmap. However, neither of the above approaches guarantees the coverage of all targets with a path of the minimum time cost.

On the other hand, target coverage with path optimization is well discussed in the problem of wireless sensor networks data collection [18–20]. In this problem, a vehicle needs to efficiently collect data from a set of sensors, and each has a limited communication range. It is very relevant to the proposed sensor planning problem, where the mobile sensor collects measurements along an optimal path with its FOV represented by a closed and bounded subset of a Euclidean space. Many methods in existing studies abstract the data collection problem into a traveling salesman problem with neighborhoods (TSPN) or group-TSP, a discrete variant of TSPN [18]. TSPN is a variant of traveling salesman problem (TSP). The two problems seek for the shortest path under different requirements for coverage. While a TSP solution needs to visit a set of points, each point exactly once, TSPN requires that at least point in each of the given connected region be visited [21], rendering the problem even more complex. Previous research on TSPN such as [22] provides approximation

algorithms of the optimal solution for the case that the regions are fat objects with possibly varying size. Other works also present approaches using genetic algorithm (GA), a meta-heuristic method developed based on the process of natural selection, for its capability of generating a high-quality solution from a large set of possible permutations [18, 23, 24]. For example, [18] has proposed a combination of Euclidean-distance based clustering algorithm and GA for data-gathering route design. However, the above methods are formulated on the Euclidean plane assuming that the environment is free of obstacles. The regions geometry is also highly limited to circles or convex fat objects. Therefore, these methods can not yet provide a general approach for sensor path planning in an obstacles-populated workspace.

To overcome the limitations of the methods discussed above, this thesis presents a novel and practical approach seeking for an optimal path with obstacle avoidance, while allowing measurements on every necessary target. It achieves the objective by representing the planning problem for sensing objective with a connectivity graph, which can be solved as a TSP with the aid of PRM and pruning operations to reduce optimization search space. One advantage over other techniques is that the method can take into account the obstacles in the workspace and in the sensor's line-of-sight. In addition, it is able to efficiently generate paths with low time cost, while fulfilling the sensing objective.

As Chapter 3 would elaborate, the approach applies an information theoretic function, the expected conditional entropy reduction (EER), to determine the targets from which valuable measurements can still be obtained. Then it constructs the corresponding directional target configurations (C-target) with

careful attention to the presence of obstacles. A series of C-target pruning techniques are then applied for complexity reduction. A connectivity graph from the C-target regions is constructed by incorporating special problem properties, and a solution with the least translation distance is obtained. As demonstrated in Chapter 4, the proposed approach meets the sensing objective with less time cost than the traditional TSP algorithms and an alternative method that utilizes cell-decomposition and integer programming [25]. Through a series of comprehensive simulations, the proposed method demonstrates its capability in generating a high-quality solution as well as the adaptability to different workspace complexities.

CHAPTER 2

PROBLEM FORMULATION AND ASSUMPTIONS

This thesis considers the problem of planning the path of a robotic sensor comprised of an Unmanned Ground Vehicle (UGV) for the purpose of classifying multiple targets in an obstacle-populated workspace.

A directional sensor is mounted on the geometric center $\mathcal{O}_{\mathcal{A}}$ of a robotic platform, whose geometry is denoted by $\mathcal{A}$. The directional sensor only obtains measurements in a bounded subset of all possible aspect angles, so that it can provide information about the target's orientation relative to the sensor, in addition to its distance. Let $\mathcal{F}_{\mathcal{W}}$ denote a fixed Cartesian frame in workspace $\mathcal{W}$, whose origin is denoted as $\mathcal{O}_{\mathcal{W}}$. Due to its properties, the robotic sensor has a configuration vector $\mathbf{q}$ that includes both x and y coordinates of $\mathcal{O}_{\mathcal{A}}$ and the sensor's orientation in $\mathcal{F}_{\mathcal{W}}$. Thus there is $\mathbf{q} = [x, y, \theta]^T \in \mathcal{C}$, where $\mathcal{C}$ denotes all possible robot configurations in the workspace $\mathcal{W} \subset \mathbb{R}^2$, i.e., $\mathcal{C} \subset \mathcal{W} \times (-\pi, \pi)$.

The sensor's field-of-view (FOV) is a closed and bounded subset of the workspace, $\mathcal{S}(\mathbf{q}) \subset \mathcal{W}$, in which target measurements can be obtained. Let $\mathcal{F}_A$ be a moving Cartesian frame embedded in $\mathcal{A}$. Assume that $\mathcal{A}$ and $\mathcal{S}$ are both rigid, and $\mathcal{S}$ is fixed with respect to $\mathcal{A}$. Then the robot configuration $\mathbf{q}$ can fully specify the position and orientation of every point in $\mathcal{A}(\mathbf{q})$, and every point in $\mathcal{S}(\mathbf{q})$. The UGV is subject to the unicycle robot kinematics with

$$\dot{\mathbf{q}}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} cos\,\theta(t) & 0 \\ sin\,\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \mathbf{g}[\mathbf{q}(t)]\mathbf{u}(t) \tag{2.1}$$

where the control input is $\mathbf{u} = [v, \omega]^T \in \mathcal{U} = \{(v, \omega\,) |\, 0 \leq v \leq v_m, 0 \leq \omega \leq \omega_m\}$

with v_m, ω_m as the maximum permissible velocity and angular velocity. $\mathbf{g}[\cdot]$ denotes a nonlinear vector function of the configuration.

The workspace is assumed to be a compact subset of a Euclidean space. It is populated with n fixed and rigid obstacles $B = \{\mathcal{B}_1, ..., \mathcal{B}_n\}$, whose position and dimensions are known. The workspace also contains r targets, denoted as $T = \{\mathcal{T}_1, ..., \mathcal{T}_r\}$. The target geometry is simplified as a point, and its coordinate can be described by vector $\mathbf{x} = [x\ y]^T$. The obstacles and targets are placed following the assumption that $\mathcal{B}_i \cap \mathcal{T}_j = \emptyset$ for $\forall i \in I_B$ and $\forall j \in I_T$, where I_B an I_T are the index sets of B and T.

In order to be observed by the sensor, a target $\mathcal{T}_i$ must be in the sensor's FOV at the given robot configuration, i.e., $\mathcal{S}(\mathbf{q}) \cap \mathcal{T}_i \neq \emptyset$. In addition, possible occlusions caused by obstacles in the directional sensor's line-of-sight (LoS) must be taken into account. Let $\mathbf{x}_T$ denote the coordinate of a target of interest $\mathcal{T}_i$. The relative position of $\mathbf{x}_T$ with respect to the sensor is $\mathbf{r}_T = (\mathbf{x}_T - \mathbf{q})$. Then at configuration $\mathbf{q}$, $\mathbf{x}_T$ is in the sensor's LoS if there is no point in the obstacle region $\mathcal{B}$ that are co-directional with $\mathbf{r}_T$ and closer to $\mathbf{q}$ than $\mathbf{x}_T$, or

$$\nexists\, \xi \in \mathcal{B} \text{ s.t. } \xi \cdot \mathbf{r}_T = \|\xi\|\,\|\mathbf{r}_T\| \text{ and } \|\xi\| < \|\mathbf{r}_T\| \tag{2.2}$$

where ξ is defined with respect to $\mathcal{F}_A$, the frame embedded on the robotic platform $\mathcal{A}$.

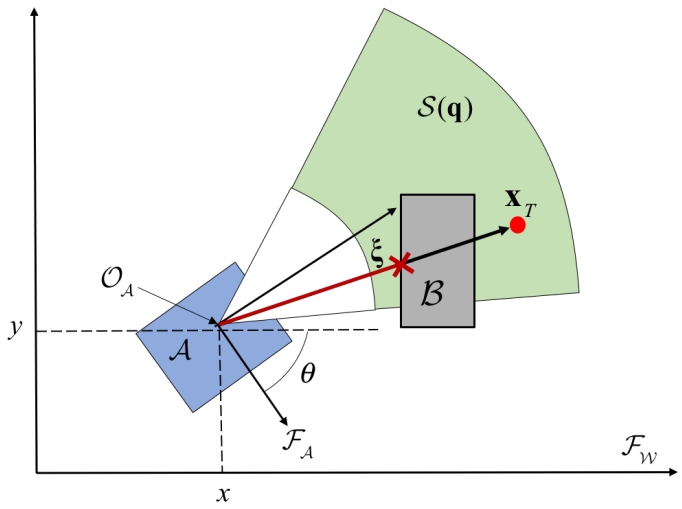


Figure 2.1: Onboard sensor with FOV $\mathcal{S}$ (green region) is mounted on the robotic platform $\mathcal{A}$. A target with coordinate $\mathbf{x}_T$ is not in the sensor's line-of-sight, since there is a point $\xi \in \mathcal{B}$ (marked by the red cross) that is in the same direction as $\mathbf{x}_T$ with respect to to the sensor's configuration, and is closer to the sensor than $\mathbf{x}_T$.

In order to perform path planning with obstacle avoidance, the free configuration is determined as the following. A C-obstacle is a subset of C that has collisions with at least one obstacle, denoted as $\mathcal{CB}_i \equiv \{\mathbf{q} \in \mathcal{C} | \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset\}$, where $\mathcal{A}(\mathbf{q})$ denotes the subset of $\mathcal{W}$ occupied by $\mathcal{A}$ with UGV configuration $\mathbf{q}$. Then, the union of all C-obstacles obtained from B is defined as the C-obstacle region, i.e., $\mathcal{CB} = \bigcup_i \mathcal{CB}_i$. The free configuration is the complement of the C-obstacle region, i.e., $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{CB}$ [26]. The robotic sensor is free to rotate and translate in this free configuration space.

The robotic sensor needs to obtain additional measurements to properly classify a subset of targets in T. Observations on a target is obtained sequentially at a constant cost c_r using the onboard active sensor camera. The sensor measurement process on each target $\mathcal{T}_i \in T$ is modeled by a joint probability mass function (PMF) of the relevant variables. In this thesis, the PMF is

represented by a Bayesian Network (BN) model. Many existing works adopted such model, since it captures the conditional dependencies between multiple variables, and provides a convenient factorization of the joint PMF [1,7,17].

A BN model utilizes a directed acyclic graph (DAG), $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, and a set of conditional probability tables (CPTs) to represent the multivariable joint PMF of a set of discrete and random variables. For each target $\mathcal{T}_i \in T$, the node set $\mathcal{N}$ consists of a categorical random variable Y_i and L discrete feature random variables, denoted by $\{X_{i,1}, ...X_{i,L}\}$. The categorical random variable has a finite range of classification, $\mathcal{Y}_i = \{y_i^1, y_i^2\}$. Y_i is hidden and must be inferred from the target feature measurements. Each feature measurement variable $X_{i,k}$ has a finite range, i.e., $\mathcal{X}_{i,k} = \left\{x_{i,k}^1, ...x_{i,k}^{N_k}\right\}$, with $x_{i,k}^q$ denotes the qth value in $\mathcal{X}_{i,k}$, the range of the variable representing the kth level measurement for target $\mathcal{T}_i$. Note that for each target, L is a constant, and so is every N_k for $k = 1, ..., L$. Also, before the robotic sensor is deployed, each target is associated with l_i levels of features that has been given, denoted as $\mathcal{F}_i = \left\{x_{i,1}^{v1}, ...x_{i,l_i}^{vf}\right\}$, where $x_{i,k}^{vj}$ denotes the jth value in the range of the kth level of feature variable.

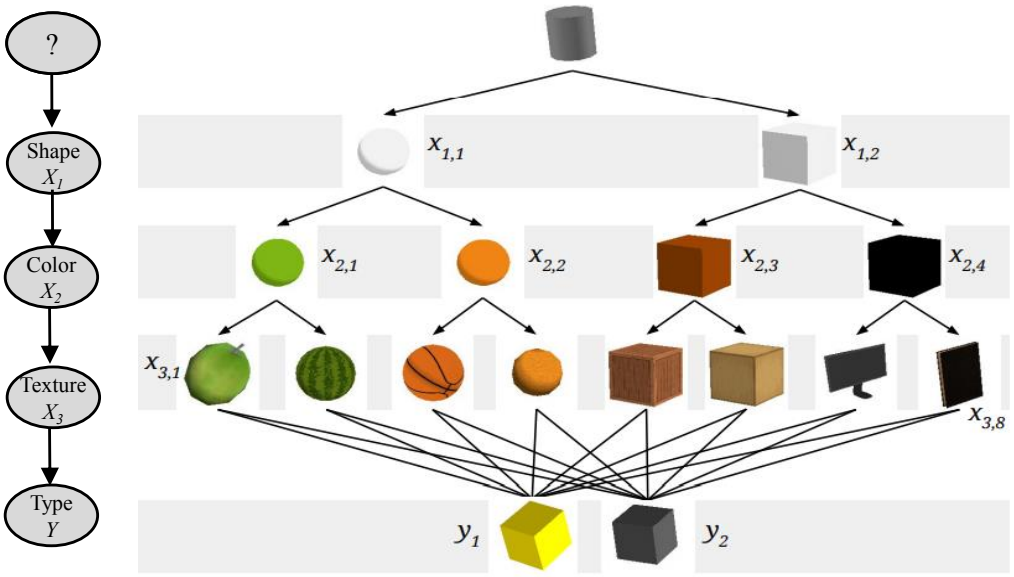


Figure 2.2: Example of the BN model used in this thesis. The feature discrete random variables are X_1 (shape), X_2 (color), and X_3 (texture), and the categorical random variable is Y, representing Treasure, Non-treasure.

The arc set $\mathcal{E}$ represents the conditional probability relationships between the variables. For target $\mathcal{T}_i$, the BN model specifies the joint PMF underlying the sensor measurements as

$$P(\mathcal{N}) = P(Y_i, X_{i,1}, ..., X_{i,L}) = P(Y_i|pa(Y_i)) \prod_{j=1}^{L} P(X_{i,j}|pa(X_{i,j})) \tag{2.3}$$

where $pa(X_{i,j})$ is the parent set of a node $X_{i,j}$, and all $P(Y_i|pa(Y_i))$ and $P(X_{i,j}|pa(X_{i,j}))$ can be obtained from the BN CPTs.

Given a set of values known from available observations $\mathcal{F}_i'$, which either is obtained directly from $\mathcal{F}_i$ or with additional sensor measurements, the target classification can be decided using the posterior PMF. One of the most commonly used rules for classification decisions, maximum a posteriori

probability rule (MAP) decides the most probable value of Y from $\mathcal{Y}_i$ based on $\mathcal{F}_i'$ as

$$\hat{Y}_i = y_i^*, \text{iff } P(\hat{Y}_i = y_i^* | \mathcal{F}_i') \geqslant P(\hat{Y}_i = y_i^{l_i} | \mathcal{F}_i'), \forall y_i^{l_i} \in \mathcal{Y}_i, y_i^{l_i} \neq y_i^* \tag{2.4}$$

The path planning objective is to allow the robotic sensor to gather all necessary information value through target measurements, with a robot trajectory that is collision-free with obstacles and is of a minimum time cost. The path of the UGV, defined as a continuous map $\tau : [0, 1]$, with the initial and goal configurations defined as $\mathbf{q}_0$ and $\mathbf{q}_f$, is determined by the planning methodology to be introduced. The initial configuration $\mathbf{q}_0$ is known, but the goal configuration $\mathbf{q}_f$ is not specified. Denote T_I as a set of target that the UGV must collect measurements from to achieve the desired information gain, the representation of which will be introduced in the next Chapter.

The robot trajectory τ must meet the following objectives:

1. Avoid collisions with all obstacles in $\mathcal{W}$.

2. The obtained trajectory must allow measurements on every target $\mathcal{T}_i \in T_I$, or

$$\forall \mathcal{T}_i \in T_I, \ \exists \, \tau(s) = \mathbf{q}_i \ \text{ s.t. } \mathbf{q}_i \in \mathcal{CT}_i \tag{2.5}$$

where $s \in [0, 1]$, and $\mathcal{CT}_i$ is the set of target configurations of a target $\mathcal{T}_i$.

