# INFORMATION-DRIVEN MULTI-VIEW PATH PLANNING FOR UNDERWATER TARGET RECOGNITION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Jane Shin

August 2021

INFORMATION-DRIVEN MULTI-VIEW PATH PLANNING FOR UNDERWATER TARGET RECOGNITION

Jane Shin, Ph.D.

Cornell University 2021

By utilizing onboard sensors such as side-scan or forward-looking sonar, autonomous underwater robots can perform many useful tasks, such as exploring and searching for targets in underwater environments. In order to recognize and classify objects with high confidence, however, these mobile sensors must obtain multiple looks or "views" for each target using different positions and orientations that allow for a different interpretation based on local occlusions and environmental conditions. As a result, when tasked with classifying many targets, the mobile sensor must find the most efficient path through multiple configurations in an effort to reduce the cost and time required by each underwater mission. This dissertation presents a novel and general approach, referred to as informative multi-view planning (IMVP), that simultaneously determines the most informative sequence of views and the shortest path between them. The approach is demonstrated both in simulations and sea tests using an unmanned underwater vehicle (UUV) equipped with a side-scan sonar (SSS) and engaged in underwater multi-target classification. Both simulation and experimental results show that IMVP achieves excellent classification performance while reducing the total time required by the mission by up to half the time required by state-of-the-art multi-view path planning methods. One reason is that IMVP utilizes knowledge of the automatic target recognition (ATR) algorithm, as well as prior measurements, in order to determine the most informative views. Additionally, by using knowledge of the target location

and field-of-view (FOV) geometry, IMVP is able to find the shortest path between them by solving a traveling salesman problem with neighborhoods (TSPN). In this dissertation, a novel physics-inspired algorithm based on Lin-Kernighan heuristic (LKH) is developed for searching for the optimal TSPN path for multiple non-disjoint neighborhoods. It is shown that the LKH algorithm is able to decrease the computational complexity of TSPN solutions by leveraging the intersections of valuable neighborhoods using computational geometry constructs known as coverage cones. When compared to state-of-the-art TSPN algorithms, the proposed method is able to find shorter paths with either comparable or reduced computation. The advantages of the LKH algorithm are found to become more significant as the number of intersecting neighborhoods increases, thus also allowing the mobile sensor to observe multiple targets from a single configuration.

## BIOGRAPHICAL SKETCH

Jane started the Mechanical Engineering PhD program at Cornell in the Fall of 2017 after graduating with her B.Sc. degree in Naval Architecture and Ocean Engineering in 2017 from Seoul National University, Korea. Her main research interests are in the development of planning methodologies for autonomous robotic systems equipped with imaging sensors. Her research interests encompass information theoretic learning, computational geometry, motion planning, machine learning, and optimization. At Cornell, Jane was named a 2020 Commercialization fellow by the college of engineering, in partnership with the Johnson Business school, and he was awarded the National Science Foundation Innovation-Corps grant to pursue commercial applications of her research.

To my family,

## ACKNOWLEDGEMENTS

I would like to thank my labmates in the Laboratory for Intelligent Systems and Controls (LISC), as well as my friends and colleagues, for their help and support along my PhD. I could always learn something by having a conversation with them in the lab. I would also like to thank my advisor and mentor, Silvia Ferrari, who is also the director of LISC. Silvia has been a great mentor who motivates me to pursue my academic dream. Because of her help and support, I could participate in exciting research projects and have a chance to learn research commercialization process through Cornell Commercialization Fellowship. I would also like to acknowledge my I-Corps team members, Carlos and Nate, as well as the teaching team for the Commercialization fellowship at Cornell. This program really broadened my horizon on how to interpret my research and how to find the value in it. Furthermore, I would like to thank the collaborators at Naval Surface Warfare Center (NSWC), Drs. Bays, Wettergren, and Weaver, for the opportunity for being a part of the exciting collaboration. Finally, I would like to thank my family for their love and support.

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION AND MOTIVATION

Many modern imaging sensors, such as underwater sonar or cameras, require multiple looks or "views" of the target before they are able to classify it with a high level of confidence. By changing the sensor position and orientation relative to each target, different information about target features, such as shape and size, may be obtained and fused in order to better infer the target class. Mobile platforms, such as uncrewed vehicles, are often utilized to allow the sensor to travel around an object and record multiple images from different viewpoints. When a sensor must classify multiple targets distributed over a large region, obtaining multiple views may require traveling over a long distance in order to visit multiple positions and orientations relative to each target, resulting in costly and time-consuming operations that may potentially outlive the battery life of the vehicle.

Multiple aspect coverage (MAC) and adaptive MAC (AMAC) algorithms have been developed to solve multi-view path planning problems by first generating a star-like path around every target and, then, computing the shortest route between them [7, 70]. MAC-type algorithms rely on the user choosing the number of views required for every target, and, subsequently, picking a subset of vehicle heading angles by sampling uniformly the 180-degree range of all possible angles. Another solution approach proposed in [20] connects multiple viewing angles decided *a priori* by means of Dubins curves that are reachable based on the vehicle kinematic constraints. The sensor's next viewing angle is chosen based on experimental results and, then, the path is planned such that every target is visited again with the same viewing angle. All of these existing algorithms seek to reduce the task-completion time by finding the shortest path between multiple views decided *a* *priori*. Because they do not take into account individual target characteristics, they may obtain too-many or too-few images resulting in paths that are too time consuming or have low classification accuracy, respectively. Moreover, because they rely on user intervention, they may be difficult to automate and adapt to evolving classification and environmental conditions.

Along a different line of research, next-best-view approaches have been developed to determine what is the next most informative view for a target, based on information gain [40, 63], or partially observable Markov decision processes (POMDPs) [60]. Information-driven path-planning approaches, reviewed comprehensively in [32], take into account both the sensing objective and the vehicle kinodynamic constraints in order to simultaneously optimize the sensor performance and the energy consumption. Finding the most efficient sensor path is especially critical in many underwater applications because, due to limited communications and rapidly changing sea conditions, the vehicles must travel back to the host ship or surface up to update their information state and complete each operation as quickly as possible. Although existing information-driven path-planning methods have been shown highly effective at optimizing the performance of mobile sensors [5, 14, 32, 34, 78, 83], these existing methods are not directly applicable to multi-view planning because they assume the information gain is independent of target-relative position and orientation. Many imaging sensors, such as cameras, active sonar, and radar, interpret a return signal (e.g. acoustic or optical wave) that bounces off an object of interest and, thus, the quality of their measurements heavily depends on their aspect angle. Furthermore, the image quality and the ability to recognize the object also depend on the object's shadow and self-occlusions, which vary with both sensor position and orientation, as well as the location of the illumination source.

This dissertation develops a novel and general approach for informative multi-view planning (IMVP) that simultaneously determines the best sensor views for each individual target, based on prior information, and plans the optimal path between them. The IMVP approach developed in this dissertation takes into account the sensor's field-of-view (FOV) and Bayesian measurement model, as well as the target's position and orientation, and constructs novel C-target regions and information gain functions applicable to imaging sensors. It is shown that, for a continuous and bounded sensor FOV, the optimal path can be found by solving a generalized traveling salesman problem (TSP) [3, 54]. Due to its high complexity, many approximate and ad-hoc solutions have been proposed for generalized TSPs [10, 24, 25, 57]. One of the most common simplifications is to assume that the regions of interest are pairwise disjoint [16, 26, 56]. However, in informative multi-view path planning, intersecting regions are often the most valuable because they allow the sensor to obtain images from multiple targets in a single pass.

The IMVP approach builds on several novel contributions that allow to first formulate and, then, solve the generalized TSP based on all available sensor and target information (Chapter 3). Novel contributions in computational geometry (Chapter 4) allow for the efficient construction of helicoidal C-targets and transcription of the path planning problem into a new generalized traveling salesman problem with intersecting neighborhoods (Sections 4.1-4.2). A new GTSPN solution algorithm is also developed (Section 4.3) that exploits all neighborhood intersections corresponding to sensor configurations able to view multiple targets at once (in the same image), while reducing computational complexity (Section 4.4). This dissertation also investigates the influence of self-localization errors and uncertainty in the target location by modifying the GTSPN solution to increase performance robustness (Section 7). The novel IMVP approach is demonstrated both on a simulated and a real underwater vehicle equipped with a simulated side-scan imaging sonar and tasked with classifying multiple underwater objects previously detected during surveying.

The new GTSPN solution algorithm developed in Chapter 4 is shown to compute time-efficient paths that outperform existing state-of-the-art methods for underwater multi-target recognition by explicitly considering the intersection of multiple neighborhoods that each correspond to a different C-target. Because computing neighborhood intersections is an expensive operation, the GTSPN solution may require implementing further approximations and heuristics in order to be applicable to problems in which the number of targets and looks required are both large. This dissertation develops a novel physics-inspired TSPN solution algorithm that is based on computational geometry constructs known as coverage cones (Chapter 5). Inspired by Fermat's principle, the proposed TSPN solution computes a light ray-like path using a coverage cone that contains all line transversals to each neighborhood. Assuming the neighborhoods can be approximated by circles, the ray-like path allows to determine an ideal sequence of circles and waypoint positions that form a TSPN tour. Unlike existing methods such as unsupervised learning [28] or evolutionary algorithms [82], the proposed research does not require any prior data. Also, while the solution approach is developed for uniform circles, the concept can be potentially extended to other neighborhood geometries, such as ovals or convex polygons.

# CHAPTER 2

## PROBLEM FORMULATION

This dissertation considers the problem of planning a time-optimal path for an underwater imaging sensor deployed onboard a UUV tasked with underwater target recognition. This problem is relevant to many other robotics applications involving mobile directional sensors tasked with observing multiple complex targets distributed over a large region of interest (ROI). Because acoustic measurements are greatly influenced by the relative sensor position and orientation, most underwater targets require many views before they can be accurately analysed and classified, resulting in time-consuming and costly operations. Prior to deploying an expensive short-range directional sonar, such as a forward-looking or side-scan sonar, the ROI is typically surveyed using long-range sensors that detect targets of interest and provide a rudimentary estimate of their position and orientation. Subsequently, the UUV-mounted directional sensor is used to obtain multiple images or "looks" for every object or target of interest in order to achieve classifications accompanied by high confidence levels [6, 7]. As a result, the time and power required to properly classify multiple targets is highly dependent on the UUV path. Furthermore, the optimal number of views and corresponding aspect angles, as well as the classification confidence levels, all depend on the ATR algorithm and target characteristics that may also be estimated during surveying. By the approach developed in this dissertation, the number of sensor views, aspect angles, and UUV path are simultaneously optimized based on prior measurements, target characteristics, and ATR properties.