3. Minimize the total time consumption throughout the path.

Objective 1 is met by planning a path in the free configuration space. To achieve objective 2, a representation of information value using expected entropy reduction (EER) method will be discussed in Chapter 3. To achieve objective 3 with a simplification on the planning problem, several assumptions are made. The first assumption is that the target locations, available target

features, obstacle geometry and location are known, so that the planner can perform the planning off-line. Secondly, the feature measurement time is negligible compared to the motion time of the UGV, and revealing features has no cost, i.e., $c_r = 0$. Thus, the sensor system is assumed to reveal all the features on a target once they are available at the given configuration. The third assumption is that the effects of acceleration and deceleration throughout the trajectory is assumed to be negligible. Therefore, the UGV is assumed to translate with the maximum translational velocity v_m and rotates with the maximum angular velocity ω_m, which are both constants. Furthermore, translational time cost is assumed to dominate in the UGV motion time, i.e., $\frac{d_t}{v_m} \gg \frac{d_\theta}{\omega_m}$.

Let t_{tot} be the total time consumption of the path with start configuration $\mathbf{q}_0$ and end configuration $\mathbf{q}_f$. The total time consists of robot platform motion time t_m and target features measurement time t_o, i.e. $t_{tot} = t_m + t_o$, while t_o is negligible comparing to t_m per the second assumption made above. Then to achieve a minimum t_{tot}, the obtained path should have a minimum UGV motion time t_m. t_m consists of the translational and rotational time cost, and per the third assumption made above, there is

$$t_m = \frac{d_t}{v_m} + \frac{d_\theta}{\omega_m} \tag{2.6}$$

By these assumptions and transformation made above, the minimum-time path planning problem is therefore to minimize d_t and d_θ. The methodology to be presented will minimize d_t first, and then d_θ if possible, based on the assumption that translation cost is dominant. The method first identifies a set of targets T_I to obtain measurements from based on expected entropy reduction, a selected information theoretic function. Then, the method will determine a path for the robotic sensor in the free configuration space, while the sensing objective

defined in Equation 2.5 is met.

CHAPTER 3

METHODOLOGY

This chapter presents a set of systematic path planning approaches to determine a minimum time path that allows measurements from all targets with information values. The goal is achieved by constructing and operating on directional target configuration (C-target) based on targets' information value. As one of the major challenges in this problem, a directional sensor only allows measurements at preferred sensing direction, so that the sensor's orientation must be considered in the planning. On the other hand, as many existing works of literature and practices have discussed [26–28], the time complexity to solve a path planning problem increases quickly with the dimension of the configuration space. To obtain a practical solution within a moderate runtime, the reduction of dimension when possible is necessary and crucial. This thesis hence proposes a technique using a switched focus of optimization in different stages of path planning, based on the assumption that the cost of translation is not less than that of rotation in the total time cost.

Firstly, the information value of each target's available measurements is described using expected entropy reduction, a practical information-theoretic function. Then, C-target is constructed with careful consideration of line-of-sight visibility constraint to ensure that a target of interest is not subject to occlusion by obstacles. After that, the path planner will minimize only the cost of translation based on the position vector $\mathbf{x} = [x \ y]^T$ that describes the robot's coordinate in the workspace, while obtaining all necessary measurements. A set of C-target regions is acquired after a pruning operation that improves planning efficiency, and a traversal order is obtained

with a traveling salesman problem (TSP) representation. Finally, waypoint optimization is performed to minimize the cost of both translation and rotation. Along the waypoints in the obtained path, the robotic sensor with configuration $\mathbf{q} = [\mathbf{x}^T \ \theta]^T$ adjusts its orientation to follow the trajectory and allow target measurements. The specially designed techniques make a high-quality solution possible even with a complex workspace map, as demonstrated in Chapter 4.

3.1 Minimizing Expected Entropy Reduction

In information-driven sensor planning, the planner navigates the robotic sensor to collect valuable sensor measurements for target classification. Although the process of performing a measurement itself has no cost, the robot will still approach a target and adjust its orientation for a measurement, consuming time for both translation and rotation. Thus, it is crucial to estimate the quality and quantity of measurements before obtaining them, so that the planning can achieve better efficiency.

As introduced in Chapter 2, the sensor measurement process is modeled by a joint probability mass function (PMF) of a set of discrete random variables that describe target classification and features. This section introduces a representation of information value by expected entropy reduction (EER) in terms of the joint PMF. EER evaluates a measurement's reward by the ability to reduce the uncertainty in the classification variable. It is preferable as it demonstrates the effectiveness and reliability in sensor planning for target classification, even when the sensor models are not Gaussian [29]. Similarly, other information-theoretic function such as discrimination gain [30], and

the Entropy and the Mahalanobis distance measure [31], have shown their applicability in other problems, such as multisensor-multitarget assignment problem [30], or the problem of sensor collaboration in ad-hoc sensor networks [31].

Entropy reduction is formulated using conditional entropy. The conditional entropy of a discrete and random variable Y given another variable Z is described by the expected value of the entropies of the conditional distributions over the range of the conditioning random variable [32]

$$H(Y|Z) = -\sum_{z}\sum_{y} P(y,z)\log_2 P(y|z) \tag{3.1}$$

where $H(\cdot)$ denotes the Shannon entropy, and $\sum_y$ denotes the marginalization over the range of Y.

Then entropy reduction, which is shown to be additive in [29], describes the reduction in uncertainty brought by a set of new measurements Z_j with prior information about Z_i as

$$\Delta\hat{H}(Y; Z_j|Z_i) \equiv H(Y|Z_i) - H(Y|Z_i, Z_j) \tag{3.2}$$

For the problem discussed in this thesis, the EER is further specified for each target $\mathcal{T}_i$ with L levels of features and a classification variable Y_i. Recall that for $\mathcal{T}_i$, an a-priori evidence set containing the result of first l_i feature measurements $\mathcal{F}_i = \left\{x_{i,1}^{v1}, ...x_{i,l_i}^{vf}\right\}$ is given before the sensor deployment. The measurement on each target feature can only be made once, that is, measurement is only allowed when $l_i < L$. Also, per the assumption previously made, the robotic sensor will reveal all features on a target once they become available at the sensor's configuration. The process of the measuring the rest features from level $l_i + 1$

to L can be described by $M_i = \left\{X_{i,(l_i+1)}, ...X_{i,L}\right\}$, where each variable $X_{i,k}$ has a finite range $\mathcal{X}_{i,k} = \left\{x_{i,k}^1, ...x_{i,k}^{N_k}\right\}$.

The entropy reduction will represent the information value brought by a new set of measurements M_i. As mentioned above, it is important to note that the actual entropy reduction $\Delta\hat{H}(Y_i; M_i|\mathcal{F}_i)$ can not be determined without knowing the result of the measurement set M_i. Instead, the expected entropy reduction is applied to address this issue. Given a set of prior knowledge $\mathcal{F}_i$, the information value of a target is represented in terms of the EER brought by measuring all of its remaining features

$$EER(\mathcal{F}_i) = \begin{cases} H(Y_i|\mathcal{F}_i) - E_{M_i}H(Y_i|\mathcal{F}_i, M_i) & \text{if } l_i < L \\ 0 & \text{if } l_i = L \end{cases} \tag{3.3}$$

where $E_{M_i}H(Y_i|\mathcal{F}_i, M_i) = \sum_{k=l_i+1}^{L}\sum_{q=1}^{N_k}[H(Y_i|x_{i,k} = x_{i,k}^q)P(x_{i,k} = x_{i,k}^q|\mathcal{F}_i)]$. The conditional entropy $H(Y_i|\mathcal{F}_i)$ and $H(Y_i|x_{i,k} = x_{i,k}^q)$ are computed using the definition in Eq.3.2 and the posterior PMF introduced in Eq.2.3. All the probabilities can be acquired using the BN CPTs.

The path planning thus needs to find a path that allows collection of information value, or conversely, allows the reduction of entropy by measuring target features. The workspace is populated with r targets, each with a set of revealed features $\mathcal{F}_i$. By the above information-theoretic function via EER, the sensing objective of the path planning can be formulated by that

$$\sum_{i=1}^{r} EER(\mathcal{F}_i) = 0 \tag{3.4}$$

Then, the planner will firstly identify a subset of all targets, whose unrevealed features may still allow further entropy reduction that

$$T_I = \{\mathcal{T}_i \in T : EER(\mathcal{F}_i) \neq 0\} \tag{3.5}$$

so that the goal is to measure all the features of the targets in T_I. Then the objective described by Eq. 3.4 is transformed into

$$\sum_{\mathcal{T}_i \in T_I} EER(\mathcal{F}_i) = 0 \tag{3.6}$$

In the following few sections, the path planning methodology will focus on planning a path that fulfills the goal in Eq. 3.6 with a minimum time cost.

3.2 Directional C-targets

As introduced in the previous subsection, the planner determines a set of targets T_I, each provides additional information value. The next task for the planner is to specify a path to measure each target $\mathcal{T}_i \in T_I$. For a directional sensor with a bounded field-of-view $\mathcal{S}$, $\mathcal{S}$ must intersect with the target $\mathcal{T}_i$ to make measurements. For the purpose of path planning, each target $\mathcal{T}_i$ in the workspace is mapped to the robot's configuration space $\mathcal{C}$ as its corresponding target configuration (C-target), defined by $\mathcal{CT}_i = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{S}(\mathbf{q}) \cap \mathcal{T}_i \neq \emptyset\}$. With the obtained C-target, the path planning goal can be specified such that: for every target $\mathcal{T}_i \in T_I$, there is at least one configuration $\mathbf{q}_i$ in the path that falls in the corresponding C-target region, i.e., $\mathbf{q}_i \in \mathcal{CT}_i$.

This subsection focuses on the construction of each C-target $\mathcal{CT}_i$ for a directional sensor with the presence of obstacles. A directional sensor can only obtain measurements at specific preferred direction. Thus, when determining the C-target, the orientation of the robotic sensor is as important as its coordinate. In addition, ensuring target visibility is also a critical yet challenging task, since a target of interest may be occluded by obstacles in the sensor's line-of-sight.

Consider a target, $\mathcal{T}$, whose coordinate in the workspace is described by a position vector $\mathbf{x}_T \triangleq [x\ y]^T$. Firstly, to ensure a target is visible, its coordinate must fall into the sensor's field-of-view (FOV). More specifically, consider the 2-dimensional sensor FOV $\mathcal{S}$ applied in this thesis. Its geometry is characterized by a maximum open angle α with respect to the sensor's orientation, a minimum range r_{min}, and a maximum range r_{max}, as demonstrated in Figure 3.1. Let $\mathbf{x}$ denote the position of the robot's geometry center ($\mathcal{O}_\mathcal{A}$) in inertial frame $\mathcal{F}_\mathcal{W}$. The relative position of the target with respect to $\mathcal{F}_\mathcal{A}$, a moving Cartesian frame embedded in the sensor, is $\mathbf{r}_T = (\mathbf{x}_T - \mathbf{x})$.

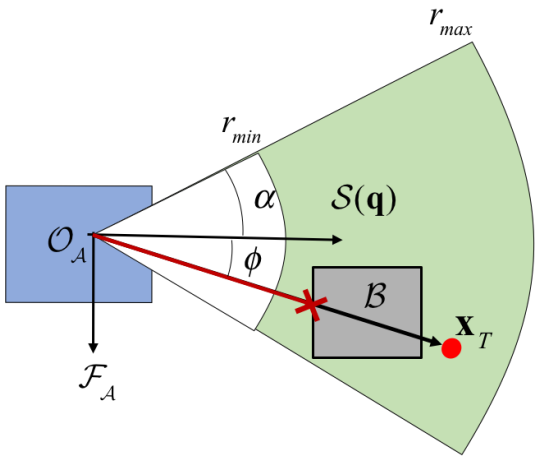


Figure 3.1: The FOV of a sensor at configuration $\mathbf{q}$ is $\mathcal{S}(\mathbf{q})$, a subset of the configuration space. The FOV has a minimum and maximum range, r_{min} and r_{max}, and a maximum open angle α. In this example, although the target (red dot) with coordinate $\mathbf{x}_T$ is in the FOV, it is not in the line-of-sight of the sensor. Therefore, the robot current configuration does not belong to the C-target region of this target.

The angle between the sensor orientation and $\mathbf{r}_T$ is denoted as ϕ. The target's distance from the sensor is the *target range* $\|\mathbf{r}_T\|$. To ensure that a target's coordinate is in the sensor's FOV, the value of target range must be within the minimum and maximum sensor detection range. and ϕ must be within the

maximum open angle

$$r_{min} \leqslant \|\mathbf{r}_T\| \leqslant r_{max} \quad \text{and} \quad |\phi| \leqslant \alpha \tag{3.7}$$

In addition, line-of-sight (LoS) visibility is another important constraint that must be taken into account. Successful measurement of a target is made only when the target is not occluded by any obstacles in the sensor's LoS, as described in Chapter 2, Equation 2.2. That is, the line segment between $\mathbf{x}$ and $\mathbf{x}_T$ does not intersect with any obstacle $\mathcal{B}_i \in B$.

3.2.1 C-target with obstacle avoidance

In a workspace populated with obstacles $B = \{\mathcal{B}_i, ..., \mathcal{B}_n\}$, obstacle avoidance is another fundamental requirement for the path planning. To achieve this objective, the path planning must be performed in the free configuration space $\mathcal{C}_{free}$, defined as the complement of the C-obstacle region in all possible robot configurations $\mathcal{C}$.

A C-obstacle is also a subset of $\mathcal{C}$ where the robot has collisions with at least one obstacles,

$$\mathcal{CB}_i = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap \mathcal{B}_i \neq \emptyset\} \tag{3.8}$$

where $\mathcal{A}(\mathbf{q})$ denotes the subset of $\mathcal{W}$ occupied by the robot geometry $\mathcal{A}$ at configuration $\mathbf{q}$. The union of such C-obstacle is called C-obstacle region, i.e. $\bigcup_{i=1}^{n} \mathcal{CB}_i$. And the complement set of all C-obstacle regions is called the free configuration space

$$\mathcal{C}_{free} = \mathcal{C} \setminus \bigcup_{i=1}^{n} \mathcal{CB}_i = \{\mathbf{q} \in \mathcal{C} \mid \mathcal{A}(\mathbf{q}) \cap (\bigcup_{i=1}^{n} \mathcal{B}_i) = \emptyset\} \tag{3.9}$$

Any configuration in $\mathcal{C}_{free}$ is called a free configuration [26].

As introduced at the beginning of the chapter, to reduce the algorithm complexity, the path planning approach minimizes only the translational distance until the step of waypoint optimization. C-target pruning (Chapter 3.3) and the graph representation of C-target (Chapter 3.4) are both performed on the 2-dimensional projection of the configuration space. Each coordinate $\mathbf{x} \triangleq [x\ y]^T$ in the 2-dimensional free configuration should allow the robot to freely rotation, i.e., $\theta \in (-\pi, \pi)$. With this property, given a determined translation trajectory, the robot can rotate freely to a desired orientation when it needs to make target measurements. Under this consideration, a special technique is applied to the construction of $\mathcal{C}_{free}$.

A commonly used technique to reduce a configuration space dimension, such as θ, is to replace the robot platform $\mathcal{A}$ by the surface it sweeps out when it moves along the independent axis of θ [26]. The robot platform geometry $\mathcal{A}$ is rigid and non-elongated, such as a rectangular with a pair of similar width and length and a geometric center $\mathcal{O}_{\mathcal{A}}$. Then, such $\mathcal{A}$ can be replaced by a disc $\mathcal{A}'$ bounded by the minimum spanning circle of $\mathcal{A}$, as demonstrated in Figure 3.2. Let the free configuration in $\mathbf{R}^2$ to be $\mathcal{C}'_{free}$. Similar to the idea in Equation 3.9, when centered at each coordinate $\mathbf{x} \in \mathcal{C}'_{free}$, $\mathcal{A}'$ must not intersect with any obstacles, i.e., $\mathcal{A}'(\mathbf{x}) \cap \bigcup_{i=1}^{n} \mathcal{B}_i = \emptyset$. Then, at each robot orientation θ, the robot coordinate $\mathbf{x}$ should be limited in the $\mathcal{C}'_{free}$, which is also represented by a cross-section of $\mathcal{C}_{free}$ perpendicular to the θ axis.

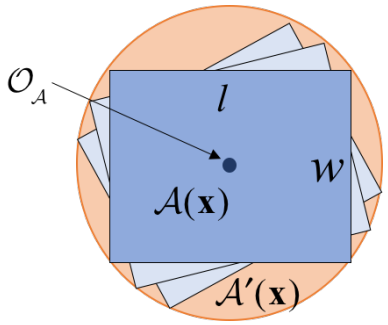
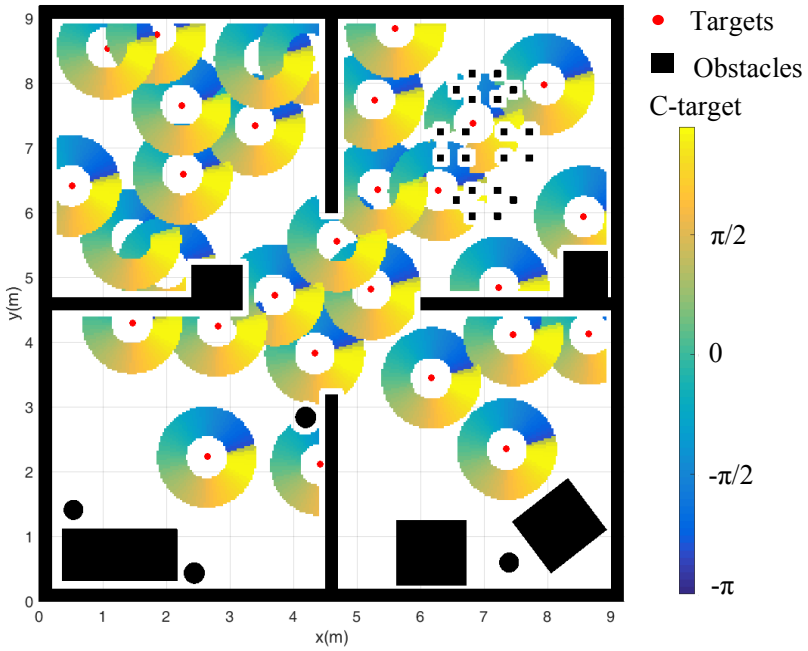
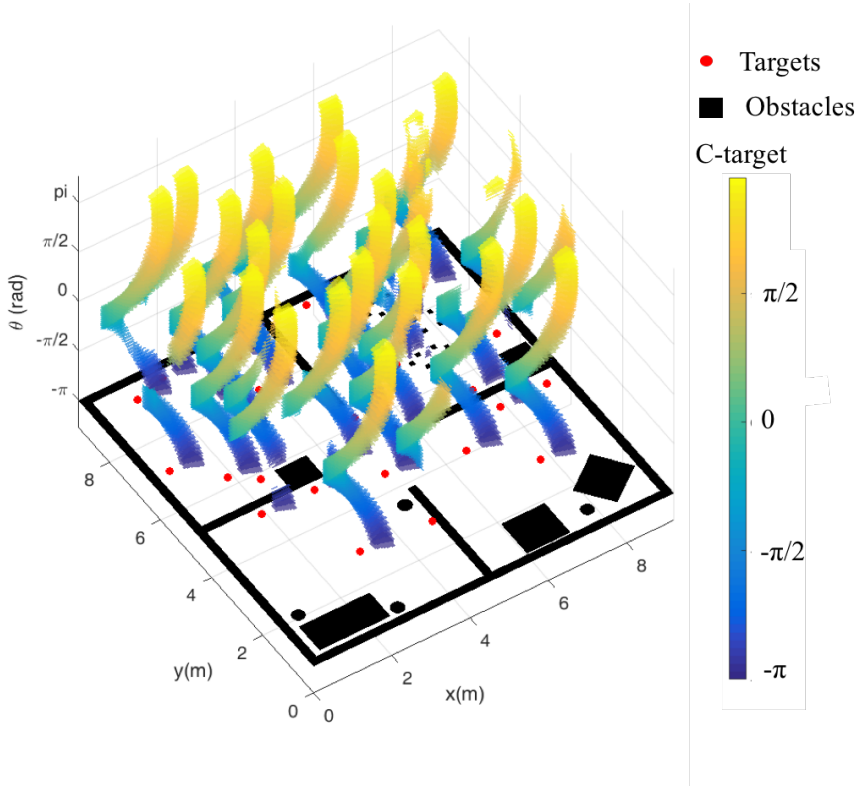


Figure 3.2: The robotic platform $\mathcal{A}$ is a rectangle with width w and length l, with geometry center at $\mathcal{O}_{\mathcal{A}}$. The disc (orange circle), $\mathcal{A}'$, is obtained by sweeping $\mathcal{A}$ around $\mathcal{O}_{\mathcal{A}}$ with all possible orientation. If at this coordinate, $\mathcal{A}'$ is collision-free with all the obstacles, such coordinate belongs to the 2-dimensional C-free $\mathcal{C}'_{free}$.



(a) $x - y$ view



(b) isometric view

Figure 3.3: Demonstration of $\mathcal{C}_{target}$ in 3-D configuration space. The test workspace is the Webots® immersive environment with 30 targets.

3.3 Pruning of C-target regions

In the previous subsection, each 2-dimension C-target region for $\mathcal{T}_i$ is determined as $\mathcal{CT}'_i$, as shown in Figure 3.3(a). With a temporary reduction of the third dimension θ, the planner is able to achieve a high planning efficiency by firstly obtaining a path with the minimum translation distance.

Consider the robot's path as a sequence of waypoint $\mathcal{P} = \langle \mathbf{x}_1, ..., \mathbf{x}_P \rangle$ that allows the sensor to obtain measurements from $\mathcal{CT}'_i$ for every target $\mathcal{T}_i$. The sensor rotate to a proper orientation at a waypoint in order to make target measurements. There are two types of waypoints. An *observation waypoint* is a waypoint in one of the C-target regions at which the sensor can make target measurements. The ith *observation waypoint* in $\mathcal{P}$ is denoted as $\mathbf{x}_o^i$. A *connecting waypoint* is to help to build a feasible path between a pair of observation waypoints, since sometimes there does not exist a straight line path in the free space to connect them. It is assumed that sensor measurements are not made on the connecting waypoints. The kth connecting waypoint in the local path between $\mathbf{x}_o^i$ and $\mathbf{x}_o^j$ is denoted by $\mathbf{x}_{(i,j)}^k$. The local connecting path is determined using the approach introduced in Chapter 3.4.

An observation waypoint needs to be placed in each C-target region to obtain target measurements. It is interesting to note that when the targets are close to each other, there appear to be abundant intersections between the C-target regions. At a waypoint located in one of such intersections, the sensor is able to observe multiple targets when oriented at certain angles. Therefore, by the assumption that rotation time cost is overall less than translation time cost, placing waypoints in such regions is very likely to improve the path efficiency.

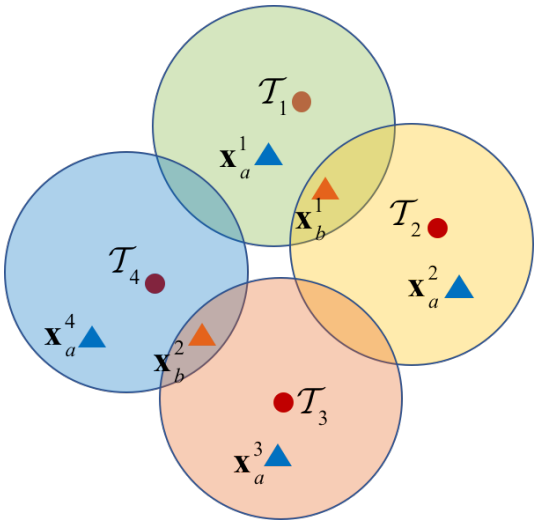


Figure 3.4: A demonstration of the case where the number of waypoints needed is less than the number of C-target regions. To make measurements on four of the targets (red dot), a possible path is to visit waypoints sequence $\mathcal{P}_1 = \langle \mathbf{x}_a^1, \mathbf{x}_a^2, \mathbf{x}_a^3, \mathbf{x}_a^4 \rangle$ (blue triangles), another is to visit $\mathcal{P}_2 = \langle \mathbf{x}_b^1, \mathbf{x}_b^2 \rangle$ (orange triangles). The second path is more likely to have a lower cost.

Thus in this subsection, a set of pruning operation on the C-target region that still fulfills the sensing objective is introduced. The pruning operation means to decrease the number of C-target region needed to be traversed, so that the resulting path is more likely to have smaller cost with fewer waypoints needed. Meanwhile, it may also reduce the size of each C-target region, resulting in a smaller search space thus less runtime in the optimization step.

In order to perform pruning, the original C-target regions obtained in the previous section need to be decomposed into another set of non-overlap *independent C-targets* G_I. Each independent C-target $\mathcal{G}_i \in G_I$ is a set of coordinates that share a distinct set of observable targets, which is represented using a mapping function as follows.

Recall the target set T_I that is used to define the original C-target regions, such that measurement from each $\mathcal{T}_i \in T_I$ must be obtained. The power set of

T_I, denoted as $\mathcal{P}(T_I)$, is the set of all subsets of T_I, including the empty set and T_I itself. A mapping $f : \mathcal{P}(T_I) \mapsto \mathbb{R}^{N_{T_I}}$ is used to distinctly label each element in this power set, where N_{T_I} is the cardinality of set T_I. The range of the mapping function is a finite set of vectors, denoted as $\mathcal{R}_{\mathcal{A}} = \{r_1, r_2, ..., r_{n_R}\}$. Then, a set of targets $A \subseteq T_I$ can be labeled by

$$f(A) = \mathbf{a} = (a_1, ..., a_{N_{T_I}}) \tag{3.10}$$

where each element in vector **a** is an indicator that represents whether a target $\mathcal{T}_i \in T_I$ is in the target set by

$$a_i = \begin{cases} 1 & \text{if } \mathcal{T}_i \in A \\ 0 & \text{if } \mathcal{T}_i \notin A \end{cases} \tag{3.11}$$

Denote $\mathcal{TC}(\mathbf{x})$ as the set of targets that the robot can observe at a coordinate **x** in the free space. Then an independent C-target region $\mathcal{G}_i$ is a set of coordinates that

$$\mathcal{G}_i = \{\mathbf{x} : f(\mathcal{TC}(\mathbf{x})) = r_i\} \tag{3.12}$$

and a $\mathcal{G}_i$ also has the property that $\mathcal{TS}(\mathcal{G}_i) = r_i$, where notation $\mathcal{TS}(\mathcal{G}_i)$ denotes the set of targets that can be measured on every coordinate in $\mathcal{G}_i$.

The pruning operation will make attempts to remove an independent C-target region, if doing so still allows measurements from every target in T_I. That is, during the operation, $\bigcup_{i=1}^{N_P} \mathcal{TS}(\mathcal{G}_i) = T_I$ is maintained for the remaining N_P independent C-target regions. The proposed pruning approaches can be described as the following three steps.

The approach first starts with the simplest case to capture the intersection regions of two original C-targets. It is labeled as case 1 (Figure 3.5), where a

region $\mathcal{G}_a$ is marked as redundant and should be pruned if

$$\exists\, \mathcal{G}_b \mid \mathcal{TS}(\mathcal{G}_a) \subseteq \mathcal{TS}(\mathcal{G}_b) \tag{3.13}$$

so that by visiting $\mathcal{G}_b$, one can observe a target set $\mathcal{TS}(\mathcal{G}_b)$ that contains all of the targets in $\mathcal{TS}(\mathcal{G}_a)$. Thus, visiting $\mathcal{G}_a$ is unnecessary.

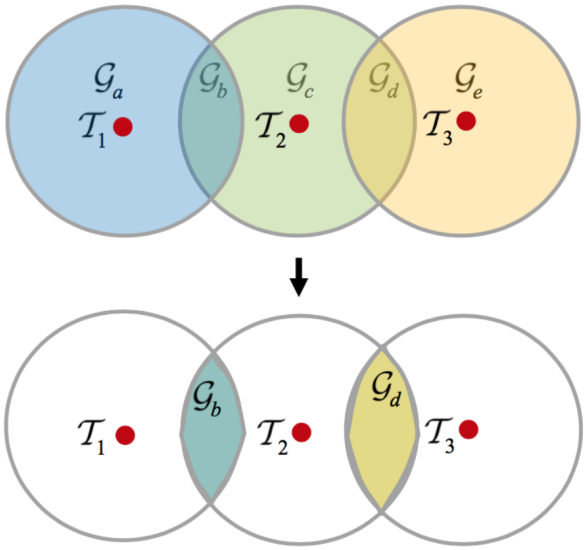


Figure 3.5: The first step of C-target pruning approach. C-target regions $\mathcal{G}_a$, $\mathcal{G}_c$, $\mathcal{G}_e$ are marked as redundant and should be pruned. The remained regions are $\mathcal{G}_b$ and $\mathcal{G}_d$, by visiting which the three targets, $\mathcal{T}_1$,$\mathcal{T}_2$ and $\mathcal{T}_3$ can all be observed.

After the first step, the regions remained are the intersections of the original C-target regions. Yet there remain other cases with redundancy to be dealt with. As observed, redundancy commonly appears when more than three C-targets are chained. The redundancy with chained C-target can be divided into two categories: with a cycle that is labeled as case 2 (Figure 3.6), and without cycle labeled as case 3 (Figure 3.7). The pruning on case 2 attempts to remove a set of regions together, thus it must be dealt with before case 3, where the pruning checks for redundancy on every single region.

For case 2, one challenging problem is that two sets of regions cover the exact same set of targets. When the number of regions in each set is different, the one with fewer regions is preferred, so that the number of waypoints needed is likely to be smaller too. When the two sets have the same number of regions, the planner greedily selects the pair of C-target regions with a larger size (Figure 3.6). This approach has demonstrated its good performance with a similar case introduced in [18]. Although as a tradeoff, selecting the set with larger regions may lead to runtime increase in the optimization step, this approach provides more flexibility in the waypoint selection thus possibly a solution with lower cost. In summary, a set of C-target regions are redundant and should be pruned if

$$\exists \{\mathcal{G}_{b1}, ..., \mathcal{G}_{bm}\} \mid \bigcup_{i=1}^{n} \mathcal{TS}(\mathcal{G}_{ai}) \subset \bigcup_{j=1}^{m} \mathcal{TS}(\mathcal{G}_{bj})$$

or

$$\exists \{\mathcal{G}_{b1}, ..., \mathcal{G}_{bm}\} \mid \bigcup_{i=1}^{n} \mathcal{TS}(\mathcal{G}_{ai}) = \bigcup_{j=1}^{m} \mathcal{TS}(\mathcal{G}_{bj}) \text{ and } n > m$$

or

$$\exists \{\mathcal{G}_{b1}, ..., \mathcal{G}_{bm}\} \mid \bigcup_{i=1}^{n} \mathcal{TS}(\mathcal{G}_{ai}) = \bigcup_{j=1}^{m} \mathcal{TS}(\mathcal{G}_{bj}),\ n = m \text{ and } \sum_{i=1}^{n} |\mathcal{G}_{ai}| < \sum_{j=1}^{m} |\mathcal{G}_{bj}| \tag{3.14}$$

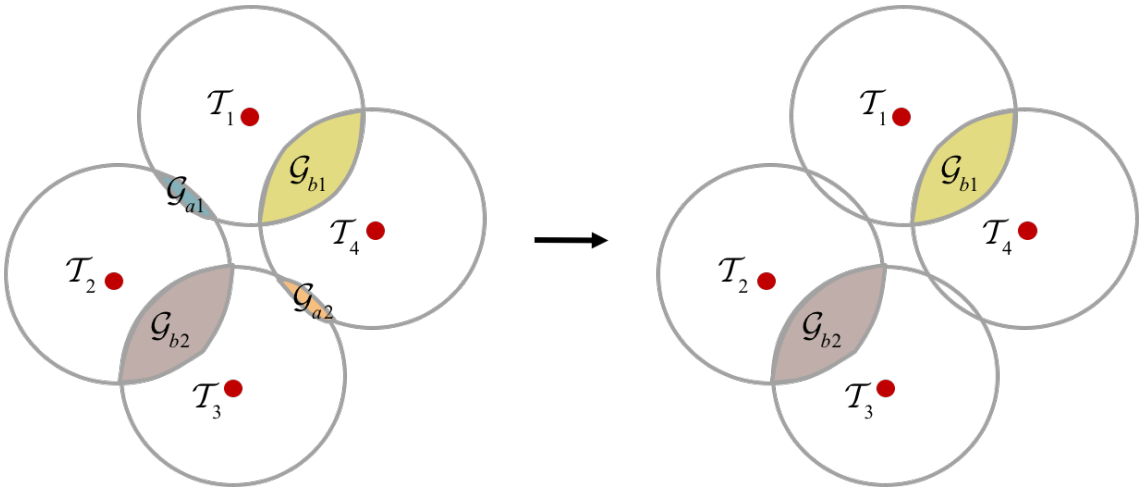


Figure 3.6: Two pairs of independent C-target regions, each contains two regions. The two pairs can observe the same set of targets. The pruning approach selects the pair that contains a larger number of coordinates. By visiting the selected regions, the four targets, $\mathcal{T}_1$,$\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$ can all be observed.