While applicable to other problem formulations, the directional sensor planning approach presented in this dissertation is demonstrated on a minimum-time benchmark problem in which multiple underwater targets are to be classified within a desired confidence level. For example, the methods developed here are also applicable to online adaptive path planning and to optimal planning for other directional sensors, such as cameras or synthetic aperture radar. The benchmark minimum-time problem consists of planning the motion of a UUV equipped with a side-scan sonar in a three-dimensional ROI, $\mathcal{W} \subset \mathbb{R}^3$. For simplicity, it is assumed all $n$ underwater targets detected and localized during surveying are distributed on a flat seabed, such that the effects of sloped and uneven bathymetry, e.g. layover, on the sonar imagery are negligible. Let an inertial frame $\mathcal{F}_{\mathcal{W}}$ with origin $\mathcal{O}_{\mathcal{W}}$ be embedded in $\mathcal{W}$ such that the $x_I y_I$-plane contains the seabed of interest denoted by a plane $P \subset \mathbb{R}^3$ (Fig. 2.1).

Every target in $P$ is characterized by unknown geometries $\mathcal{T}_1, ..., \mathcal{T}_n$, where $\mathcal{T}_i \subset \mathcal{W}$ is a compact set, and $i = 1, ..., n$, and its inertial position, estimated during pre-surveying, is denoted by $\mathbf{x}_{T_i} = [x_{T_i}\ y_{T_i}]^T$. Letting $\mathcal{F}_{\mathcal{T}_i}$ denote a local reference frame embedded in $\mathcal{T}_i$, the target *orientation*, $\theta_{T_i}$, can be defined as the rotation angle from the $x_{\mathcal{T}_i}$-axis to the $x_I$-axis, about $z_I$-axis (Fig. 2.2). Due to the nature of acoustic measurements, the image constructed by the sensor is highly dependent on the so-called *aspect angle*, defined as the off-normal angle between the target and the sensor orientations, measured relative to the sonar centerline and denoted by $\varphi$ (Fig. 2.2). Therefore, the target state or *configuration*, $\mathbf{q}_{T_i} = [\mathbf{x}_{T_i}^T\ \theta_{T_i}]^T$, must be estimated for $i = 1, ..., n$, in order to optimize the quality of the sonar imagery for all $n$ targets of interest.

Using a chosen ATR algorithm, the UUV-based sonar must classify all $n$ targets with a desired confidence level, based on $f$ target features, such as target geometry, size, and texture. Every target feature can be represented by a categorical random

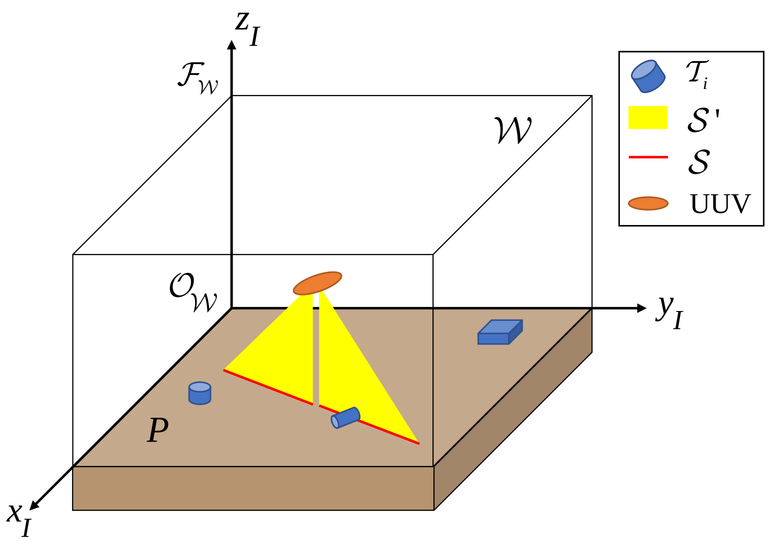



Figure 2.1: Region of interest (ROI) and key geometrical constructs

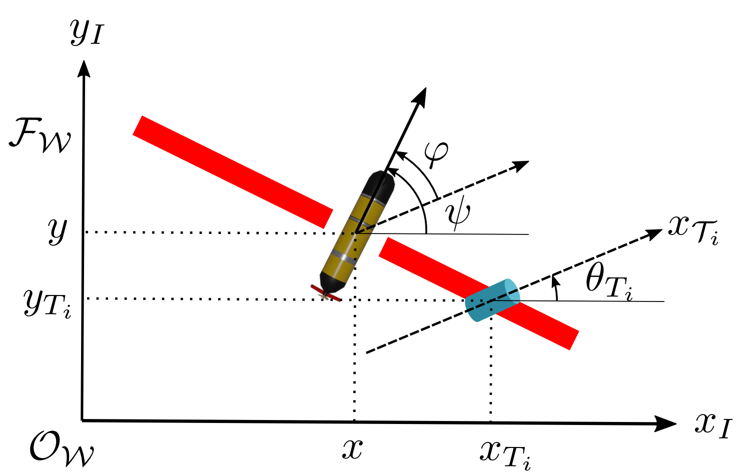



Figure 2.2: Definition of target orientation and aspect angle for a UUV-based sonar (ROI top view, UUV projected on the seabed)

variable $X_{ij} \in \mathcal{X}_j$, where $\mathcal{X}_j$ is the discrete and finite range of the $j^{th}$ feature ($j = 1, ..., f$). Then, the feature set of the $i$th target is denoted by $X_i = \{X_{i1}, ..., X_{if}\}$. The target classification is denoted by another categorical variable, $Y_i \in \mathcal{Y}$, with discrete and finite range $\mathcal{Y} = \{y_1, ..., y_c\}$. Both target features and classification variables are viewed as discrete random variables. But while the features may be estimated from the target sonar image, the classification variable is hidden and must be inferred from the target features (Chapter 3). Therefore, the ATR algorithm must carry out feature estimation and target classification for every new sonar image obtained by the UUV. By optimizing the sonar viewing angle and by taking into account the geometry of the field-of-view, the approach presented in this dissertation not only minimizes classification time but also minimizes the number of images required, thus minimizing computing and power requirements.

Although the approach is demonstrated for a side-scan sonar accompanied by a convolutional neural network ATR algorithm, it can be easily extended to other underwater imaging sensors systems such as sector-scan sonar and synthetic aperture sonar and feature-extraction ATR [38]. Consider the case in which a pair of side-scan sonar sensors is installed on each port and starboard side of the UUV. The sensor field-of-view (FOV) is defined as the region in which target measurements can be obtained [15, 32]. Each sensor transmits a narrow fan-shape acoustic pulse, whose geometry is denoted by $\mathcal{S}' \subset \mathcal{W}$. As the UUV moves forward, $\mathcal{S}'$ sweeps the seabed, and a sonar image matrix is constructed by stacking the interpreted data from successive scan lines on the seabed (Fig. 2.3). Because all the targets live in $P$, the sonar FOV can be reduced to the two-dimensional region $\mathcal{S} = \mathcal{S}' \cap P$, as shown in Fig. 2.3. By considering the FOV position and geometry relative to the UUV and the targets, the sensor path and viewpoints are optimized subject to prior measurements and chosen ATR.

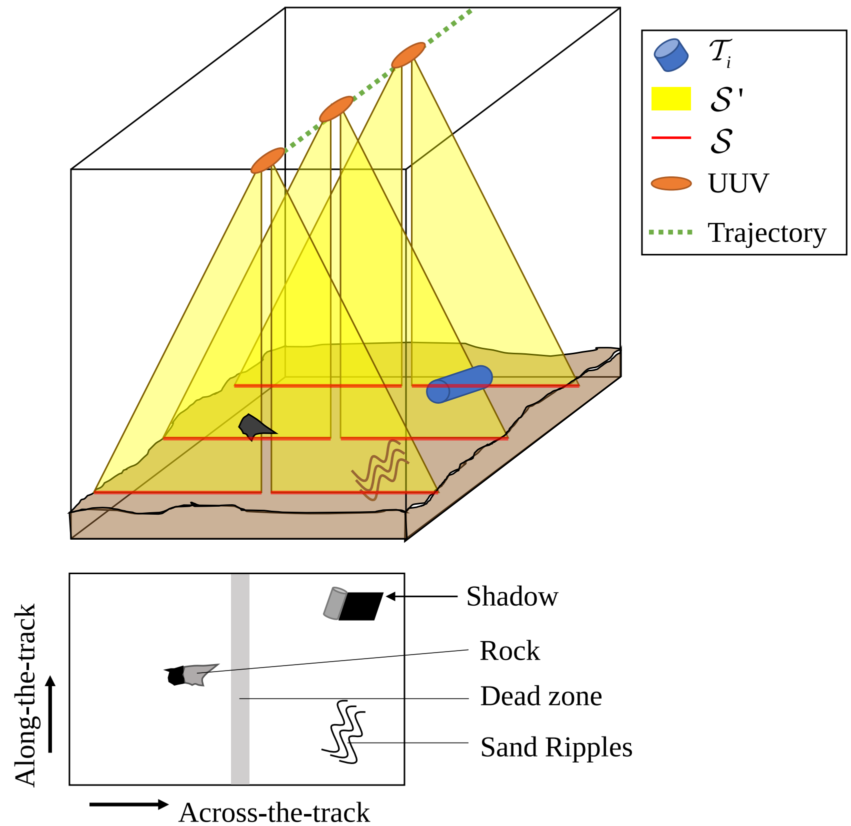



Figure 2.3: Side-scan sonar image construction and sensor FOV.

As shown in [15, 17, 32, 34, 53, 83–85], the sonar ATR and measurement process can be modeled by a probabilistic sensor model in the form of a joint probability mass function (PMF) learned from data (Chapter 3). Without loss of generality, let the set $Z_i(k)$ denote the sensor measurements obtained from target $i$ at time instant $t_k$. The sensor mode and relevant environmental conditions at $t_k$ are denoted by $\Lambda_i(k)$. In this dissertation, $\Lambda_i(k)$ consists of the sonar viewing angle and relative target position, both of which can be estimated by an onboard localization algorithm. Then, with the evidence at $t_k$ denoted by $E_i(k) = \{Z_i(k), \Lambda_i(k)\}$, the learned PMF model can be factorized as follows,

$$p(Z_i, X_i, Y_i, \Lambda_i) = p(Z_i|\Lambda_i, X_i)p(X_i|Y_i)p(Y_i)p(\Lambda_i) \tag{2.1}$$

where the PMF notation $f_Y(y) = P(\{Y = y\})$ is abbreviated by $p(Y)$ (Chapter

3).