The third case is similar to the first, where only two C-targets are involved (Figure 3.5). Pruning of case 3 checks for redundancy in a single independent C-target by whether its corresponding target set can be covered by a set of other regions. As shown in Figure 3.7, a set of C-target regions are redundant and should be pruned if

$$\exists \{\mathcal{G}_{b1}, ..., \mathcal{G}_{bm}\} \, | \mathcal{TS}(\mathcal{G}_a) \subseteq \bigcup_{j=1}^{m} \mathcal{TS}(\mathcal{G}_{bj}) \tag{3.15}$$

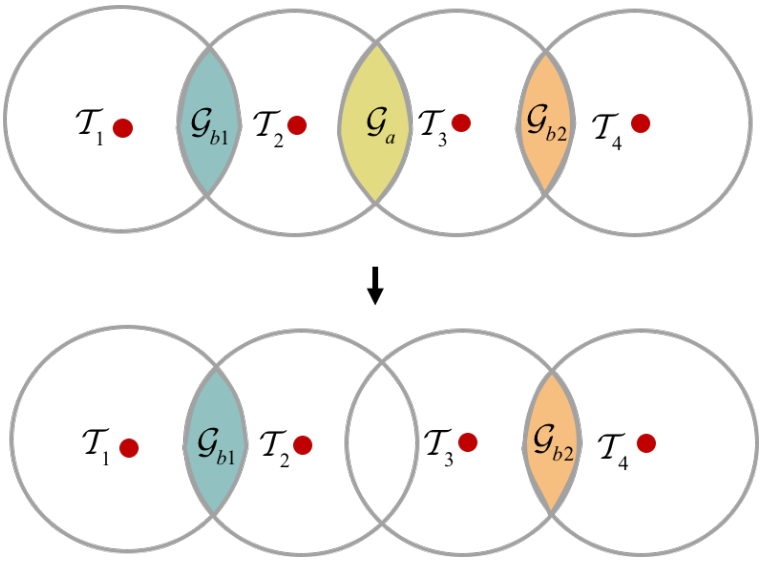
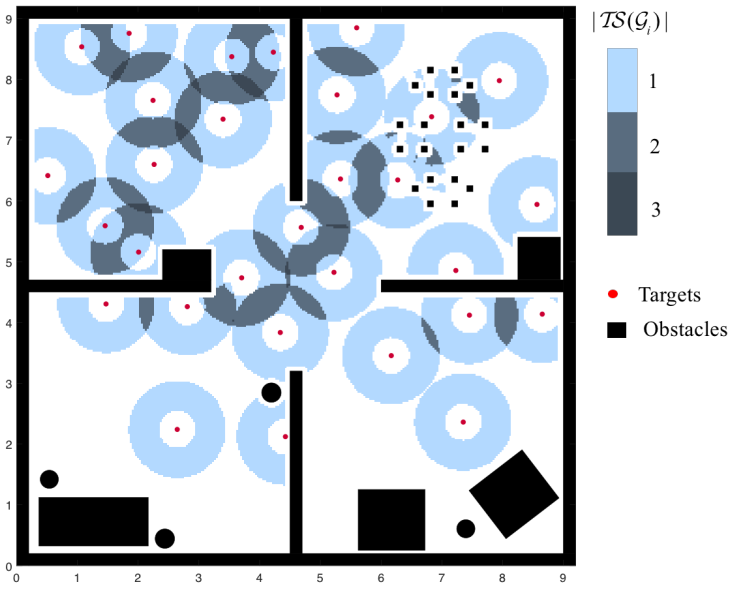
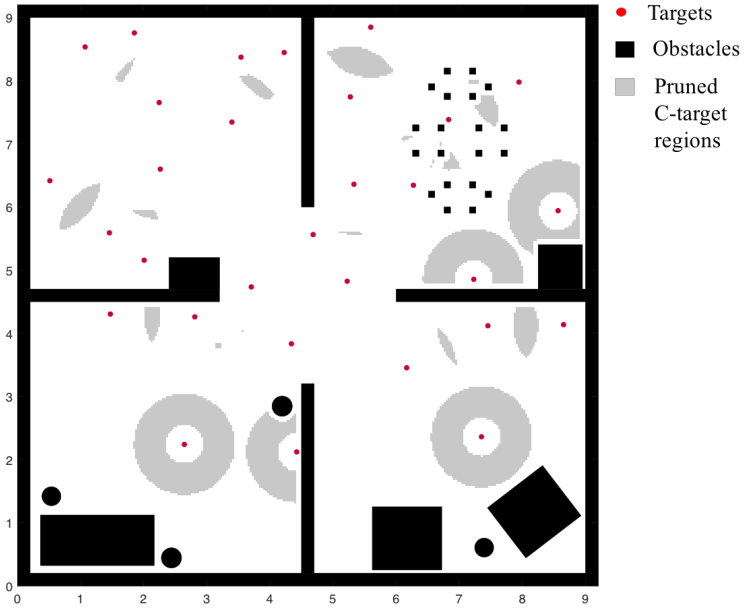


Figure 3.7: For the chained C-target regions, the pruning approach first selects the intersecting regions as described in case 1. Then, at the third step, the approach further checks the redundancy and discards unnecessary regions if possible. By visiting the selected regions, the four targets, $\mathcal{T}_1$,$\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$ can all be observed.

The three-step C-target pruning approach performed so far can deal with most of the commonly seem cases of C-target redundancy in this problem, and more pruning techniques can be applied as future works. The result set of independent C-target regions after the pruning is denoted as $P = \{\mathcal{G}_1, ..., \mathcal{G}_{N_P}\}$. By visiting each region, the sensor can obtain measurements from every target in T_I.



(a) The test map has 30 targets. Thus there are 30 original C-target regions, denoted in blue. The color for each independent regions indicates the number of targets that can be observed from any coordinate in the region.



(b) After the pruning, there remain 17 C-target regions, some with smaller size compared to the original ones.

Figure 3.8: The effect of C-target pruning operation on the test workspace for satisficing experiments performed in Webots®. The operation greatly reduces the number of C-target regions needed to be traversed, allowing an improvement of planning efficiency.

3.4 Traversal of C-target with minimum translation

With the pruned independent C-target regions $P = \{\mathcal{G}_1, ..., \mathcal{G}_{N_P}\}$ obtained in the previous section, the next task is to determine the actual waypoints needed to traverse the elements in P with the minimum translation cost. Some crutial issues need to be discussed to accomplish this goal, including the order of visiting these C-target regions and the selection of waypoint in each of them. It is also important to note that the placement of a waypoint may be highly influenced by previous trajectory, which depends on the waypoint position in the regions previously visited. Thus, the traversal order of C-target needs to be addressed first.

Traveling salesman problem (TSP) representation is a convenient and efficient problem abstraction when seeking for a traversal solution. The problem requires a line traversal solution with minimum cost, often defined by translational distance, that visits the targets each exactly once. Abundant existing techniques are demonstrated to provide high-performance TSP solutions in a reasonable runtime, including 2-opt heuristics [33], greedy heuristic algorithm [34], genetic algorithm [33, 35], and ant colony algorithm [33,36,37].

Although the TSP representation has the potential of providing efficient solutions, the proposed problem has some essential properties so that special techniques are required when transformed into a TSP. In the proposed problem, the solution requires visiting at least a point in the target "neighborhoods", each is a C-target region that consists of one or multiple points. However, with the presence of obstacles in the workspace, the problem is by nature very different

from the traditional traveling salesman problem with neighborhood (TSPN). Thus its solutions cannot be directly applied. Instead, to be introduced in Section 3.4.1, the proposed method provides techniques to describe the C-target regions by nodes in a TSP connectivity graph. In addition, traditional TSP algorithms require that the connectivity graph is complete, meaning that each pair of nodes is connected by an edge. The task is challenging due to the uncertainty of finding a direct path between C-targets with the presence of obstacles. Section 3.4.2 thus focuses on the edge assignment in a complete connectivity graph that can provide a proper estimation of the pairwise distance between C-target regions.

3.4.1 C-target connectivity graph

As a combinatorial optimization problem, TSP is often modeled as an undirected weighted graph $G = \{\mathcal{V}, \mathcal{E}\}$. For the proposed problem, a node $v_i \in \mathcal{V}$ represents a pruned C-target region $\mathcal{G}_i \in P$, and the edge set $\mathcal{E} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$ measures the distance between any unordered pair of regions. The graph is weighted, meaning that for each arc between an unordered pair of nodes (v_i, v_j), a weight is assigned by a function $w(v_i, v_j)$, which will be discussed in more details in the next section. The graph is also undirected, since traveling in both direction between nodes is allowed, and has the same cost.

Each node in the graph represents a C-target region $\mathcal{G}_i \in P$, which is a set of coordinates. Due to the presence of obstacles and the sensor's FOV geometry, a C-target region is not necessarily a connected area or a convex polygon. Also,

some methods that measure the Euclidean distance between points sets are not directly applicable, e.g., Euclidean Hausdorff distance method, since they do not take obstacles into account. Instead, a technique in [18] uses the centroid coordinate to represent each region, which is demonstrated as a convenient and reasonable simplification.

Similarly, each C-target region $\mathcal{G}_i$ is associated with a coordinate denoted as $\mathbf{r}_i$, which is used to determine the direct connectivity and distance between node pairs. A pruned C-target region $\mathcal{G}_i$ is pruned from an original C-target $\mathcal{CT}_i$. Constrained within the sensor's FOV, each coordinate in $\mathcal{CT}_i$ is not further than r_{max} away from the target's coordinate, where r_{max}, the sensor's maximum detecting range, is relatively small compared to the workspace size as assumed in the problem formulation. Thus, the region's centroid is a proper candidate for its point representation $\mathbf{r}_i$. Yet it is also important to note that due to its complex geometry, a C-target region $\mathcal{G}_i$ may not necessarily has its centroid located in the free space. In such case, a new coordinate has to be sampled from $\mathcal{G}_i$ (Figure 3.9). Considering that the size of region is relatively small, uniform sampling is able to achieve this goal properly with easy implementation.

In summary, the selection of the point representation for each node is as follows. $\mathbf{r}_i$ is the point representation for C-target region $\mathcal{G}_i$ consisting of N_G coordinates, each with coordinate with respect to the fixed Cartesian frame $\mathcal{F}_\mathcal{W}$ that $\mathcal{G}_i = \{(x_1, y_1), ..., (x_{N_G}, y_{N_G})\}$. The centroid of this region is $\mathbf{C}_i = (\bar{x}_i, \bar{y}_i) = (\frac{1}{N_G}\sum_{i=1}^{N_G} x_i, \frac{1}{N_G}\sum_{i=N_G}^{N_G} y_i)$. Then there is

$$\mathbf{r}_i = \begin{cases} \mathbf{C}_i & \text{if } \mathbf{C}_i \in \mathcal{C}'_{free} \\ \mathbf{x}_i & \text{otherwise} \end{cases} \tag{3.16}$$

where $\mathbf{x}_i$ is a coordinate uniformly sampled from $\mathcal{G}_i$.

Last but not the least in the discussion on graph nodes, the planning problem requires a fixed start. In the graph, a start node should properly represent the given initial position as well. An extra node v_{init} is added to the graph if the initial configuration does not belong to any C-target region (Figure 3.9). Although the extra start node is not treated differently in this graph, target measurements cannot be made on its corresponding coordinate, so it should be properly labeled for convenience of waypoint optimization.

3.4.2 Edge weight for a complete connectivity graph

In the previous subsection, the node representation for C-target regions using the TSP graph is introduced. This subsection discusses the edge cost assignment regarding the problem's special properties. As mentioned before, the graph model for the TSP is undirected, weighted and complete. In order to determine a traversal solution of C-target regions that has the minimum translation cost, the graph needs to provide proper estimation of pairwise traveling distance of the regions. That is, the weight for each arc needs to be assigned properly.

Consider a pair of unordered nodes (v_i, v_j) that associated with coordinates $\mathbf{r}_i = (x_i, y_i)$ and $\mathbf{r}_j = (x_j, y_j)$ accordingly, as defined in the previous subsection. Denote $L_{i,j}$ as the line segment with $\mathbf{r}_i$ and $\mathbf{r}_j$ as its endpoints. If $L_{i,j}$ is collision-free with obstacles, (v_i, v_j) is known as directly connected, and the edge weight is the Euclidean distance between the coordinates, i.e., $w(v_i, v_j) = \|\mathbf{c}_i - \mathbf{c}_j\|$. Unfortunately, the majority of node pairs cannot be directly connected in the problem. One technique to ensure the graph completeness is utilizing existing edges. Any other nodes in the graph may be used as "stepping stones"

to find a free path between an un-connected node pair, with the assistance of Dijkstra's algorithm for example. However, the nodes available to assist the connectivity are highly limited, and the resulting path no longer visits each node exactly once. Thus, this approach may result in a solution with undesirably high path cost, especially for a large workspace with many obstacles and sparse target placement (Figure 3.9).

The methodology thus applies the probabilistic roadmap method (PRM) to provide a better approximation of the distance between C-target regions. PRM is a randomized motion planning algorithm for generating computationally feasible solution in a relatively large workspace that has multiple obstacles, and the given start and goal positions [38]. In a *learning phase*, the PRM planner samples milestones in the free space to construct a roadmap. The roadmap is an undirected graph with a set of milestones $M = \{\mathbf{m}_1, ..., \mathbf{m}_{N_\mathbf{m}}\}$, and a set of arcs $A = \{(\mathbf{m}_i, \mathbf{m}_j) \mid \mathbf{m}_i, \mathbf{m}_j \in M\}$, with each arc representing a straight line path to connect two milestones. The number of PRM nodes in the roadmap is denoted as n_{PRM}, which can be tuned to meet the need of workspaces with different size and complexity. The milestones are uniformly sampled from any possible value in the 2-dimensional free configuration ($\mathcal{C}'_{free}$).

Then in the *query phase*, given a pair of arbitrary start coordinate $\mathbf{c}_0$ and goal coordinate $\mathbf{c}_f$, the planner quickly constructs a feasible path using the roadmap [1, 38]. One commonly used strategy is to connect $\mathbf{c}_0$ to milestones in the roadmap by the order of increasing Euclidean distance. After $\mathbf{c}_0$ is successfully connected to the roadmap, a similar approach is applied to connect $\mathbf{c}_f$ [38]. Note that once a path is found from $\mathbf{c}_a$ to $\mathbf{c}_b$, the path from $\mathbf{c}_b$ to $\mathbf{c}_a$ can be obtained by reversely traversing the same set of milestones. A smoothing algorithm based

on Dijkstra is used to obtain a shorter path. The obtained path is a set of straight lines in the free space linking the start and goal coordinates, stored using a milestone sequence $P_{PRM} = \{\mathbf{m}_1, ..., \mathbf{m}_{N_{ms}}\}$. The distance cost of such a path $d(P_{PRM})$ can be easily calculated as $d(P_{PRM}) = \sum_{k=1}^{N_{ms}-1} \|\mathbf{m}_{k+1} - \mathbf{m}_k\|$, the sum of Euclidean distance of the path line segments.

With the assistance of PRM, the edge weight assignment of the complete connectivity graph can be defined. For a pair of nodes v_i and v_j representing the C-target regions $\mathcal{G}_i$ and $\mathcal{G}_j$, $L_{i,j}$ is the line segment between their corresponding coordinates $\mathbf{r}_i$ and $\mathbf{r}_j$. In summary, the weight of the edge between v_i and v_j is

$$w(v_i, v_j) = \begin{cases} \|\mathbf{c}_i - \mathbf{c}_j\| & \text{if } L_{i,j} \text{ is collision-free} \\ d(P_{PRM}^{i,j}) & \text{otherwise} \end{cases} \tag{3.17}$$

where $P_{PRM}^{(i,j)}$ is the path found by the PRM to connect v_i and v_j. The path obtained is stored as the corresponding connection waypoints between two observation waypoints, as introduced in Section 3.2.

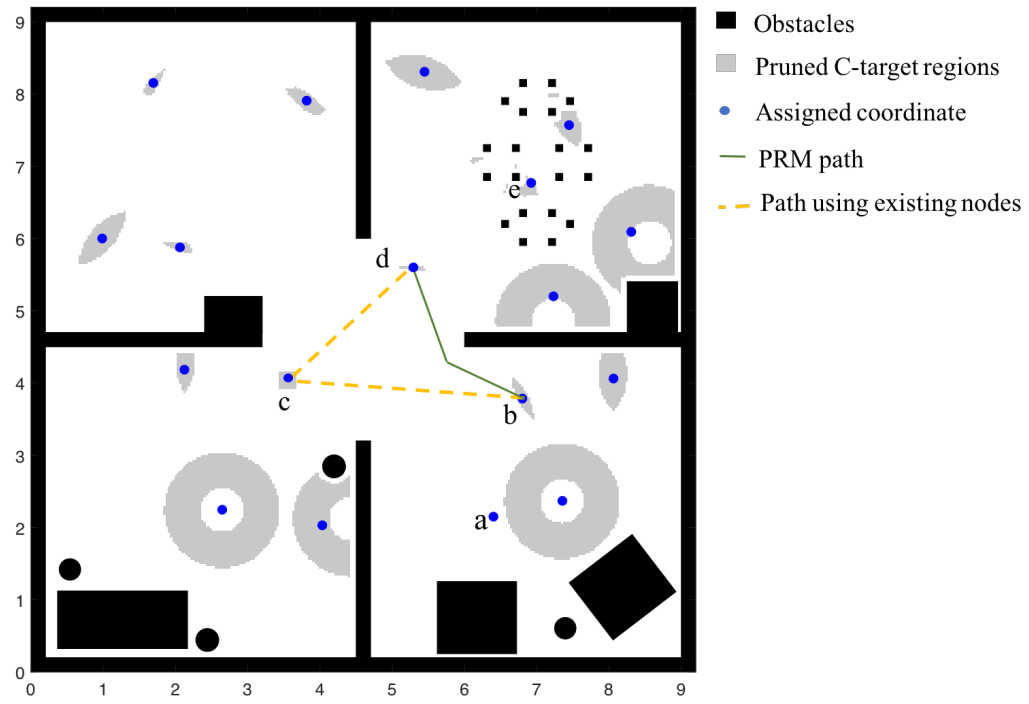


Figure 3.9: An example of the connectivity graph construction on the test workspace. A blue dot represent an assigned coordinate $\mathbf{r}_i$ for each C-target region $\mathcal{G}_i$. Point **a** is the start position. To estimate the distance cost between point **b** and **d**, the approach using existing nodes, such as using point **c**, may result in an inappropriate estimation. Instead, a path generated by PRM is used. Point **e** denotes the coordinate for a region whose geometry is not connected. Instead of its centroid, which does not belong to C'_{free}, an alternative point is sampled from the region.

By this set of representation and approximation, the problem of traversing C-target regions is transformed into a TSP. As introduced at the beginning of the section, there are abundant existing methods to obtain a TSP solution. 2-opt heuristics algorithm, which is also known as pairwise exchange algorithm, is able to provide a fast and good-quality solution [33, 39]. On a complete graph, it is shown to produce a solution no worse than the average cost of a tour in a polynomial number of iterations [39]. It removes two arbitrary edges in the path, then tries to reconnect the path using new edges. If such a move reduces the total path cost, which is defined by the path length, the change is kept [40].

The resulting solution is the optimal order of visiting all the required C-target regions.

3.5 Waypoint optimization of translational and rotational cost

The techniques introduced previously provide an optimal order of visiting each pruned C-target regions. The final step of the path planning is to determine waypoints placements with the minimum cost in terms of translation and rotation. As introduced in Section 3.2, the mobile sensor visits a sequence of waypoint $\mathcal{P} = \langle \mathbf{x}_1, ..., \mathbf{x}_P \rangle$ in the free workspace and takes measurements along the way to fulfill sensing objective. A waypoint is either an observation waypoint where the sensor obtains measurement by rotating to the proper direction, or a connection waypoint that ensures the path connectivity in free space. Each observation waypoint is constrained in its corresponding C-target region, while connection waypoints are constrained in the free space. The performance of a candidate waypoint sequence is evaluated using the path's translational distance d_t and rotational angle d_θ. Each d_θ^i is the minimum rotation required to follow the path and measure all targets

$$d_t = \sum_{i=1}^{n_P-1} \left\| \mathbf{x}_{(i+1)} - \mathbf{x}_i \right\|, \quad d_\theta = \sum_{i=1}^{n_P-1} d_\theta^i \tag{3.18}$$

Then the position of each waypoint is optimized for a minimum total cost J, defined as a weighted combination of translation cost d_t and rotation cost d_θ

$$J = (w_t)d_t + (1 - w_t)d_\theta \tag{3.19}$$

where the weight of translation is w_t, while $(1 - w_t)$ is the weight of rotation correspondingly. The selection of w_t is based on the translational and rotational

velocity of the vehicle, such as $\frac{\omega_m}{v_m+\omega_m}$. When ω_m is very large, the rotational cost is negligible and w_t is selected as 1. If the optimization on rotation is encouraged, a smaller w_t can be applied. The effect of this parameter will be further discussed in the next chapter. The waypoints are optimized using genetic algorithm, and is smoothed by discarding unnecessary intermediate points to further improve the result.

CHAPTER 4

SIMULATIONS AND RESULTS

In this chapter, the path planning approach introduced is tested on a set of workspaces. Simulations are performed to examine the effectiveness of the proposed method and the influence of relevant parameters. The selection of the two parameters in the proposed approach and its corresponding effects will be demonstrated and discussed. The proposed method shows significant performance improvements when compared to existing methods, including the "nearest neighbor" and 2-opt heuristics classical traveling salesman problem (TSP) algorithm, and cell decomposition approach with integer programming optimization.

4.1 Simulation workspaces and sensor parameters

In the simulation, the directional sensor has a FOV as described in Figure 3.1. The detection range is bounded by r_{max} and r_{min}, which are 0.8 meters and 0.3 meters accordingly. The open angle α is $\pi/6$. Those parameters are determined from the sensor applied in the immersive satisficing experiment introduced in Chapter 1, which also supports the onboard computer vision algorithms for target feature extraction.

For a comprehensive evaluation of the proposed algorithm, the three test workspace maps applied have different complexities and characteristics. The first map (Figure 4.1) is relatively simple and small-scale. It contains seven obstacles and nine targets. The majority of target pairs are blocked by obstacles

and cannot be connected by a straight line path.

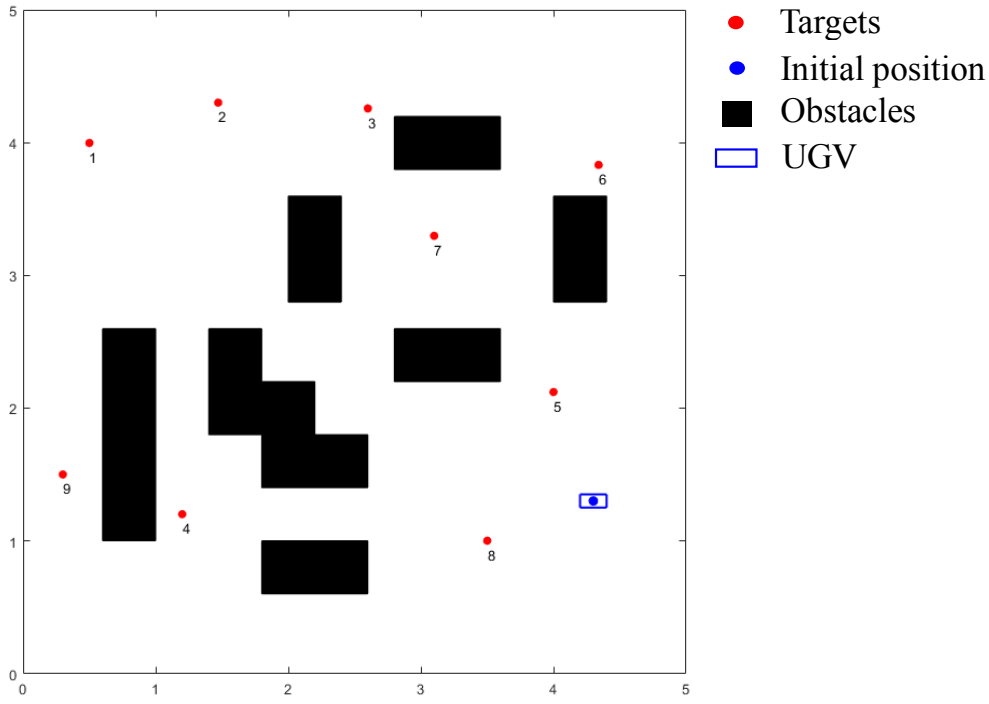


Figure 4.1: Map 1 applied in the simulations. It is relatively small scale, with 9 obstacles and 7 targets.

Map 2 (Figure 4.2) is the abstraction of the environment used in the immersive satisficing experiment. It contains twenty-five obstacles and has a four-room layout, providing simulations of a complex real-life scenario. Some of the obstacles populated have a non-polygon geometry, for example, a circular shape. Thirty targets are placed in the four rooms evenly. A target is placed under chairs and a table, each is represented by its four legs. Map 2 is tested with two sets of targets. The two sets of targets are of different prior knowledge of target features, such that all targets are of non-zero EER in the first set (Figure 4.2), and only 15 targets have non-zero EER in the second set (Figure 4.3).

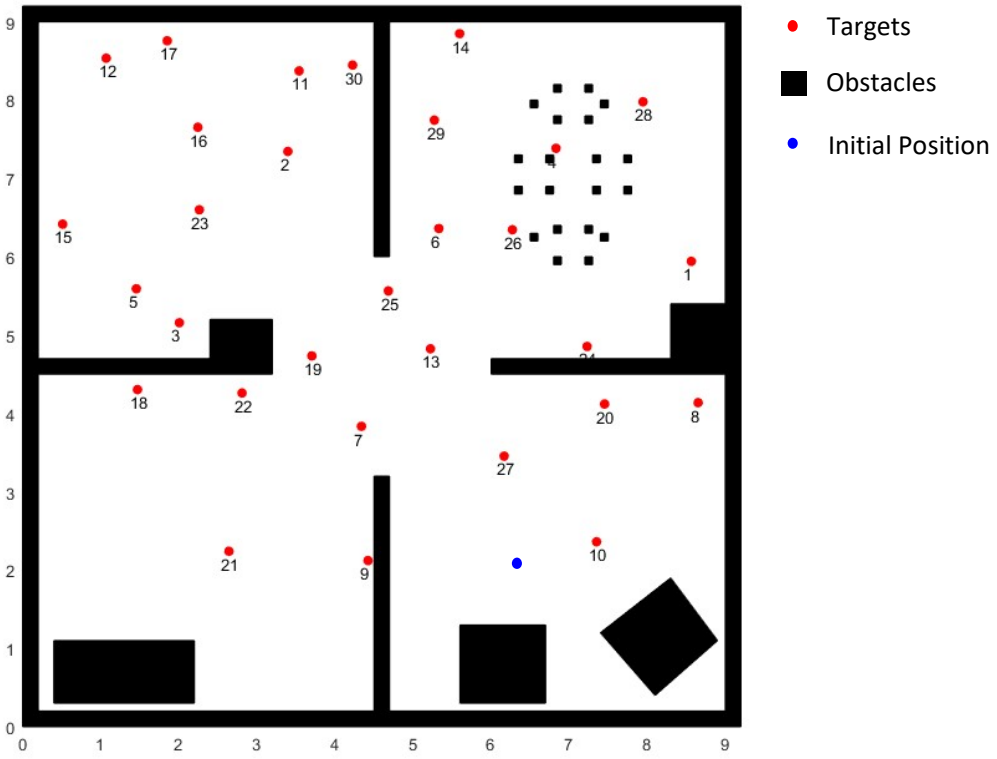


Figure 4.2: Map 2 has 25 obstacles and 30 targets. It is associated with two sets of targets of different prior feature knowledge. In the first set, 30 targets have non-zero EER.

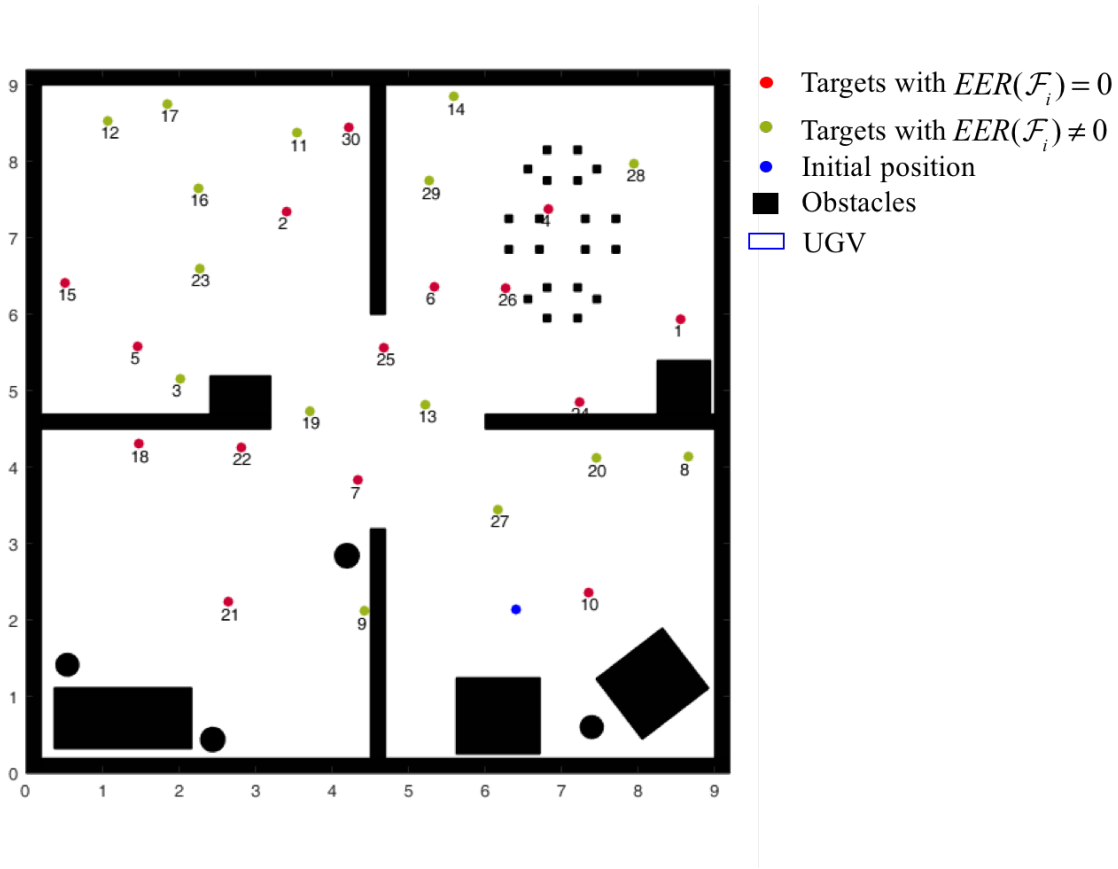


Figure 4.3: Map 2 has 25 obstacles and 30 targets. It is associated with two sets of targets of different prior feature knowledge. In the second set, 15 out of 30 targets have non-zero EER, marked as red dots, while the targets with zero EER are marked by green dots.