From the evidence $E_i(k)$, the target classification can be obtained by one of several approaches, including the maximum *a-posteriori* (MAP) rule, the maximum likelihood estimate (MLE), or the Neyman-Pearson rule [32]. In order to allow for Bayesian updates, the approach presented in this dissertation adopts MAP classification, and the posterior PMF is used to estimate the classification confidence level. In particular, given the set of all evidence obtained up to time step $t_k$, denoted by $M_i(k) = \{E_i(1), ..., E_i(k)\}$, the confidence level (CL) of the $i^{th}$-target's classification at time $t_k$ is given by

$$c(Y_i; M_i(k)) \triangleq \max_{y \in \mathcal{Y}} [P(\{Y_i = y\} | M_i(k))] \tag{2.2}$$

Then, the desired classification performance can be specified via a CL threshold, $\varepsilon_{CL} \in (0, 1)$, chosen by the user based on the application and the acceptable rate of false alarms. Let $T$ denote the total UUV travel time, corresponding to the final discrete time step index $K$. Then, the goal of the path planning algorithm is to achieve a satisfactory CL for all targets in the ROI, or

$$c(Y_i; M_i(K)) \geq \varepsilon_{CL}, \quad \forall i, \quad i = 1, \ldots, n \tag{2.3}$$

A higher CL threshold results in lower classification uncertainty and in a larger travel time required to obtain more images of the targets.

In addition to meeting sensing requirements, the path planning approach must also take into account UUV motion constraints. For example, in order to minimize the geometric distortions of sonar images, the UUV must be held at a constant speed, altitude, and heading angle with zero roll and pitch angles during every time interval while sonar data is being recorded [11, 18]. The UUV kinodynamic constraints may be similarly accounted for, e.g. in order to minimize energy consumption subject to ocean current velocity fields as shown in [32]. For simplicity, the sonar is assumed to operate at a constant frequency so that the UUV altitude is maintained at a constant value, $h$, chosen based on the sensor mode. Then, the UUV configuration can be represented by $\mathbf{q} = [\mathbf{x}\ \psi]^T$, where $\mathbf{x} = [x\ y]^T$, $x$ and $y$ denote the position of the UUV in $x_I$ and $y_I$, respectively, and $\psi$ denotes UUV's heading angle. Then, the target aspect angle is given by $\varphi = \psi - \theta_{T_i}$ (Fig. 2.2), and the space of all possible UUV configurations is denoted by $\mathcal{C}$. The UUV path is defined as a continuous mapping, denoted by $\tau : [0,\ 1] \rightarrow \mathcal{C}$ with $\tau(0) = \mathbf{q}_{(0)}$ and $\tau(1) = \mathbf{q}_f$, where $\mathbf{q}_{(0)}$ is the given initial configuration and $\mathbf{q}_f$ is a final configuration to be determined. Finally, the minimum-time benchmark problem can be summarized as follows:

**Problem 1 (Sensor path planning)** *Given $n$ target positions and orientations, $\mathbf{q}_{T_1}, \ldots, \mathbf{q}_{T_n}$, the sensor ATR model in* (2.1)*, the sensor FOV $\mathcal{S}$, and the sensor initial configuration $\mathbf{q}_{(0)}$, find a path $\tau$ that minimizes the travel time $T$ such that the CL constraints* (2.3) *are met for all $n$ targets.*

CHAPTER 3

# BACKGROUND ON AUTOMATIC TARGET RECOGNITION (ATR) AND IMAGING SONAR

Imaging sonar is a powerful tool that is utilized in a variety of underwater tasks ranging from commercial applications, such as ship hull inspection, to environmental research, such as bathymetric mapping, biomass estimation, and demining [12, 65, 66, 68, 76]. Therefore, many sonar automatic target detection (ATR) and classification algorithms have been developed to date in the literature using convolutional neural networks (CNNs), feature extraction, and other image processing methods described in [32]. From a labeled measurement database and the chosen ATR algorithm, it is possible to learn correlations between characteristic highlight-shadow patterns and physical object features such as shape, size, and orientation in the form of a graphical model [42, 43, 47, 59, 80]. Because sonar images are highly dependent on environmental conditions and sensor-target aspect angle [74], high-quality classification requires fusing multiple images obtained by different viewpoints [30, 79, 81].

In this dissertation, the UUV-mounted side-scan sonar system is simulated using a high-fidelity physics-based closed-loop software developed by Dr. Isaacs at Naval Surface Warfare Center (NSWC). The mobile side-scan sonar is simulated by generating images obtained from the sonar FOV, integrated with a dynamic model of a REMUS 100 vehicle, and by $\omega - k$ beamforming of the time-domain signals [42–45] (Section 6). The results in Section 6.1 show that the IMVP approach significantly outperforms existing methods by achieving the desired classification performance in some cases in half the travel time. In addition to improving classification efficiency and confidence gain by up to 88% and 91%, respectively, IMVP also provides much higher performance robustness than existing algorithms for different classification databases, target layouts, and environmental conditions. By determining the number of views and aspect angles based on their information value and, simultaneously, considering the problem geometry, the sensor paths obtained by IMVP are not only shorter but also produce sonar images that contain on average many more contacts and provide better quality automatic target recognition. Autonomous Vehicle Architecture (AVA) simulations (Section 6.2) and sea tests were conducted by Dr. Weaver on the NSWC unmanned underwater vehicle (UUV) swimming in the Saint Andrew Bay area near Panama City, FL (Sections 6.1-6.3). The sea tests showed that the real REMUS 100 not only was able to execute the IMVP optimal path, but also outperformed the AVA simulation results under all performance metrics.

For the purpose of coupling the ATR and sensing process with the UUV path planning problem, the relationships between target characteristics, sonar measurements and ATR are modeled by a probabilistic sensor model in the form of a joint probability mass function (PMF) learned from labeled data [15,17,32,34,53,83–85]. The method adopted here and developed in [17, 85] is reviewed here for completeness. Other Bayesian classification methods, such as [40, 63], can be similarly implemented to learn the joint PMF.

A sonar image may contain from zero to multiple targets (Fig. 3.1(a)). In this dissertation, each raw sonar image matrix is first processed to locate all possible targets and, then, it is segmented to obtain smaller image matrices that each contain only one target, e.g. using a matched filter. Let $t_k$ represent the time at which a target $i$ is detected in the sonar image and, thus, inside the sensor FOV. From its sonar image segmentation (Fig. 3.1(b)), a measurement set $Z_i$ may be obtained as follows. First, raw target features are extracted from each sonar image segmentation using a pre-trained convolution neural network (CNN), *AlexNet* [48]. Subsequently, the set of estimated features, $\hat{X}_{i1}, ..., \hat{X}_{if}$, and inferred classification, $\hat{Y}_i$, are obtained using the support vector machine (SVM) proposed in [85], such that $Z_i = \{\hat{X}_{i1}, ..., \hat{X}_{if}, \hat{Y}_i\}$ (Fig. 3.1(b)).

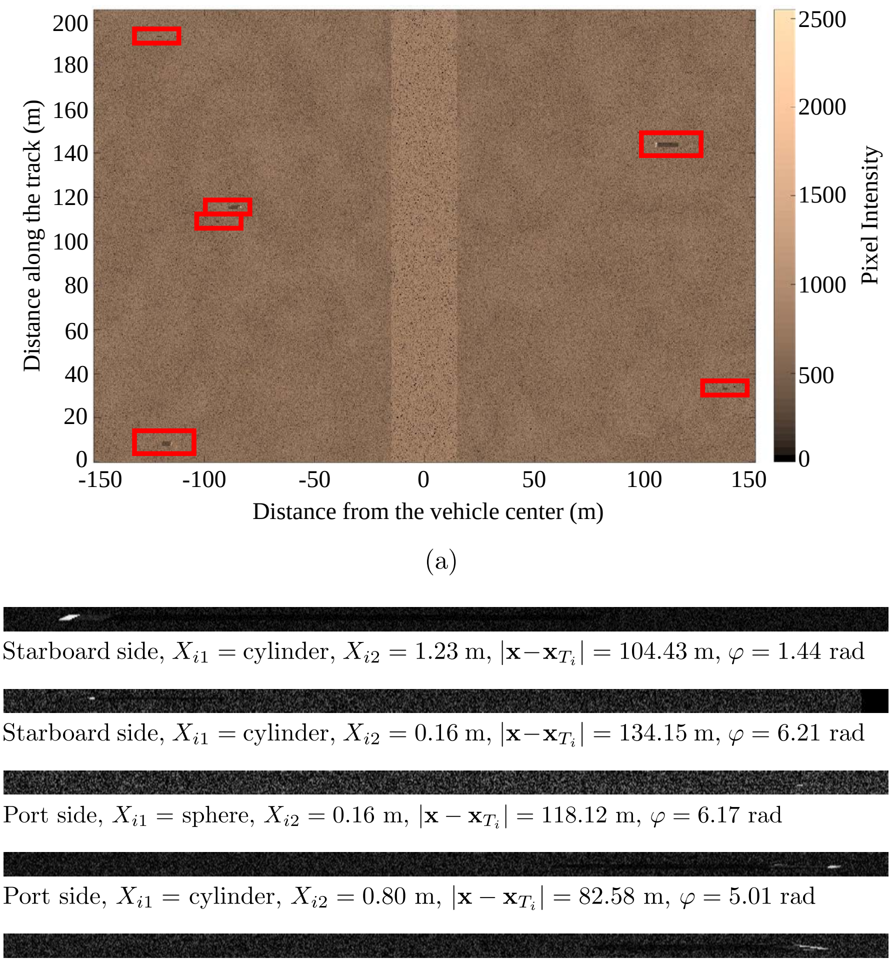



Figure 3.1: (a) Example of raw sonar image matrix, where each red box indicates a detected object, and (b) examples of sonar image segmentations and corresponding features extracted via CNN-SVM (adapted from [85], with permission).

Estimating the confidence level in consecutive sensor measurement processes requires a probabilistic Bayesian model that captures the influence of sensor mode, environmental conditions, and target features on the hidden target class and observable sensor measurements. The probabilistic sensor model can be defined as a joint PMF, and from the chain rule of probability,

$$p(Z_i, X_i, Y_i, \Lambda_i) = p(Z_i|\Lambda_i, X_i, Y_i)p(X_i|\Lambda_i, Y_i)p(Y_i|\Lambda_i)p(\Lambda_i) \tag{3.1}$$

Because $\Lambda_i$ is independent of $X_i$ and $Y_i$, and $p(Z_i|\Lambda_i, X_i, Y_i) = p(Z_i|\Lambda_i, X_i)$, the probabilistic sensor model is represented by (2.1). The conditional PMF, $p(Z_i \mid \Lambda_i, X_i)$, is also referred to as a sensor measurement model. The prior PMFs, $p(X_i, Y_i)$, $p(Y_i)$, and $p(\Lambda_i)$, can be computed either from the first principle, experiments, or simulation data; if this information is not available, the PMF can be assumed to be uniformly distributed. In this dissertation, the joint PMF is learned from sonar image data and represented by a Bayesian network (BN) model using a directed graph (Fig. 3.2) and a set of conditional probability tables (CPTs) that can be learned from the labeled data or constructed from the first principle. Target classification is performed based on the MAP rule using the posterior PMF, which can be computed recursively as follows

$$p(Y_i|M_i(k)) = \frac{p(E_i(k)|Y_i)p(Y_i|M_i(k-1))}{\sum_{Y_i} p(E_i(k)|Y_i)p(Y_i|M_i(k-1))} \tag{3.2}$$

The posterior probability of the chosen classification value provides the classification CL as a measure from zero to one of how probable the value is to be correct (where higher probability denotes higher confidence, with one representing certainty).