Map 3 (Figure 4.4) has a larger dimension, and a more complex maze-like layout. It has 27 targets, placed throughout the workspace. Similar to map 2, this map also contains some circular obstacles. Despite of the large map size, some targets are placed near each other to form a "cluster" , for example, target 22, 23, 24 and 25.

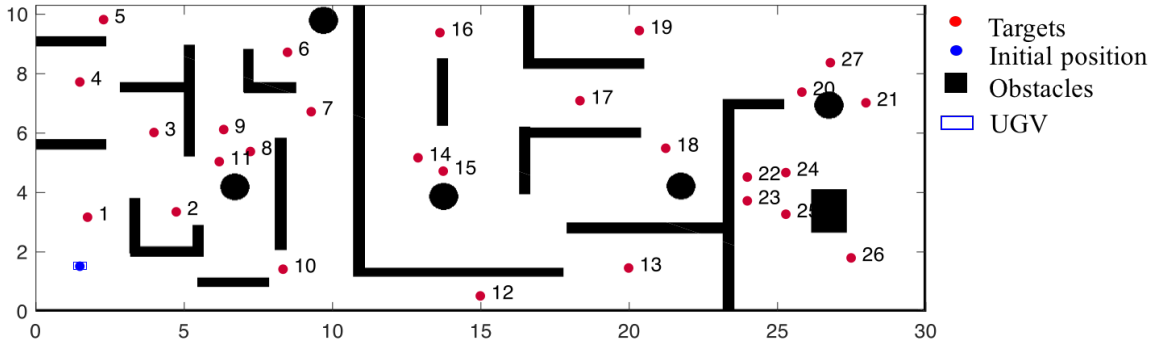


Figure 4.4: Map 3 is a large-scale maze. 27 targets are placed throughout the map. Some targets, such as 22, 23, 24 and 25 are placed near each other to form a "cluster".

The simulations are performed in MATLAB and Webot®, a professional robot simulator that provides modules for sensors, robots, and their interactions with a 3-D virtual environment. The satisficing experiments introduced in Chapter 1, which study the human navigation and decision making strategies, are also performed based on Webot®.

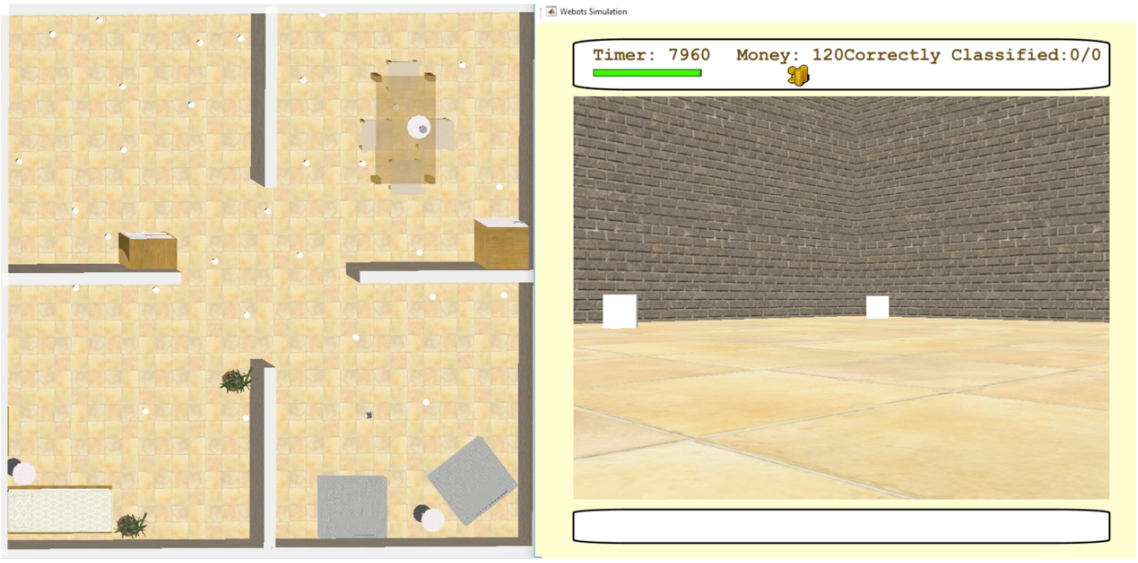


Figure 4.5: The top view of the Webot® immersive environment along with the view of sensor camera. The workspace setup corresponds to the second map introduced above. 30 targets are placed in the environment, and they may subject to different initial observable feature set $\mathcal{F}_i$.

4.2 Influence of important parameters in the planning approach

In the previous chapter, two important parameters are introduced, the selection of whose value may have interesting effects on the obtained solution. This section presents a series of comparisons of solution performance and algorithm computation time on selected workspaces.

The first parameter to be discussed is n_{PRM}, the number of nodes used to construct the PRM roadmap for estimating the edge weight in the TSP representation. As presented in Figure 4.6, if the n_{PRM} used is far less than necessary, the method may fail to achieve the solution with minimum cost, since the edge weights assigned in the connectivity graph cannot represent the pairwise distance between targets well. On the other hand, a tradeoff is expected between the solution performance and the computation time. Therefore, the final translational distance and the computational time spent on constructing connectivity graph via PRM are plotted against a group of n_{PRM} (Figure 4.7). The time for edge assignment approximates the time consumption of constructing the roadmap and answering several queries. The experiment is performed on map 2, since it can provide a moderate workspace complexity without taking too much computational time. The solution performance is evaluated using the total translation distance when the weight on translation w_t is 1, meaning that translational cost dominates and the rotational cost is negligible.

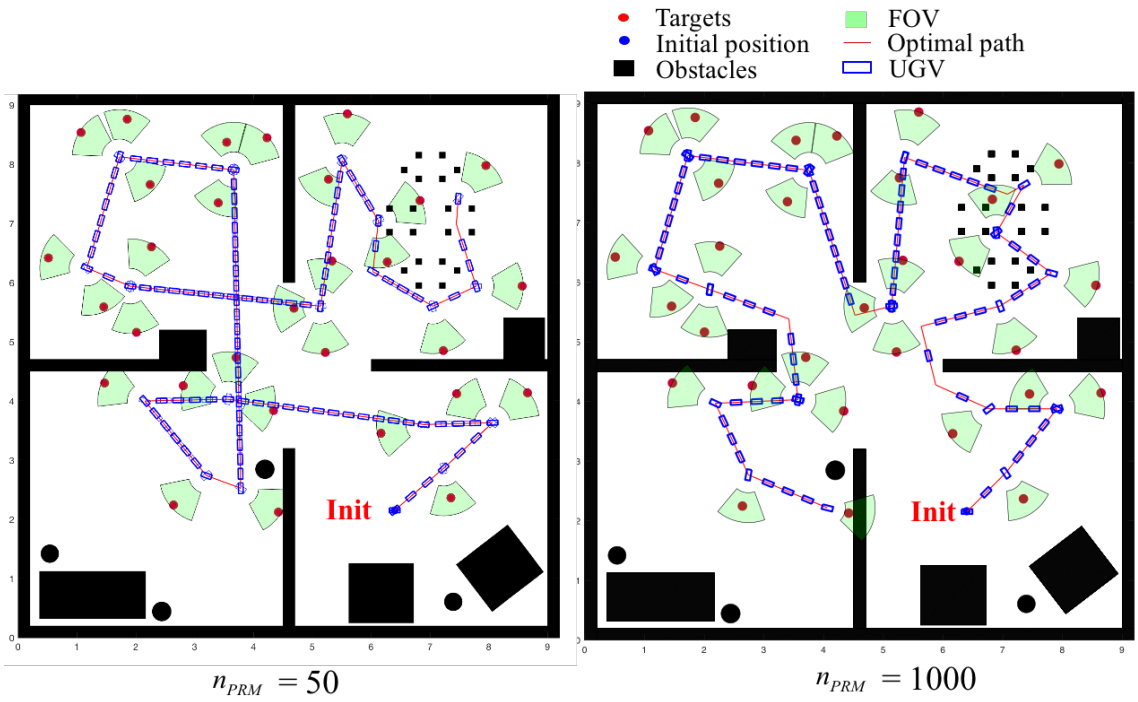


Figure 4.6: A demonstration of the possible effect of n_{PRM} on map 2. The plot on the left has a number of nodes far less than needed given the map complexity. Thus, the order of traversing the C-targets with the minimum cost may not be obtained.

Overall, with the increase of n_{PRM}, a decreasing trend is observed in the translational distance and a significant increase in the PRM runtime. The result shows that a finer roadmap density provides a better approximation of pairwise distance in the connectivity graph, thus leads to a better solution performance. The proper selection of n_{PRM} according to the map complexity is highly encouraged in order to obtain a satisfactory solution with a moderate runtime.

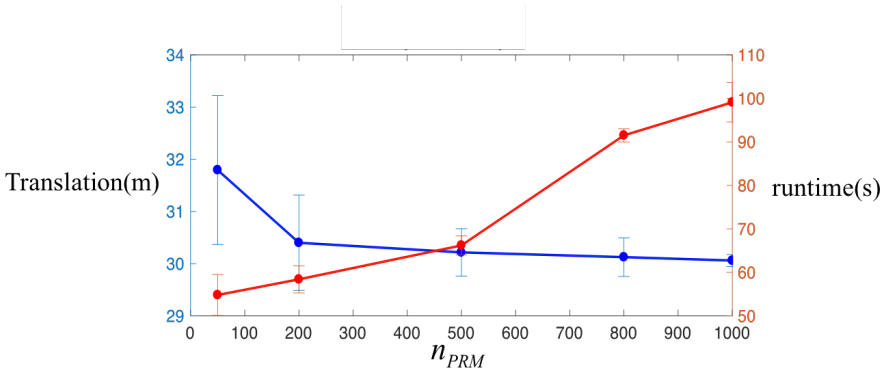


Figure 4.7: A summary of the effect of n_{PRM} on map 2. The result is evaluated in terms of the translational distance, plotted in blue with a scale on the left axis. Their corresponding time consumptions of edge assignments via PRM is also plotted in red. The standard deviation obtained from the experiment is shown by the error bars.

The second important parameter is w_t, the weight on translation in the waypoint optimization. Although the cost of rotation is not considered in the earlier steps of the proposed method, it can still be included in the waypoint optimization step to adapt to the problem specifications. A set of experiments are performed for the first two maps to examine the effect of w_t on translational distance and rotational angles in the solution (Figure 4.8). For each map, the n_{PRM} is fixed and set as the best fit for the corresponding map complexity. A larger translation weight indicates a higher cost associated with translation in the cost function to be optimized. Therefore, the optimization will encourage the placement of a waypoint that helps reducing rotation with a sacrifice of translation cost if necessary. Overall, such trend is observed in both maps. However, since the algorithm has determined the C-target visiting order in the previous steps, the ability to reduce rotational cost is limited. For example in map 2, even with a w_t less than 0.5, the change in rotational angles is trivial.

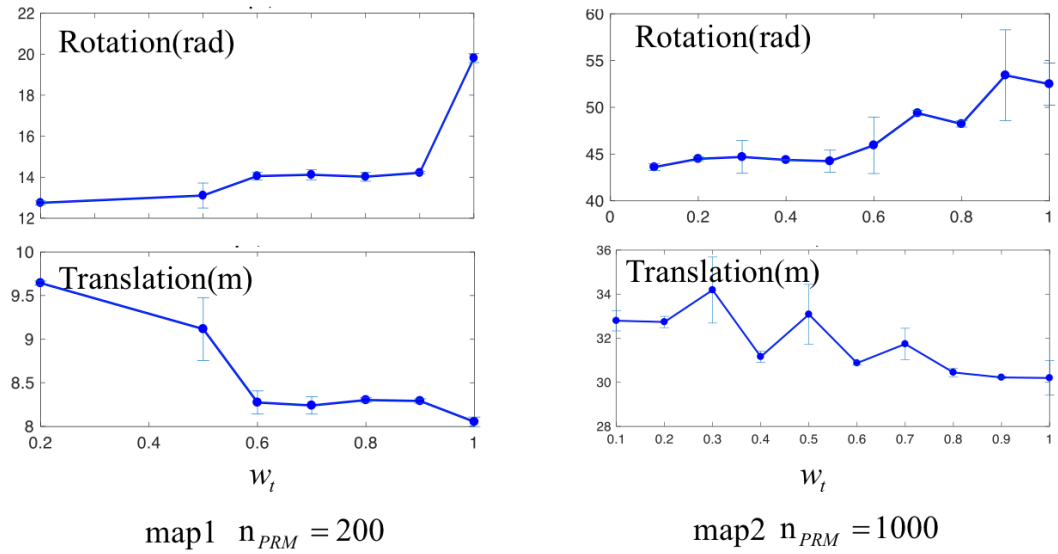


Figure 4.8: Effect of w_t in terms of rotational angle and translational distance for map 1 and map 2. The standard deviation obtained from the experiment is shown by the error bars.

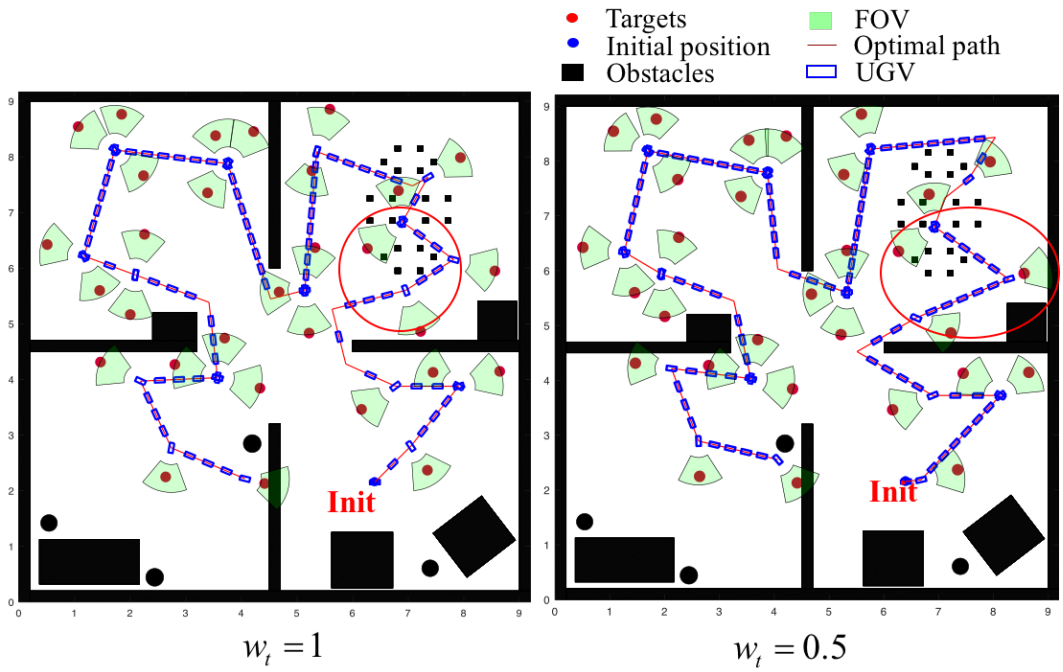


Figure 4.9: Effect of w_t demonstrated on map 2. The circles in each plot highlights the interesting effect of the weight on translation. To observe the two targets, the path on the right travels a significantly longer distance but with less rotation, since it has less weight on translation.

4.3 Simulations in MATLAB and Webots® environment

This section demonstrates the effectiveness of the proposed method through examples of MATLAB simulations of the three maps introduced. In addition, screenshot images from the Webots® simulations show the method's applicability to serve as a benchmark solution in the satisficing experiment and to adapt to complex real-world scenarios.

The simulation from map 1 (Figure 4.10) shows that the proposed method can reduce translational distance significantly by visiting a waypoint in the pruned C-target region that allows measurements from multiple targets. Although the rotational cost potentially increases, with the assumption that translation cost dominates, the C-target pruning operation is shown to be practical and efficient. The similar effects can be observed in the MATLAB simulations for map 2 (Figure 4.11 and Figure 4.14) and map 3 (Figure 4.15), and in the simulations performed in Webots® (Figure 4.13).

The example solutions to map 2 also demonstrate the influence of applying information-theoretic function EER. By identifying the targets with potential information gain before sensor deployment, the obtained path can be highly cost-efficient. For example in map 2, if without the information gain evaluation, the robot has to visit all 30 targets even when some of them cannot provide any additional information value. The result in Figure 4.14 shows the reduction in path cost brought by minimizing EER, which is the first step of the proposed method.

Overall, the proposed method has demonstrated its capability in providing a highly efficient solution with obstacle avoidance in different environment. The

solution allows the robot to avoid obstacles collisions and obtain measurements with careful consideration of potential line-of-sight occlusions, even under some complex and real-world scenarios, such as the target under the table in map 2 (Figure 4.12) and map 3 (Figure 4.15).

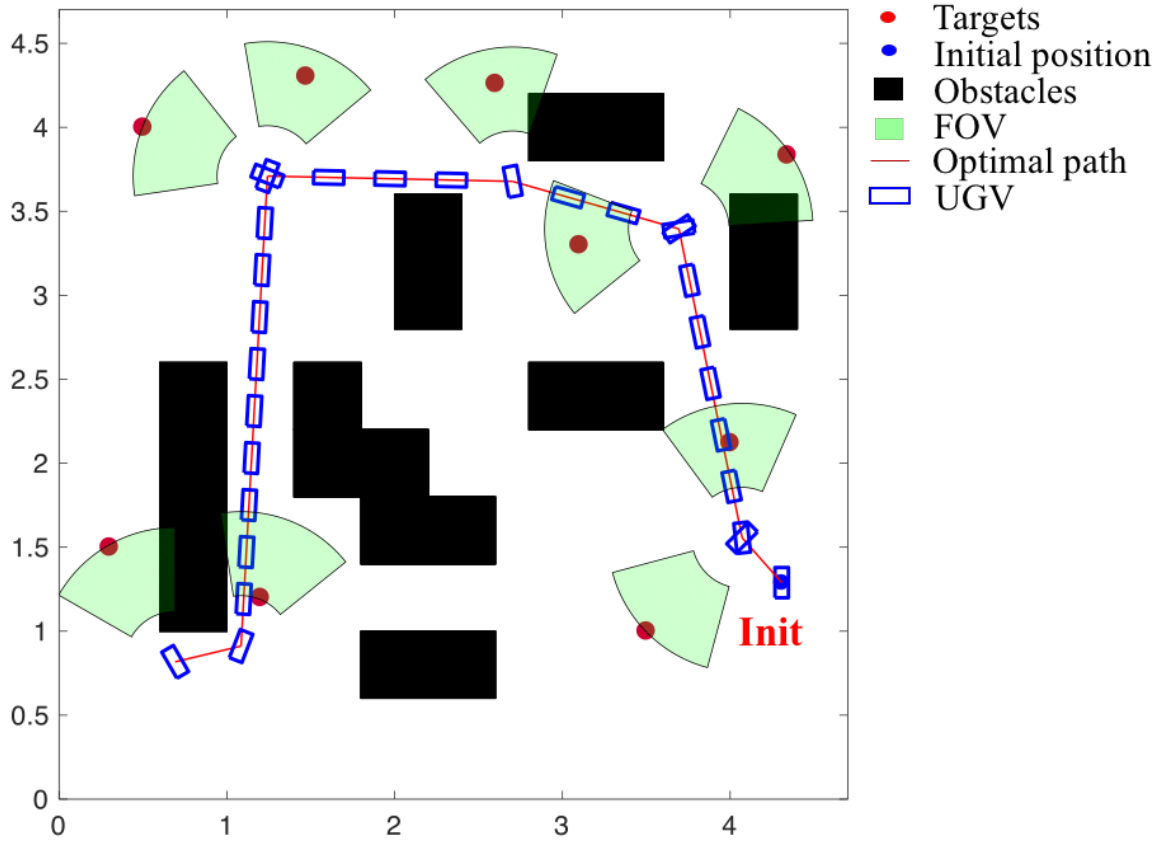


Figure 4.10: An example result for map 1 performed using MATLAB, with w_t = 1, $n_{PRM} = 200$. The robot saves translational distance significantly by rotating at certain waypoints to observe multiple targets. The total translation is 8.1 m and total rotation is 19.8 rad.

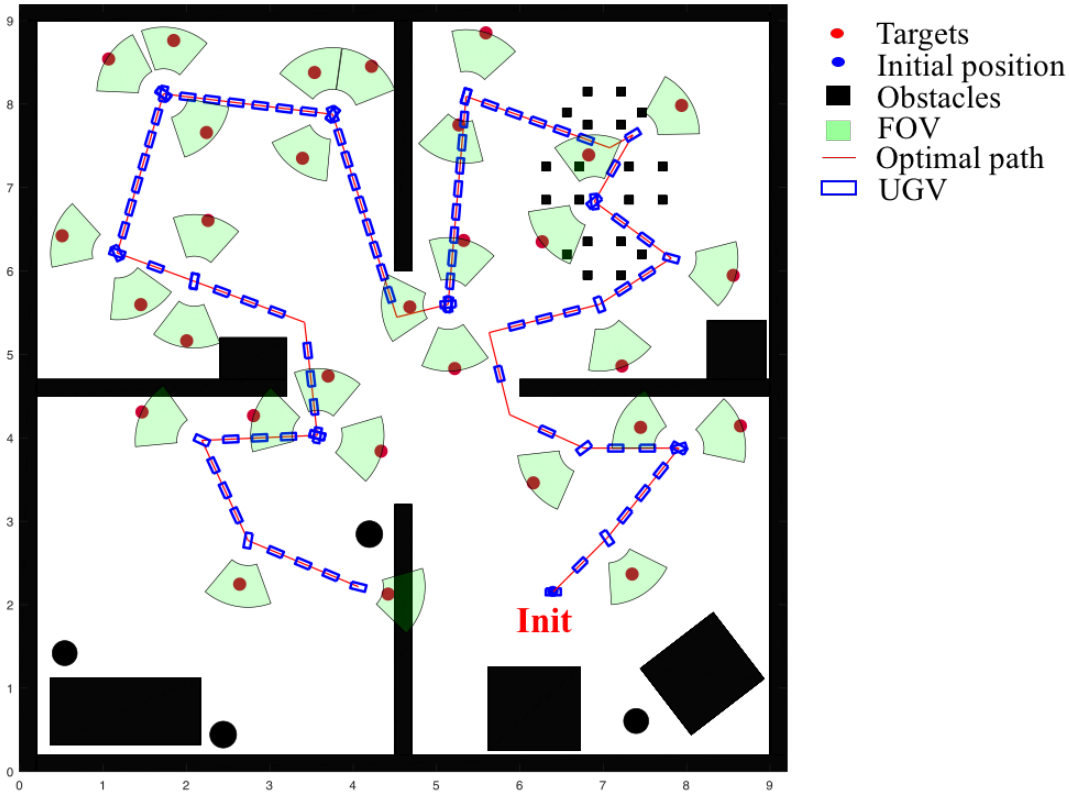
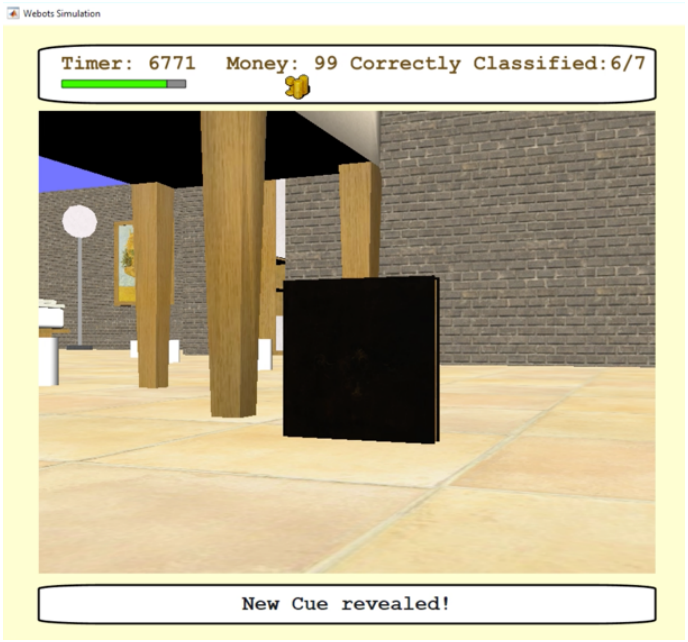


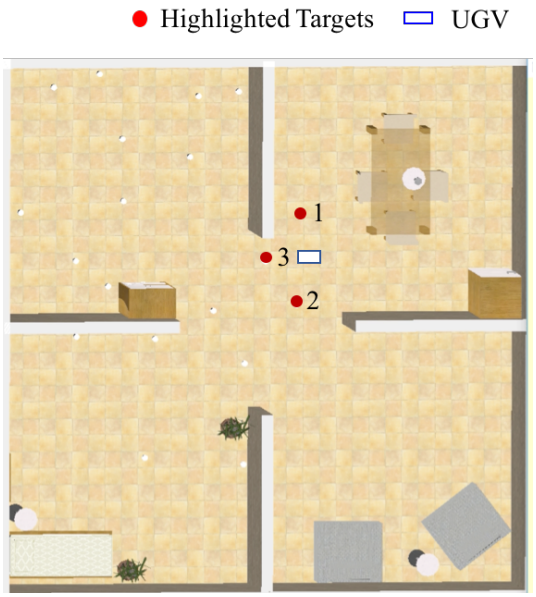
Figure 4.11: The optimal path for map 2, with 30 targets of non-zero EER, determined by C-target pruning with w_t = 1, $n_{PRM} = 1000$. The robot rotates at a waypoint in pruned C-target region to observe multiple targets if allowed. In this manner, the translation will be greatly reduced. The total translation is 30.0 m and total rotation is 45.1 rad.

(a) The mobile sensor is approaching the target under the table and chairs to take measurements.



(b) The sensor has taken feature measurements from the target under the table and can then make a classification.

Figure 4.12: Demonstrated in Webot® simulations, the mobile sensor navigates itself through a complex environment that is populated with multiple obstacles. The figure shows the process of the sensor obtaining measurements from a target under the table.



(a) At a waypoint (rectangle UGV's position), the sensor is able to observe three targets (red dots), following the labeled order.

(b) At the waypoint, the sensor rotates to make measurements on each target until it proceed to the next waypoint.

Figure 4.13: An example solution to map 2 that has 30 targets with non-zero EER, simulated in Webot®. With the help of C-target pruning operation, the mobile sensor reaches waypoints that may allow measurements on multiple targets. By adopting this approach, the translational cost is significantly reduced.

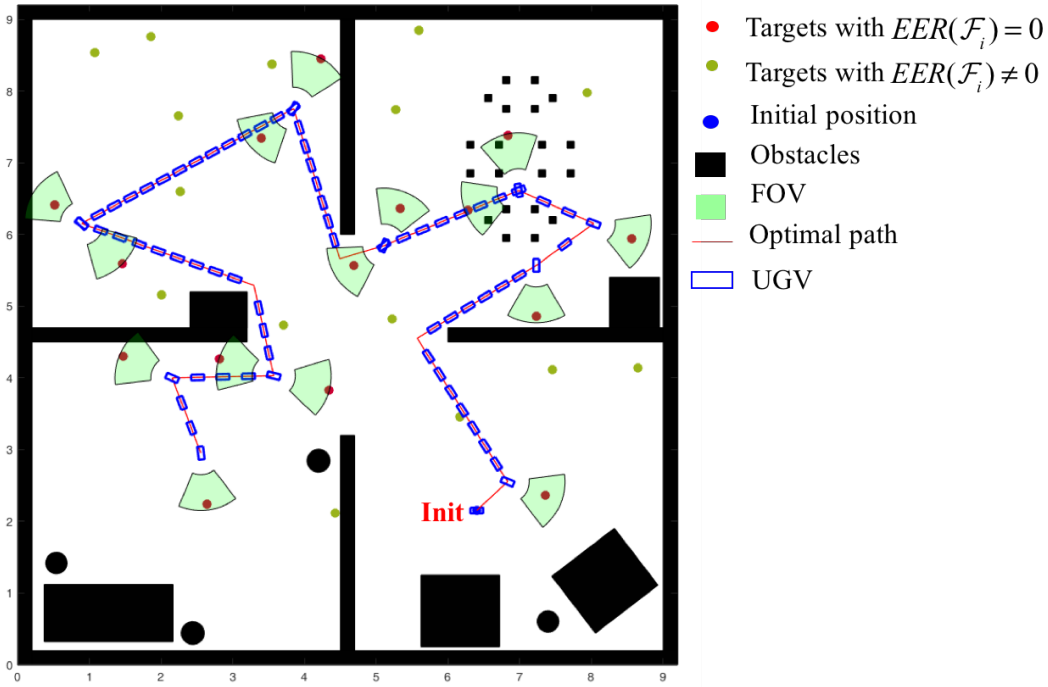


Figure 4.14: The optimal path for map 2, with 15 targets of non-zero EER, determined by C-target pruning with $w_t = 1$, $n_{PRM} = 1000$. The total translation is 21.6m. By determine a set of target with additional information gain, the translation cost is reduced by 28%.

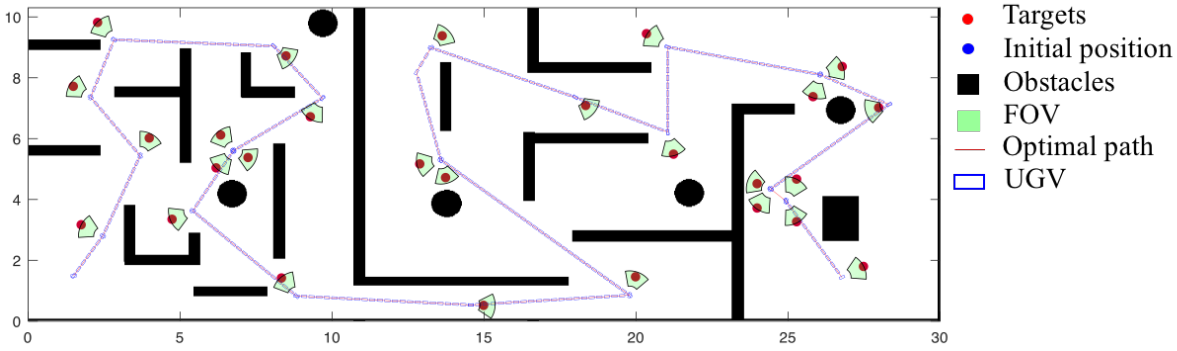


Figure 4.15: The optimal path for map 3, determined by C-target pruning with $w_t = 0.5$, $n_{PRM} = 1200$. The robot rotates at a waypoint in pruned C-target region to observe multiple targets if allowed. Benefit from this behavior, the solution's translational cost is greatly reduced. The total translation is 76.6 m.