Because the CL can only be obtained after the image has been processed by the ATR algorithm, this dissertation utilizes the expected confidence level (ECL), defined as the one-step conditional expectation of the CL with respect to the

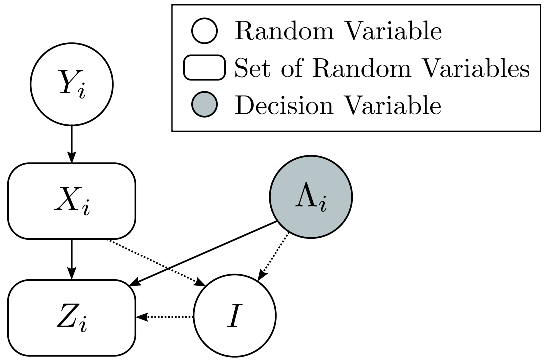



Figure 3.2: Probabilistic measurement model for a sonar imaging sensor. Dashed lines represent the ATR algorithms.

next (future) measurement that would be obtained at a possible sensor configuration. Assuming for simplicity that the environment is homogeneous and the sensor mode is fixed, $\Lambda_i$ represents all possible UUV viewpoints. Then, based on the evidence set at the present time $t_k$, i.e. $M_i(k) = \{E_i(1), ..., E_i(k)\}$ where $E_i(k) = \{Z_i(k), \Lambda_i(k)\}$, the ECL can be obtained as follows:

$$\hat{c}(\Lambda_i(k+1); M_i(k)) = \mathbb{E}_{Z_i(k+1)}\Big[\max_{y\in\mathcal{Y}} P(Y_i = y \mid M_i(k), E_i(k+1))\Big] \tag{3.3}$$

Note that $\Lambda_i$ is a decision variable, while $Z_i(k)$ is assumed unknown.

The ECL defined in (3.3) can be computed using the joint conditional probability, which corresponds to the sensor measurement model. By taking the expectation with respect to $Z_i$, the equation (3.3) can be written by

$$\hat{c}(\Lambda_i(k+1); M_i(k)) = \sum_{Z_i(k+1)} c(Y_i; M_i(k), E_i(k+1))p(Z_i(k+1)|M_i(k), \Lambda_i(k+1)) \tag{3.4}$$

The conditional PMF, $p(Z_i(k+1)|m_i(k), \Lambda_i(k+1))$, is calculated by marginalizing the joint probability from the sensor measurement model over the unknown target class $Y_i$,

$$p(Z_i(k+1)|M_i(k), \Lambda_i(k+1)) = \sum_{Y_i} p(Z_i(k+1)|Y_i, \Lambda_i(k+1))p(Y_i|M_i(k)) \tag{3.5}$$

where $p(Z_i(k+1)|Y_i, \Lambda_i(k+1))$ and $p(Y_i|M_i(k))$ can be obtained from the graphical model CPTs. Because multiple measurements are necessary for successful target classifications, a set of viewpoints must be planned for each target. Let the number of viewpoints required to meet the desired confidence level for the $i$th target be denoted by an unknown variable $n_i$. The ECL of the sensor, after visiting $(k+n_i)$ viewpoints, is given by

$$\begin{aligned} & \hat{c}(\{\Lambda_i(k+1), ..., \Lambda_i(k+n_i)\}; M_i(k)) \\ = \; & \mathbb{E}_{Z_i(k+1),...,Z_i(k+n_i)}\Big[c(Y_i; m_i(k), E_i(k+1), ..., E_i(k+n_i))\Big] \end{aligned} \tag{3.6}$$

Then, the expectation with respect to the next $n_i$ images can be computed by recursively updating the ECL at each time step, immediately after updating the actual CL (recursively) from (3.2) based on the ATR output. The expectation with respect to future measurements is computed as shown in (3.4).

CHAPTER 4

# INFORMATIVE MULTI-VIEW PLANNING (IMVP)

The nature of acoustic wave propagation processes is such that the sonar position and orientation relative to the target of interest greatly influence the image quality, as well as the information value of the ATR algorithm's output. Other directional sensors, such as synthetic aperture radar (SAR) and cameras, are similarly influenced by their position and aspect angle relative to the target. Furthermore, analysing and classifying complex targets requires obtaining multiple images using different positions and aspect angles. The IMVP approach presented in this chapter optimizes the sensor path in terms of the classification confidence, based on prior feature estimation, by both determining and planning the path that enables the best viewpoints. By this approach, it is possible to maximize the number of targets captured in each image based on their location and on the FOV geometry, while also minimizing the number of sensor viewpoints and images as well as the distance traveled by the UUV.

The IMVP approach developed in this dissertation utilizes the geometry of the sensor FOV and prior estimates of target positions and orientations to map targets onto UUV configuration space, obtaining the so-called C-targets (Section 4.1). The most valuable viewpoints and corresponding C-target regions are determined from the expected confidence level, using the probabilistic measurement model described in Chapter 3 (taken from [17]). Finally, the IMVP approach determines the shortest path that visits the C-target regions required to achieve the desired (expected) CL for each target by solving a generalized traveling salesman problem with neighborhoods, as shown in Section 4.2.

## 4.1 C-target Definition and Construction

A C-target region represents the subset of all UUV configurations that enable intersections between the target geometry and the sensor FOV, thus allowing sensor observations [15,32,53]. In this dissertation, each target is approximated as a point for simplicity and its position is used to construct the corresponding C-target, defined as:

**Definition 1 (C-Target)** *The C-target region of the $i^{th}$ point target located at $\mathbf{x}_{T_i}$ is the set of all UUV configurations for which the sensor FOV contains the target position, i.e.:*

$$\mathcal{CT}_i \triangleq \{\mathbf{q} \in \mathcal{C} \mid \mathbf{x}_{T_i} \in \mathcal{S}(\mathbf{q})\} \tag{4.1}$$

For the side-scan sonar FOV geometry in Fig. 2.3, the C-target can be derived in closed form as follows. Let $r_{\min} \in \mathbb{R}_{\geq 0}$ and $r_{\max} \in \mathbb{R}_{>0}$ denote the minimum and maximum distances at which a measurement can be obtained, for a known sensor range $\mathcal{D} = (r_{\min}, r_{\max})$. Because the sonar is installed on a mobile UUV, the FOV geometry is a function of the UUV configuration, $\mathbf{q} = [x\ y\ \theta]^T$, and for a side-scan sonar can be approximated by two line segments perpendicular to the vehicle heading:

$$\mathcal{S}(\mathbf{q}) = \left\{ \mathbf{p} \in \mathcal{W} \;\middle|\; \mathbf{p} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \pm \begin{bmatrix} r\cos(\psi + \frac{\pi}{2}) \\ r\sin(\psi + \frac{\pi}{2}) \\ 0 \end{bmatrix},\ r \in \mathcal{D} \right\} \tag{4.2}$$

From Definition 1, the C-target corresponding to target $i$ is given by,

$$\mathcal{CT}_i = \left\{ \mathbf{q} \in \mathcal{C} \;\middle|\; \mathbf{q} = \mathbf{q}_{T_i} \pm \begin{bmatrix} r\cos(\varphi + \theta_{T_i} + \frac{\pi}{2}) \\ r\sin(\varphi + \theta_{T_i} + \frac{\pi}{2}) \\ \varphi \end{bmatrix},\ r \in \mathcal{D},\ \varphi \in \mathbb{S}^1 \right\} \tag{4.3}$$

where $\mathbb{S}^1$ is a 1-dimensional manifold or *circle*,

$$\mathbb{S}^1 = \{(x, y) \mid x^2 + y^2 = 1\} \tag{4.4}$$

For easier visualization and algorithmic implementation, $[0, 2\pi)$ can replace $\mathbb{S}^1$ using a quotient space, $[0, 2\pi]/\sim$ , in which the identification declares that 0 and $2\pi$ are equivalent, denoted by $0 \sim 2\pi$. This quotient space homeomorphic to $\mathbb{S}^1$ "glues" 0 and $2\pi$ of $[0, 2\pi]$, i.e., the value of $\varphi \in \mathbb{S}$ runs from 0 up to $2\pi$ and then "wrap around" to 0 [51]. This manifold definition allows to represent the distance between two vehicle orientations by means of the minimum angle between them, avoiding the discontinuity at 0 and $2\pi$. An example C-target geometry is shown in Fig. 4.1, where parameters $r$ and $\varphi$ represent the sonar distance from the target and the aspect angle, respectively.

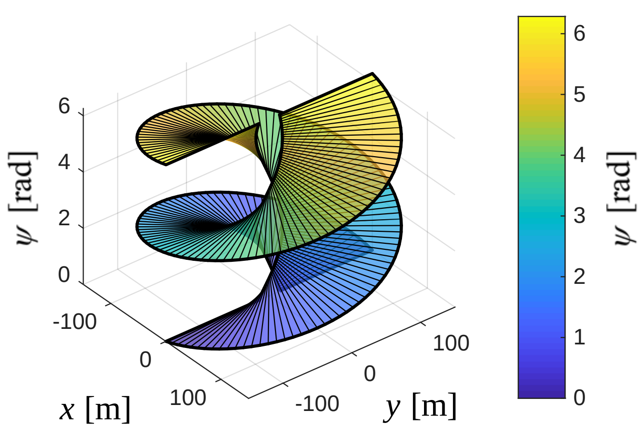



Figure 4.1: The C-target geometry of a target at $\mathbf{q}_i = [0\ 0\ 0]^T$ observed by a side-scan sonar with ranges $r_{\min} = 15$ and $r_{\max} = 150$.