4.4 Performance comparisons with benchmark methods

In this section, the solution obtained by the proposed method is compared with those by three benchmark methods. The first two methods are the nearest neighbor (NN) and 2-opt heuristics, subject to modification to accommodate the specific problem with obstacles. The two algorithms, especially 2-opt, are widely used approaches for traditional TSP because of their ability to provide a good quality solution in short runtime [33, 39]. In these methods, the robot is required to visit the exact location of a target, as the algorithms do not take the geometry of the sensor's FOV into consideration. NN is a straightforward tour construction approach that always decides to approach the nearest unvisited neighbor [41]. With the presence of obstacles, for each candidate target that cannot be connected with the robot's current coordinate by a straight path, a path is generated using PRM. The algorithm iterates until all targets are visited. As introduced in Section 3.4, 2-opt algorithm is a popular heuristics approach to reduce the path cost generated by NN.

The third benchmark approach is an alternative method for this directional sensor path planning problem that combines cell decomposition and integer programming methods [25]. Even though cell decomposition requires relatively high computational cost, it has the benefit of being a resolution complete [42]. Integer programming allows adding constraints iteratively and re-optimization, so that it is applicable to solve a TSP [25, 43]. The method decomposes the free configuration space into non-overlapping rectangular cells, and then constructs a directed connectivity graph from the obtained cells. Then it transforms the path planning problem into a graph problem with constraints, and obtain a solution via integer programming, a method for making discrete decisions

about edges under linear constraints. The method considers the geometry and line-of-sight visibility of a directional sensor, and provides a good-quality solution to the problem with a relatively low workspace complexity. It is also able to adapt to different weight of translation, for example, the case where translation or rotation dominates, as demonstrated in the result below. However, cell decomposition has limitations such as high computational cost, and it may not be able to handle non-polygon geometry easily [26]. Due to the computational cost and constraints in obstacle geometry, the benchmark method may not be perfectly applicable to some real-world scenarios.

The proposed method using C-target pruning is compared with the three benchmark methods introduced above, denoted as CTP in the tables below. It is tested with w_t of both 1 and 0.5 in order to provide a comprehensive understanding of its capability. The nearest neighbor method, denoted as NN, and 2-opt heuristics method are tested with three of the workspaces. The same n_{PRM} is used for proposed method, NN and 2-opt to provide consistency for the performance comparisons. The cell decomposition with integer programming approach, denoted as IP, is performed on map 2 without circular obstacles due to its limitations with obstacle geometry handling. Also, it is not tested with map 3 since the runtime required for obtaining an integer programming solution is very high for such a complex map. The performance of the methods is evaluated in terms of translational distance and rotational angle, shown in Table 4.1, Table 4.2, and Table 4.3.

Table 4.1: Performance comparison of the proposed method and three benchmark methods using map 1

	CTP w_t = 1	CTP w_t = 0.5	IP (Translation Dominates)	IP (Rotation Dominates)	NN	2-opt
Distance(m)	8.1 ± 0.1	9.1 ± 0.4	11.1	14.0	16.7	13.0
Rotation(rad)	19.8 ± 0.2	13.1 ± 0.6	14.7	11.8	14.0	12.7

Table 4.2: Performance comparison of the proposed method and three benchmark methods using map 2

	CTP w_t = 1	CTP w_t = 0.5	IP (Translation Dominates)	IP (Rotation Dominates)	NN	2-opt
Distance(m)	30.2 ± 0.7	33.1 ± 1.4	39.2	48.6	60.8	45.2
Rotation(rad)	52.5 ± 2.3	44.2 ± 1.2	58.5	38.0	45.1	42.2

Table 4.3: Performance comparison of the proposed method and three benchmark methods using map 3

	CTP w_t = 1	CTP w_t = 0.5	NN	2-opt
Distance(m)	77.6 ± 1.5	77.8 ± 0.4	90.3	87.8
Rotation(rad)	59.5 ± 1.3	47.1 ± 0.4	49.0	49.8

As demonstrated in the three tables above, for map 1, the proposed method is able to achieve 38% reduction of path distance compared to 2-opt heuristic method, and 27% reduction of distance compared to cell decomposition method. In map 2, the proposed method achieves 34% reduction of path distance compared to 2-opt heuristic method, and 23% reduction of distance compared to cell decomposition method. In map 3, the proposed method achieves 11% reduction of path distance compared to 2-opt heuristic method. In addition to its significant improvement of performance in translational cost, the proposed method also demonstrates its ability of reducing rotation by tuning

w_t. In the three maps, the proposed method has achieved a rotational cost no worse than any of the benchmark methods.

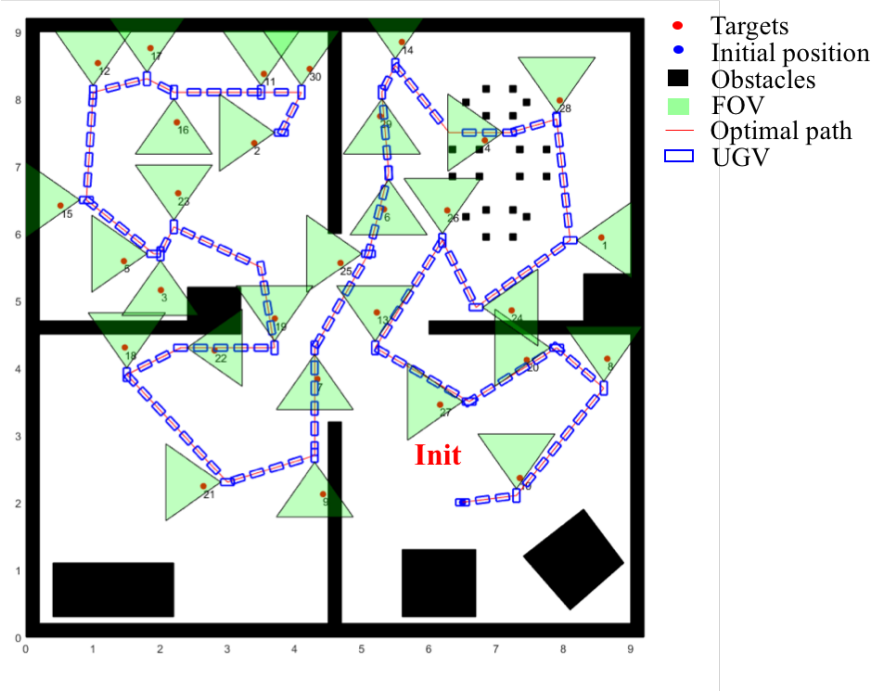


Figure 4.16: The optimal path determined by cell decomposition and integer programming (IP) for map 2, with 30 targets of non-zero EER, and the optimization assumption that translation dominates. The total translation is 39.2 m and total rotation is 58.5 rad, while with similar assumptions, the translational distance of the proposed method (CTP) is 30.2 m, and rotational angle is 52.5 rad.

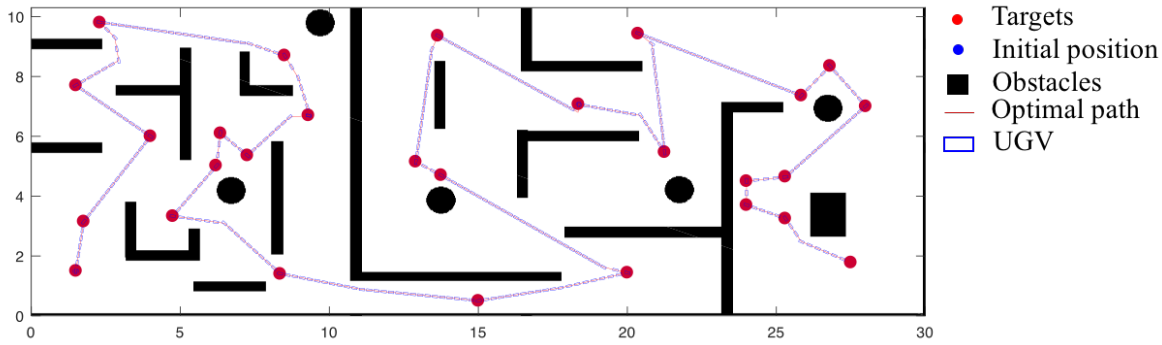


Figure 4.17: The optimal path for map 3, with $n_{PRM} = 1200$, determined by Heuristic TSP via 2-opt. Without benefiting from the sensor's FOV, the robot needs to visit at the exact location of every target. The total translation is 87.8 m. While with similar assumptions, the translational distance of the proposed method (CTP) is 77.6 m.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis provides a systematic methodology based on target configuration operation and probabilistic method for path planning for a mobile directional sensor to classify targets in the obstacle-populated workspace within minimum time. In the proposed method, the sensing objective is fulfilled by estimating the target's potential information value. The target configuration is determined and pruned while maintaining the requirement of obtaining target measurements. Then, a TSP representation of the problem is achieved using a connectivity graph that adapts to the problem's special properties, such as the presence of obstacles. The methodology is also able to incorporate rotation cost in waypoint optimization. Simulations on several workspaces of different complexity have demonstrated the effectiveness and performance improvement of the proposed approach when compared to benchmark methods, including traditional TSP algorithms like "nearest neighbor" and 2-opt heuristics, and a directional sensor planning method using cell decomposition and integer programming.

The proposed method may be further improved with future works. For example, the method may be able to incorporate rotational cost earlier, so that the method is further applicable to such problem where rotation cost is as important as translation cost. Some of works like [44] has investigated the time-optimal trajectory problem in the unobstructed plane using optimal control, providing inspirations that may be applied in the sensor path planning as well. The method can be also developed to incorporate with different sensing objective, for example, an objective of gaining information value subject to time pressure, or gaining the most information with a limited time constraint. With

such adaptability, the solution can provide more comprehensive comparisons with human decision strategies in the satisficing experiments.

BIBLIOGRAPHY

[1] Guoxian Zhang, Silvia Ferrari, and M Qian. An information roadmap method for robotic sensor path planning. *Journal of Intelligent and Robotic Systems*, 56(1-2):69–98, 2009.

[2] Ercan U Acar, Howie Choset, Yangang Zhang, and Mark Schervish. Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods. *The International Journal of Robotics Research*, pages 441–466, 07 2003.

[3] N. S. V. Rao. Robot navigation in unknown generalized polygonal terrains using vision sensors. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6):947–962, 1995.

[4] Anthony Lazanas and Jean-Claude Latombe. Motion planning with uncertainty: a landmark approach. *Artificial Intelligence*, 76(1-2):287–317, 1995.

[5] Kai-Tai Song and C. C. Chang. Reactive navigation in dynamic environment using a multisensor predictor. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(6):870–880, 1999.

[6] Rachael Bis, Huei Peng, and Galip Ulsoy. Velocity occupancy space: Robot navigation and moving obstacle avoidance with sensor uncertainty. *ASME 2009 Dynamic Systems and Control Conference, Volume 1*, 2009.

[7] Chenghui Cai. *Information-driven sensor path planning and the treasure hunt problem*. PhD thesis, Duke University, 2008.

[8] R. Siegel. Land mine detection. *IEEE Instrumentation Measurement Magazine*, 5(4):22–28, Dec 2002.

[9] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. *ACM SIGOPS Operating Systems Review*, 36(5):96–107, 2002.

[10] David Culler, Deborah Estrin, and Mani Srivastava. Overview of sensor networks. *Computer*, 37(8):41–49, 2004.

[11] Hanna Oh, Jeffrey M Beck, Pingping Zhu, Marc Sommer, Silvia Ferrari, and Tobias Egner. Satisficing in split-second decision making is characterized by strategic cue discounting. 42, 06 2016.

[12] Hanna Oh-Descher, Jeffrey M. Beck, Silvia Ferrari, Marc A. Sommer, and Tobias Egner. Probabilistic inference under time pressure leads to a cortical-to-subcortical shift in decision evidence integration. *NeuroImage*, 162, 2017.

[13] Herbert A. Simon. Invariants of human behavior. *Annual Review of Psychology*, 41(1):1–19, 1990.

[14] Greg Hager and Max Mintz. Computational methods for task-directed sensor data fusion and sensor planning. *The International Journal of Robotics Research*, 10(4):285–313, 1991.

[15] Howie Choset. Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, Oct 2001.

[16] Enric Galceran and Marc Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258 – 1276, 2013.

[17] Wenjie Lu, Guoxian Zhang, and Silvia Ferrari. An information potential approach to integrated sensor path planning and control. *IEEE Transactions on Robotics*, 30(4):919–934, 2014.

[18] D. G. Macharet, A. A. Neto, V. F. da Camara Neto, and M. F. M. Campos. Data gathering tour optimization for dubins' vehicles. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, June 2012.

[19] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12), 2008.

[20] Bo Yuan, M. Orlowska, and S. Sadiq. On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(9):1252–1261, 2007.

[21] Esther M. Arkin and Refael Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55(3):197–218, 1994.

[22] Khaled Elbassioni, Aleksei V. Fishkin, Nabil H. Mustafa, and RenÃ©

Sitters. Approximation algorithms for euclidean group tsp. *Automata, Languages and Programming Lecture Notes in Computer Science*, pages 1115–1126, 2005.

[23] Melanie Mitchell. *An introduction to genetic algorithms*. MIT Press, 1996.

[24] Jing-Sin Liu, Shao-You Wu, and Ko-Ming Chiu. Path planning of a data mule in wireless sensor network using an improved implementation of clustering-based genetic algorithm. *2013 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, 2013.

[25] Zeyu Liu. A cell decomposition approach to autonomous path planning for directional mobile sensors. Master's thesis, Cornell University, Ithaca, 2018.

[26] Jean-Claude Latombe. *Robot motion planning*. Kluwer, 2010.

[27] Qingji Gao, Zunchao Zheng, and Guochen Niu. Path planning based on dimension reduction and region clipping for aircraft fuel tank inspection robot. *Journal of Software*, 9(6):1626, 2014.

[28] Zhe Lv, Liying Yang, Yuqing He, Zhong Liu, and Zhonghua Han. 3d environment modeling with height dimension reduction and path planning for uav. pages 734–739. IEEE, 2017.

[29] C. Cai and S. Ferrari. Comparison of information-theoretic objective functions for decision support in sensor systems. In *2007 American Control Conference*, pages 3559–3564, July 2007.

[30] M. P. Kolba and L. M. Collins. Information-based sensor management in the presence of uncertainty. *IEEE Transactions on Signal Processing*, 55(6):2731–2735, 2007.

[31] Feng Zhao, Jaewon Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Magazine*, 19(2):61–72, 2002.

[32] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley, 1991.

[33] Wang Hui. Comparison of several intelligent algorithms for solving tsp problem in industrial engineering. *Systems Engineering Procedia*, 4:226 – 235, 2012. Information Engineering and Complexity Science - Part II.

[34] Haider Abdulkarim and Ibrahim F Alshammari. Comparison of algorithms for solving traveling salesman problem. ISSN:2249 – 8958, 08 2015.

[35] Tsai Chun-Wei, Shih-Pang Tseng, Chiang Ming-Chao, Yang Chu-Sing, and Tzung-Pei Hong. A high-performance genetic algorithm: Using traveling salesman problem as a case. *The Scientific World Journal*, 2014.

[36] Yuan-bin Mo. *The Advantage of Intelligent Algorithms for TSP*. INTECH Open Access Publisher, 2010.

[37] T. Ramadhani, G. F. Hertono, and B. D. Handari. An ant colony optimization algorithm for solving the fixed destination multi-depot multiple traveling salesman problem with non-random parameters. *AIP Conference Proceedings*, 1862(1), 2017.

[38] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug 1996.

[39] Punnen, Margot, and Kabadi. Tsp heuristics: Domination analysis and complexity. *Algorithmica*, 35(2):111–127, 2003;2002;.

[40] Keld Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106 – 130, 2000.

[41] David S. Johnson and Lyle A. Mcgeoch. The traveling salesman problem:. *Local Search in Combinatorial Optimization*, pages 215–310, 1997.

[42] Chenghui Cai and S. Ferrari. Information-driven sensor path planning by approximate cell decomposition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3):672–689, 2009.

[43] P. Miliotis. Integer programming approaches to the travelling salesman problem. *Mathematical Programming*, 10(1):367–378, Dec 1976.

[44] Devin J. Balkcom and Matthew T. Mason. Time optimal trajectories for bounded velocity differential drive vehicles. *The International Journal of Robotics Research*, 21(3):199–217, 2002.