Because the aspect angle $\psi \in \mathbb{S}^1$ wraps around every $2\pi$, the geometry of $\mathcal{CT}_i$ in $\mathcal{C} \subset \mathbb{R}^2 \times \mathbb{S}^1$ can be considered as a generalized helicoid in $\mathbb{R}^3$, defined in [1] as follows:

**Definition 2 (Generalized helicoid)** *Let $\Pi$ be a plane in $\mathbb{R}^3$, $l$ be a line in $\Pi$, and $\mathscr{C}$ be a point set in $\Pi$. Suppose $\mathscr{C}$ is rotated in $\mathbb{R}^3$ about $l$ and simultaneously displaced parallel to $l$ so that the speed of displacement is proportional to the speed of rotation, also called screw motion. Then, the resulting point set $\mathcal{M}(\mathscr{C}, c)$ is called the generalized helicoid generated by $\mathscr{C}$, also called the profile curve of $\mathcal{M}$. The line $l$ is called the axis of $\mathcal{M}$. The ratio of the speed of displacement to the speed of rotation is called slant of $\mathcal{M}$ and is denoted by $c$.*

Now, adopting Definition 2, the geometry of the $i^{th}$ C-target, $\mathcal{CT}_i$, corresponds to a generalized helicoid $\mathcal{M}(\mathscr{C}_i, 1)$, which is generated by the point set,

$$\mathscr{C}_i = \left\{ \mathbf{q} \in \mathcal{C} \;\middle|\; \mathbf{q} = \mathbf{q}_{T_i} \pm [0\ r\ 0]^T,\ r \in \mathcal{D} \right\} \tag{4.5}$$

on the plane,

$$\Pi_i = \{[x\ y\ \psi]^T \in \mathcal{C} \mid\ \cos(x - x_{T_i}) + \sin(y - y_{T_i}) = 0,\ \psi \in \mathbb{S}\} \tag{4.6}$$

by applying the screw motion to the line (axis),

$$l_i = \{[x\ y\ \psi]^T \in \mathcal{C} \mid x = x_{T_i},\ y = y_{T_i},\ \psi \in \mathbb{S}\} \tag{4.7}$$

with slant $c = 1$.

In order to transcribe the sensor planning problem (Problem 1) into a traveling-salesman problem, the relative UUV configuration is first discretized and, then, treated as the sensor operating condition ($\Lambda_i$) in the sensor measurement model (Chapter 3). This is accomplished by partitioning each C-target, $\mathcal{CT}_i$ ($i =$

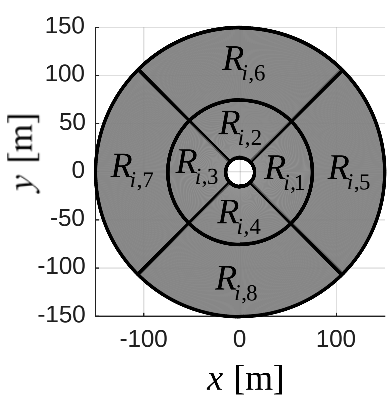
$1, \ldots, n$), into $M$ regions by uniformly discretizing the heading angle interval, $[0, 2\pi)$, and the sensor range, $\mathcal{D}$, as exemplified in Fig. 4.2. This approximation is useful because it reduces the computational complexity of the planning problem while maintaining sensing efficiency (since similar values of $r$ and $\varphi$ yield similar measurements). Other helicoid partitioning methods can also be applied, as explained in [9]. Now, a partition of $\mathcal{CT}_i$ is a pariwise disjoint family,

$$\mathcal{V}_i = \{R_{i,j} \mid j \in \{1, ..., M\}\} \tag{4.8}$$

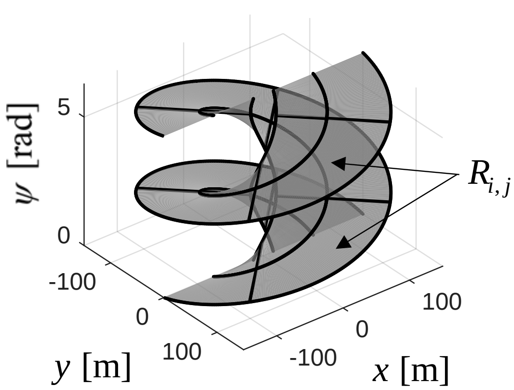
such that,

$$\bigcup_{j \in \{1,...M\}} R_{i,j} = \mathcal{CT}_i \tag{4.9}$$

and, throughout this dissertation, each element $R_{i,j}$ is referred to a *viewpoint region*. The viewpoint region $R_{i,j} \subset \mathcal{CT}_i$ represents a set of points that comprises two disjoint and congruent annular sectors on $\mathcal{CT}_i$ (Fig. 4.2). Since $\mathcal{CT}_i$ is periodic with a period $\psi = \pi$, one annular sector can be defined as translating another one with a distance $\psi = \pi$.




(a)




(b)

Figure 4.2: (a) Top view and (b) Isometric view of a partitioned C-target and corresponding viewpoint-regions geometries.

## 4.2 Generalized Traveling Salesman Problem with Neighborhoods (GTSPN) Formulation of IMVP Problem

A traditional approach to classifying multiple targets with high confidence is to continue obtaining measurements until the classification CL exceeds a desired threshold, $\varepsilon_{CL}$, for every target. Since the true target classification (ground truth) is unknown, this approach allows the user to reduce errors and uncertainty below an acceptable level decided based on the application of interest. In a similar vein, suppose an ECL threshold, $\hat{\varepsilon}_{CL}$, is chosen by the user *a priori* based on the desired level of confidence. After the ECL of every target is computed using the approach in Section 3, the ECL threshold is used to select a minimum number of viewpoint regions required to exceed $\hat{\varepsilon}_{CL}$. Because the actual CL may be lower than the ECL, a conservative choice typically assumes $\hat{\varepsilon}_{CL} > \varepsilon_{CL}$. Assuming the $n$ targets are independent (i.e., the features and classification of one target are independent of those of the other targets in the ROI), the minimum set of viewpoint regions or *neighborhoods* to be visited, denoted hereon by $\mathcal{R}_i \subset \mathcal{V}_i$, may be obtained independently and in any order. Then, a UUV path that visits every region $\mathcal{R}_i$ ($i = 1, \ldots, n$), guarantees that the images required to achieve the desired ECL will be obtained from every target or

$$\hat{c}(\mathcal{R}_i; M(k)) > \hat{\varepsilon}_{CL}, \quad \forall i, \quad i = 1, \ldots, n \tag{4.10}$$

Then, the solution to the informative path planning Problem 1 can be found by computing the shortest path between the $n$ regions.

The geometry of each neighborhood consists of the two congruent annular sectors defined in Section 4.1. Multiple neighborhoods intersect at UUV configurations that enable measurements from multiple targets. Under these properties and assumptions, the shortest UUV path visiting all of the target neighborhoods can be found by solving the following generalized traveling salesman problem:

**Problem 2 (GTSPN)** *Given a set of $m$ neighborhoods,*

$$\mathcal{R} = \left\{ R_\iota : \; R_\iota \in \bigcup_{i=1}^{n} \mathcal{V}_i, \;\; \iota = 1, ..., m \right\} \tag{4.11}$$

*find the minimum-time path that visits each neighborhood starting from the initial UUV configuration $\mathbf{q}_{(0)} \in \mathcal{C}$.*

The solution of *Problem 2* provides a time-optimal sensor path which is able to classify all $n$ targets in the ROI within a required expected classification confidence level, in minimum time (assuming for simplicity that the UUV travels at a constant speed). This generalized TSP problem seeks to find the shortest path that is guaranteed to visit every neighborhood in a possibly disjoint set at least once. Because an exact GTSPN solution is not available, the next section presents an algorithm for finding an approximate solution to Problem 2.

## 4.3 Approximate Solution of GTSPN with Intersecting Neighborhoods

Generalized forms of TSPs arise in many robot path planning and sensor coverage problems requiring the minimization of time and energy consumption (e.g. [3, 54] and references therein). Unlike traditional TSP formulations, in which an agent must visit every node in a graph or every point in a Euclidian space, in generalized TSPs (Fig. 4.3) the agent must visit any point in each (continuous) neighborhood or in each discrete set of points at least once [13, 73]. In *generalized TSP (GTSP)*, also known as *group TSP* [26] or *One-of-a-Set TSP* [58], one seeks to find the shortest tour that visits all of the predefined subsets of points at least once. In *TSP with neighborhoods (TSPN)* one seeks the shortest tour that intersects every continuous region at least once.

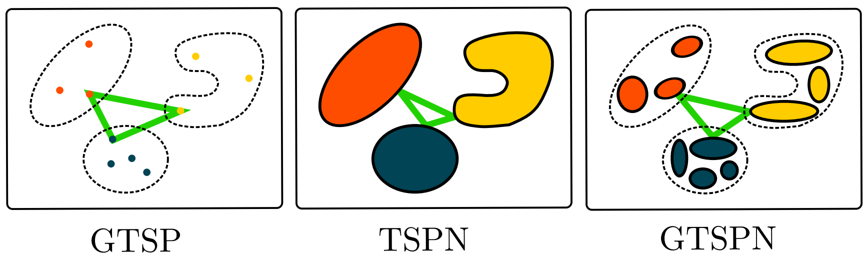



Figure 4.3: Graphical representations for the GTSP, TSPN, and GTSPN

As formulated in Problem 2, the UUV-based sonar path planning problem corresponds to a *GTSPN*, because one seeks to find the shortest tour that visits every neighborhood at least once, but each neighborhood consists of multiple (non-Euclidean) regions [75]. In particular, the neighborhoods in Problem 2 are each comprised of two disjoint continuous regions in the UUV configuration space:

**Problem 3 (GTSPN in Configuration Space)** *Find the shortest tour that visits every neighborhood in the set $\mathcal{R} = \{R_1, ..., R_m\}$, comprised of $n_\iota$ disjoint continuous regions, i.e.,*

$$R_\iota = \{S_{\iota,\xi} : \ S_{\iota,\xi} \subset \mathcal{C}, \ \xi = 1, ..., n_\iota\}, \quad \iota = 1, ..., m \tag{4.12}$$

Furthermore, the UUV configuration space is a smooth manifold that is locally like $\mathbb{R}^3$ but globally different. [49]. Nevertheless, the topology of $\mathcal{C}$ is the subset topology derived from the Euclidean metric [50, pg. 85].

Then, the minimum-time path can be approximated by the shortest path given the assumption on the constant UUV speed, and a distance metric can be defined as weighting the translating and rotating motions in the quotient space. GTSPN was first introduced in a Euclidean space in [75] and solved using a Hybrid Random-Key Genetic Algorithm (HRKGA). The high computational complexity of HRKGA was later reduced using a decoupled algorithm transcribing the GTSPN into a GTSP by first sampling a centroid of each region of every neighborhood set and, then, locally adjusting the waypoint locations toward the neighborhood boundaries to improve the solution [29]. A Growing Self-Organizing Array (GSOA) algorithm originally proposed in [28] was also applied to GTSPN in [29]. These existing algorithms are not ideally suited to solving Problem 3 because, when selecting waypoints, they do not take into account intersecting neighborhoods, which contain the most valuable configurations because they enable observations from multiple targets. Also, since the UUV path does not necessarily require returning to the initial configuration (tour), it is possible to first sample waypoints and, then, to compute their optimal ordering.

This dissertation presents a new GTSPN solution approach, referred to as IMVP, that is tailored to multi-view path planning and, thus, provides a more efficient solution to Problem 3 than existing GTSPN methods. Unlike previous methods, after constructing the $m$ neighborhoods from the C-targets (as shown in Section 4.1), the IMVP samples the neighborhood intersections using an approximate TSPN algorithm referred to as *TSPN-Intersecting* [27]. The TSPN-Intersecting algorithm uses the *hitting point set*, defined as a set of waypoints from each neighborhood, obtained such that a path connecting the hitting points intersects every neighborhood. Then, the IMVP approach seeks to sample a minimal number of hitting points by preferentially sampling the neighborhood intersections of highest degree, as follows. A collection of subsets, referred to as *minimal disjoint coverage set*, is defined such that the points sampled from each subset maximize the number of hitting points sampled from the intersection of neighborhoods (Fig. 4.4).

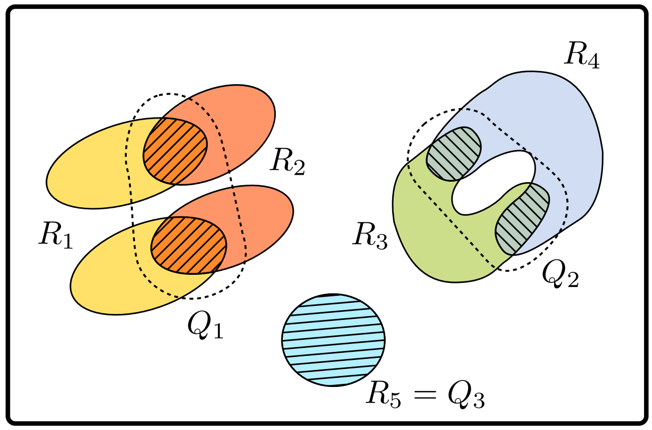



Figure 4.4: A minimal disjoint coverage set $\mathcal{Q} = \{Q_1, Q_2, Q_3\}$ of the set of neighborhoods $\mathcal{R} = \{R_1, ..., R_5\}$.

**Definition 3 (Minimal Disjoint Coverage Set)** *A set comprised of a minimum number of regions,*

$$\mathcal{Q} = \{Q_1, ..., Q_{m'}\} \tag{4.13}$$

*is a minimal disjoint coverage set of* $\mathcal{R} = \{R_1, ..., R_m\}$ *if the regions in* $\mathcal{Q}$ *are pairwise disjoint, and there exists* $\zeta \in \{1, ..., m'\}$ *such that* $Q_\zeta \subset R_\iota$ *for* $\iota = 1, ..., m$.

Then, the number of the disjoint regions satisfies $m' \leq m$, and the equality holds if the neighborhoods in $\mathcal{R}$ are disjoint, and $\mathcal{Q} = \mathcal{R}$. The greedy algorithm summarized in Algorithm 1 is developed in order to compute the minimal disjoint coverage set for a given IMVP neighborhood set, $\mathcal{R}$. The greedy search replaces any two intersecting regions with their mutual intersection for sampling. Each element $Q_\zeta \in \mathcal{Q}$ may consist of multiple disjoint continuous regions depending on the geometry, position, and orientation of the neighborhoods in $\mathcal{R}$. Thus, the hitting pointset is extended to a collection of node sets, $\mathcal{P} = \{P_1, ..., P_{m'}\}$, such that each node set $P_\zeta \subset Q_\zeta$, for $\zeta = 1, ..., m'$, consists of points sampled from each disjoint region in $Q_\zeta$ (Fig. 4.5). Different rules can be applied for sampling a point from each disjoint region in $Q_\zeta$: sampling a centroid if each region in $Q_\zeta$ is convex; sampling a pole of inaccessibility [35] if regions in $Q_\zeta$ are not convex; sampling a point on the boundary of each region in $Q_\zeta$ in order to obtain a shorter path. As a result, Problem 3 is reduced to a GTSP that seeks the shortest path visiting every node set in a collection $\mathcal{P} = \{P_1, ..., P_{m'}\}$, and, thus, can be solved efficiently as a classical asymmetric TSP using Noon and Bean transformation [62].

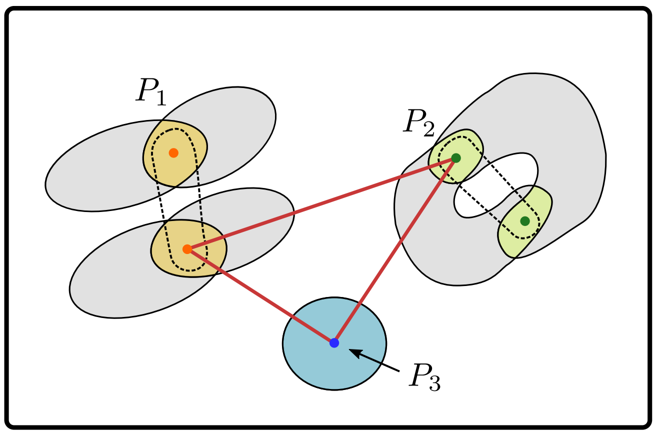



Figure 4.5: A collection of nodesets $\mathcal{P} = \{P_1, P_2, P_3\}$ from the minimal disjoint coverage set $\mathcal{Q}$ in Fig. 4.4. The shortest TSP tour is represented by a red line.

The previous sections show how, by considering the constraints and characteristics of the UUV-based imaging sensor, the sensor path planning problem defined in *Problem 2* can be reduced to the general GTSPN in *Problem 3* and, then, solved as an asymmetric TSP. Now, the geometry of the sensor FOV can be used to further simplify the computation required, as follows. In the case of a side-scan sonar (Fig. 2.1), each neighborhood consists of two congruent annular sectors (orange sectors in Fig. 4.6) translated by an angle $\psi = \pi$, because of the periodic geometry of the C-target. It can be easily shown that each element $Q_\zeta$ of the minimal disjoint

Algorithm 1: Greedy Search for Minimal Disjoint Coverage Set

**Require:** $\mathcal{R} = \{R_1, ..., R_m\}$

**Ensure:** Minimal disjoint neighborhood set of $\mathcal{R}$

initialize $\mathcal{Q} \leftarrow \mathcal{R}$

**while** every element in $\mathcal{Q}$ is not pairwise disjoint **do**

  **for all** pairs of elements $Q_i \in \mathcal{Q}$ and $Q_j \in \mathcal{Q}$ **do**

    **if** $Q_i \cap Q_j \neq \emptyset$ **then**

      replace $Q_i$ and $Q_j$ with $Q_i \cap Q_j$

    **end if**

  **end for**

**end while**

**return** $\mathcal{Q}$

coverage set $\mathcal{Q}$ (obtained by Algorithm 1) also consists of a pair of disjoint regions that are congruent and translated by $\psi = \pi$. Thus, the projection of $Q_\zeta$ onto the ROI is comprised of annular sectors, as shown in Fig. 4.6. In order to capture not only acoustic highlights but also shadows of targets inside sonar images for the purpose of classification, waypoints are chosen from the centroids (rather than from the boundary) of each region in $Q_\zeta$, as illustrated by the blue dots in Fig. 4.6, providing the node set $P_\zeta \in \mathcal{P}$ ($\zeta = 1, ..., m'$). Robustness to navigation errors and target uncertainty may be increased by choosing the waypoints directly using the swath planning approach developed in MAC algorithm [7]. By this approach, a waypoint characterized by the highest detection probability is chosen by considering the target field as a 2-dimensional Gaussian distribution and by modeling the sensor profile as a function of its range. This sensor profile function allows to model the degradation of sonar image quality as a function of range.

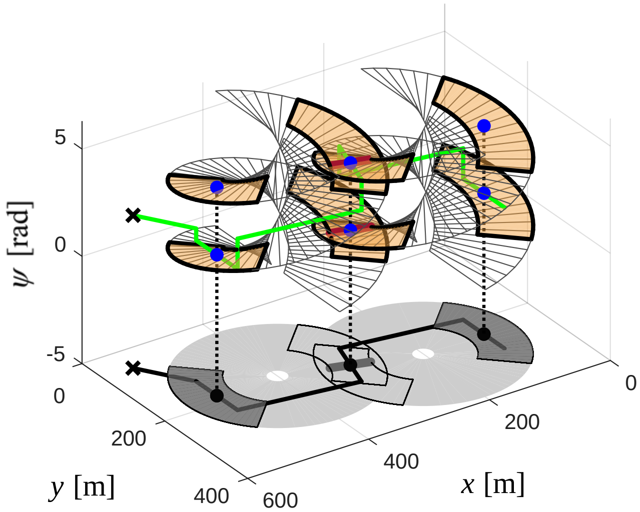



Figure 4.6: Illustrative example of shortest path (green solid line) in UUV configuration space obtained by IMVP algorithm for an initial condition symbolized by a black cross, through sampled waypoints symbolized by blue dots.

A symmetric TSP is obtained based on the following observations. From the neighborhood geometry, each node set $P_\zeta$ consists of two UUV configurations characterized by the same position but opposite headings, i.e.,

$$P_\zeta = \{[\mathbf{x}_{(\zeta)}^T \;\; \psi_{(\zeta,1)}]^T, [\mathbf{x}_{(\zeta)}^T \;\; \psi_{(\psi,2)}]^T\}, \quad \zeta = 1, ..., m' \tag{4.14}$$

such that

$$|\theta_{(\zeta,1)} - \theta_{(\zeta,2)}| = \pi \tag{4.15}$$

Then, the GTSP can be reduced to a symmetric TSP on a Euclidean plane which seeks the shortest path visiting every waypoint position $\mathbf{x}_{(1)}, ..., \mathbf{x}_{(m')}$ starting from the given initial UUV configuration, $\mathbf{q}_{(0)}$. From the waypoint positions, the shortest path can be computed using an existing TSP solver, such as Lin-Kernighan heuristic [52], by adding two dummy points: (1) $\mathbf{x}_{d,1}$, whose distances to all the other points are zero; (2) $\mathbf{x}_{d,2}$, which is only connected to $\mathbf{x}_{d,1}$ and $\mathbf{q}_{(0)}$ with zero distance. Adopting the LKH algorithm, in this dissertation, the path is obtained and, then, modified in order to take account for sensor's heading angles and geometric distortions of sonar images. This is accomplished by converting each waypoint $\mathbf{q}_{(\zeta)}$ ($\zeta = 1, ..., m'$) to a line segment of length $d$, i.e.,

$$\mathcal{T}_{(\zeta)} = \left\{ \mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = \mathbf{x}_\zeta + t \, [\cos\theta_\zeta \quad \sin\theta_\zeta]^T, \quad t \in \left[ -\frac{d}{2}, \frac{d}{2} \right] \right\} \tag{4.16}$$

where $d$ is chosen by the user based on the sensor application. Visiting each waypoint through this line segment path ensures that the UUV-based sonar is able to observe targets while traveling along a straight path with constant heading angle, as required for high image quality. Finally, the UUV path can be constructed by choosing the heading angle as $\theta_{(\zeta,1)}$ or $\theta_{(\zeta,2)}$, based on which node sequence results in the shortest path.

## 4.4 Computational Complexity Analysis

The IMVP solution algorithm is comprised of two steps: the first step is to obtain a set $\mathcal{Q}$ of downselect discretized C-target regions characterized by satisfactory ECL for every object in the ROI; the second step is to solve a TSPN problem and produce the shortest path that visits all regions in $\mathcal{Q}$ at least once. Then, the computational complexity of the first step is lower than $O(n \cdot 2^M)$, because checking every possible combinations of C-target regions takes $\sum_{k=1}^{M} \binom{M}{k} = 2^M$ steps, where $M$ is the number of partitioned viewpoint regions in every C-target. In this dissertation, the value $M = 8$ was found to provide a good approximation to the C-target region for the chosen sensor FOV. Let $L$ denote the number of regions selected to meet the desired ECL (assumed equal for all targets for the purpose of analysis only). Then, the next step of the TSPN solution (considering intersecting neighborhoods) has at worse a complexity $O(L^2)$. The relative values of $M$ and $L$, which depend on the target positions and characteristics, determine which of the terms $O(n \cdot 2^M)$ and $O(L^2)$ dominates the complexity of the algorithm.

It can be seen that the leading term in the computational complexity is $O(2^M)$, a term which derives from evaluating the ECL of every possible combination of discretized C-target regions. When the maximum number of viewpoint regions selected is fixed *a priori*, as in CMAC algorithms, this computation is reduced to $O(M)$, but the sensor may obtain more views than necessary in this case. Also, computing neighborhood intersections requires computation $O(L^2)$, dictating the computational complexity of the TSPN solution. The complexity of this computation too may be reduced, in this case, by adopting a greedy TSPN solution algorithm that does not consider neighborhood intersections. However, in this case, the sensor path may suboptimal and, in particular, and require longer travel times and more energy than the IMVP path solutions.

CHAPTER 5

# PHYSICS-INSPIRED SOLUTION TO TRAVELING SALESMAN PROBLEM WITH UNIFORM CIRCLE NEIGHBORHOODS

## 5.1 Introduction and Overview

This chapter presents a new solution approach to GTSPN that exploits the intersections of C-targets, or viewpoint regions, known to be critical to sensor path planning because they enable sensors to observe multiple objects simultaneously, thus reducing the operation time. The new GTSPN solution described in Chapter 4 employs heuristics that consider the intersection explicitly to obtain a shorter path. However, computing the intersecting regions is computationally expensive (Section 4.4) because all the possible pairs of neighborhoods must be considered as described in Algorithm 1. The expense of this computation may prevent robots with a limited onboard central processing units (CPUs), for example, to optimize the path online and, potentially, account for new information such as images obtained by other sensors or changed environmental conditions. Therefore, this chapter introduces a new TSPN numerical solution approach that obtains neighborhood intersections efficiently by means of recent computational geometry results [4, 31]. For simplicity, the approach is described for TSPN, and is developed for uniform circular neighborhoods. In future research, the approach will be extended to other neighborhood geometries, as well as, potentially to GTSPN solutions.

Due to its high problem complexity, most of the existing TSPN solution methods employ simplifying assumptions on the neighborhoods, such as convexity, fatness, and disjointness. For example, exact solution algorithm proposed in [36] formulates the TSPN as a mixed-integer nonlinear programming (MINLP) for convex polyhedral or ellipsoidal neighborhoods. The resulting global non-convex MINLP solver was shown to be computationally feasible up to the 15 neighborhoods. Another heuristic algorithm for neighborhoods with arbitrary shapes characterized by reasonable computational complexity was proposed in [2] and successfully demonstrated for problems comprised of up to 17 neighborhoods. Other heuristic methods have been developed by focusing on proving their approximation factors, e.g.: a constant-factor approximation for the neighborhoods with comparable diameters [23], a constant-factor approximation for convex, fat, and disjoint neighborhoods [22], and a $O(\log n)$-approximation for arbitrary neighborhoods [27] (where $n$ is the number of neighborhoods). Also, evolutionary algorithms are proposed in [19, 82] to solve TSPN for neighborhoods comprised of disjoint circles.

Even with the above simplifying assumptions, existing algorithms remain applicable to small ses of neighborhoods, thus preventing onboard path computation and adaptation for ROIs with many targets. Both space and time complexities are critical to planning paths, thus the geometric approach developed in this chapter seeks to closed-form representations of neighborhood intersections by means of coverage cones that can be computed with low time and space complexities. Coverage cones were originally developed and successfully applied to track coverage problems in [4, 31]. The novel geometric approach developed here adopts coverage cones to efficiently compute a minimum hitting sets, which is a set of minimal number of points such that all the given neighborhoods intersect with at least one of the points in the set [26]. Then, a TSPN tour is constructed by shooting a light ray that travels the shortest path based on Fermat's principle (also known as the principle of least time).

Assume the TSPN formulated in Problem 4 is characterized by $n$ viewpoint regions (or neighborhoods) that equal $n$ circles of equal circles that are not necessarily disjoint. This chapter develops an efficient approximate solution to the following problem:

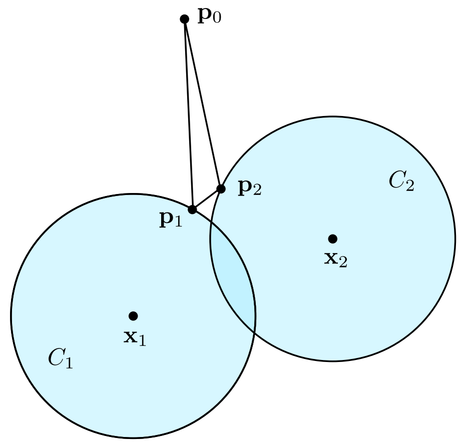



Figure 5.1: Example problem and notations of TSPN with two unit circles.

**Problem 4 (TSPN with Unit Circles)** *The traveling salesman problem with neighborhoods (TSPN) with unit circles seeks to find the shortest tour that visits $n$ known unit circles in a Euclidean plane starting from and returning to point $\mathbf{p}_0 \in \mathbb{R}^2$. For every circle $C_i$, the center $\mathbf{x}_i \in \mathbb{R}^2$ is known such that,*

$$C_i \triangleq \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x} - \mathbf{x}_i\| \leq r^2\}, i = 1, \ldots, n \tag{5.1}$$

*where $r = 1$.*

Because $\mathbf{p}_0$ is given, it can be considered as another circular neighborhood with zero radius. A hitting pointset, or set of waypoints, is defined as $P \subset \bigcup_{i=1}^{n} C_i$, such that $P \cap C_i \neq \emptyset$ for $i = 1, ..., n$, as shown in Fig. 5.1 [26]. Then, the hitting point set is defined as $P = \{\mathbf{p}_1, ..., \mathbf{p}_n\}$, where $\mathbf{p}_i \in C_i$ for $i = 1, ..., n$. The sequence of circles to visit is described as a permutation of the indices, $\Sigma = (\sigma_1, ..., \sigma_n)$, such that $1 \leq \sigma_i \leq n$, for any pair $\sigma_i, \sigma_j \in \{1, ..., n\}$, where $\sigma_i \neq \sigma_j$. Then, the shortest tour is computed by finding $P$ and $\Sigma$ such that the length of the tour,

$$L(P, \Sigma) \triangleq \sum_{i=1}^{n} \|\mathbf{p}_i - \mathbf{p}_{i-1}\| + \|\mathbf{p}_0 - \mathbf{p}_n\| \tag{5.2}$$

is minimized.

## 5.2 Exact Solutions of TSPN with Two Unit Circles

As a first stpe, the exact solutions of TSPNs with two intersecting unit circles is considered to understand how the shortest TSPN tour changes depending on the circle's positions. Intuitively, it can be seen that the shortest tour depends only on the relative positions and on $\mathbf{p}_0$. The configuration of two circles is uniquely defined as their angle of intersection [46]. The angle of intersection of two intersecting circles, denoted by $\varphi$, is defined as the angle between their tangents at either of the intersection points, as shown in Fig. 5.2(a). Two circles are said to be orthogonal if and only if $\varphi = \frac{\pi}{2}$, as shown in Fig. 5.2(b), and described in [21].

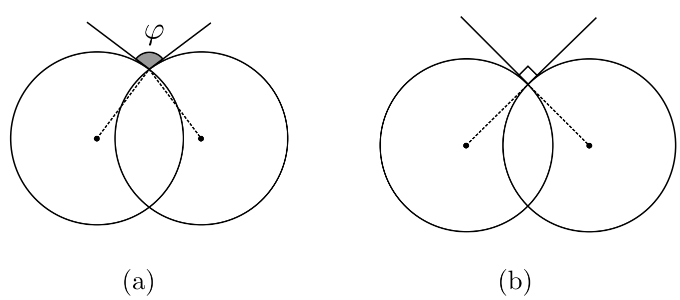



Figure 5.2: (a) Angle of intersection definition, and (b) example of orthogonal circles

In order to find the exact solution, $\mathbf{x}_1$ and $\mathbf{x}_2$ are considered in a Cartesian coordinate. Without loss of generality, let $\mathbf{x}_1$ and $\mathbf{x}_2$ be located at $A(-p, 0)$ and

$B(p, 0)$, where $0 \leq p \leq r$. Then, the two intersecting points of the two circle's boundaries on the $y$-axis are denoted by $C(p, \sqrt{r^2 - p^2})$ and $C'(p, -\sqrt{r^2 - p^2})$ as shown in Fig. 5.3. The starting point $\mathbf{p}_0$ is located at $D(k \sin \psi, k \cos \psi)$, where $-\pi \leq \psi \leq \pi$ and $k > 0$. Let $\theta_1$ and $\theta_2$ denote the angles of $(\mathbf{p}_1 - \mathbf{x}_1)$ and $(\mathbf{p}_2 - \mathbf{x}_2)$ with respect to the $x$-axis counterclockwise and clockwise, respectively. Then, $\mathbf{p}_1$ is located at $E(-p + r \cos \theta_1, r \sin \theta_1)$, and $\mathbf{p}_2$ is located at $F(p - r \cos \theta_1, r \sin \theta_1)$, $0 \leq \theta_1, \theta_2 \leq 2\pi$. Note that the relationship between the angle of intersection and

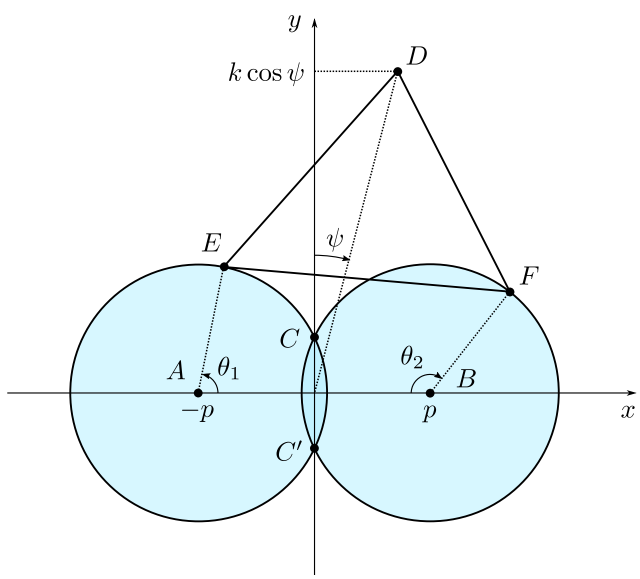



Figure 5.3: TSPN with two intersecting unit circles specified in terms of parameters $p, k, \theta_1, \theta_2$, and $\psi$

the the distance between two centers is,

$$\frac{\varphi}{2} = \arccos\left(\frac{p}{r}\right) \tag{5.3}$$

where $0 \leq \varphi \leq \pi$ and $0 \leq \frac{p}{r} \leq 1$. Based on the relative values of the above TSPN parameters, the optimal solution falls into one of the special cases described in the following subsections.

### 5.2.1 Case Study: Symmetric Configurations

First case study is performed to identify the circle configurations that the exact TSPN solutions visit the neighborhoods intersections. For simplicity, the case in which $C$, $C'$, and $D$ are collinear, i.e., the starting point is on the $y$-axis, as shown in Fig. 5.4, is considered. It can be seen that $E$ and $F$ are symmetric about the $y$-axis, thus let $\theta_1 = \theta_2 = \theta$, such that $\frac{\varphi}{2} \leq \theta \leq \frac{\pi}{2}$ and $k > \sqrt{r^2 - p^2}$. Now, let $\theta^*$ denote the optimal value of $\theta$. Then, the following result provides the optimal TSPN solutions for $n = 2$.

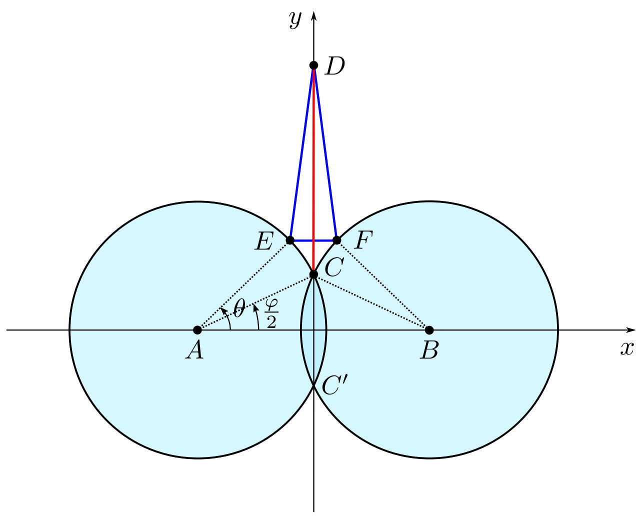



Figure 5.4: Symmetric configuration in TSPN on two unit circles

**Proposition 1** *If $0 \leq \varphi \leq \frac{\pi}{2}$, the optimal angle $\theta^* = \frac{\varphi}{2}$ provides the shortest TSPN tour in terms of the waypoints $\{D, C, D\}$ $C$.*

**Proof:** Based on the configuration in Fig. 5.4,

$$\overline{CD} = k - \sqrt{r^2 - p^2} \tag{5.4}$$

$$\overline{DE} = \sqrt{(k - r\sin\theta)^2 + (r\cos\theta - p)^2} \tag{5.5}$$

$$\frac{1}{2}\overline{EF} = p - r\cos\theta \tag{5.6}$$

Then, the half of the TSPN tour length is

$$f(\theta) = \overline{DE} + \frac{1}{2}\overline{EF} \tag{5.7}$$

$$= \sqrt{(k - r\sin\theta)^2 + (r\cos\theta - p)^2} + p - r\cos\theta \tag{5.8}$$

Since $f(\theta)$ is convex, $\theta^* = \frac{\varphi}{2}$ if and only if $\left.\frac{\mathrm{d}f(\theta)}{\mathrm{d}\theta}\right|_{\theta=\frac{\varphi}{2}} \geq 0$.

$$\frac{\mathrm{d}f(\theta)}{\mathrm{d}\theta} = \frac{-kr\cos\theta + pr\sin\theta}{\sqrt{(k - r\sin\theta).^2 + (p - r\cos\theta).^2}} + r\sin\theta \tag{5.9}$$

Because $\cos\varphi = \frac{p}{r}$, $\sin\varphi = \frac{\sqrt{r^2-p^2}}{r}$, and $k > \sqrt{r^2 - p^2}$

$$\left.\frac{d}{d\theta}\right|_{\theta=\frac{\varphi}{2}} = \frac{-kr\cos\varphi + pr\sin\varphi}{\sqrt{(k - r\sin\varphi).^2 + (p - r\cos\varphi).^2}} + r\sin\varphi \tag{5.10}$$

$$= \frac{-kp + p\sqrt{r^2 - p^2}}{\sqrt{(k - \sqrt{r^2 - p^2}).^2 + (p - p).^2}} + \sqrt{r^2 - p^2} \tag{5.11}$$

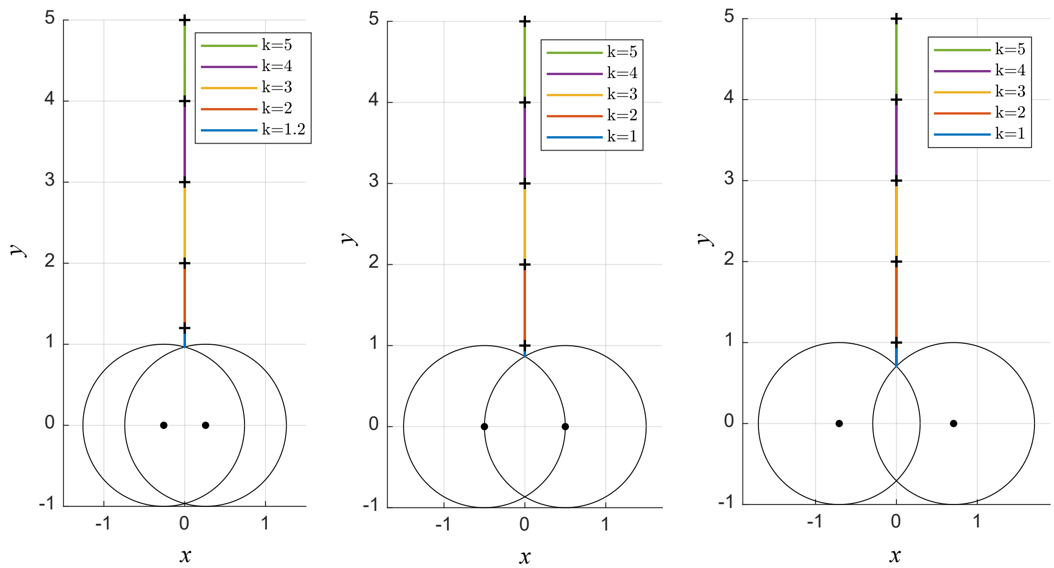
$$= \frac{p(-k + \sqrt{r^2 - p^2})}{k - \sqrt{r^2 - p^2}} + \sqrt{r^2 - p^2} \tag{5.12}$$

$$= -p + \sqrt{r^2 - p^2} \tag{5.13}$$

Then, $-p + \sqrt{r^2 - p^2} \geq 0$ if and only if $0 \leq p \leq \frac{1}{\sqrt{2}}r$. Therefore, from (5.3), $\theta^* =\leq \frac{\varphi}{2}$ if and only if $0 \leq \varphi \leq \frac{\pi}{2}$. □

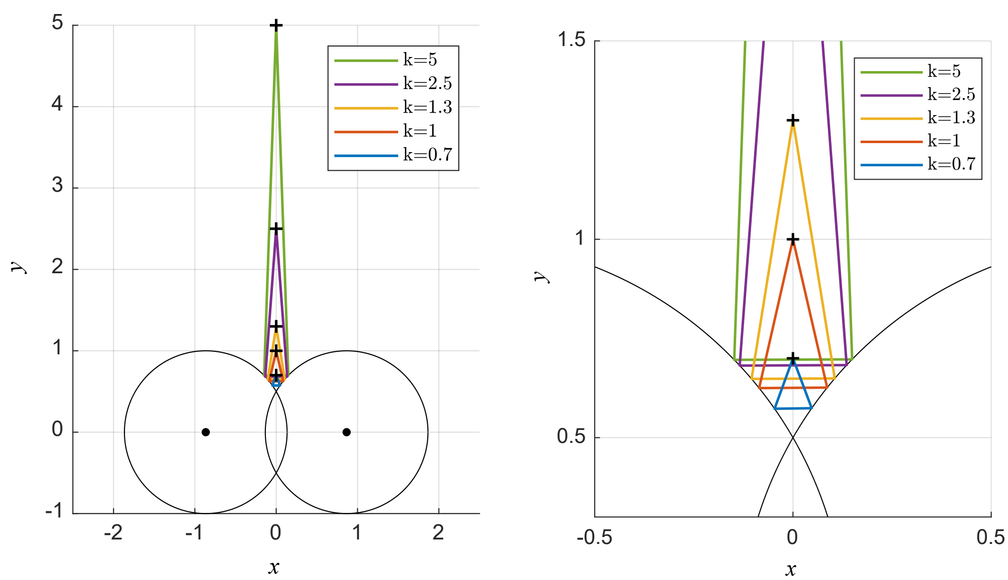
**Proposition 2** *If $\frac{\pi}{2} < \varphi \leq \pi$, the optimal angle satisfies $\theta^* > \frac{\pi}{2}$, and the shortest tour does not visit the intersection of two circles, and, instead, visits two different waypoints in each circle.*

**Proof:** Similarly to **Proposition 1**, $\theta^* > \frac{\varphi}{2}$ if and only if $\left.\frac{\mathrm{d}f(\theta)}{\mathrm{d}\theta}\right|_{\theta=\frac{\varphi}{2}} < 0$. Also, $-p + \sqrt{r^2 - p^2} < 0$ if and only if $p > \frac{1}{\sqrt{2}}r$. Therefore, from (5.3), $\theta^* > \frac{\varphi}{2}$ if and only if $\frac{\pi}{2} < \varphi \leq \frac{\pi}{2}$. □




(a) (b) (c)

Figure 5.5: Exact TSPN solution when (a) $\varphi = \frac{5\pi}{6}$ (b) $\varphi = \frac{2\pi}{3}$ (c) $\varphi = \frac{\pi}{2}$, starting from a black cross.




(a) (b)

Figure 5.6: (a) Exact TSPN solutions when $\varphi = \frac{\pi}{3}$ starting from a black cross; (b) a zoomed view of (a)

Note that whether the shortest tour visits the intersection is independent of the $k$ value of the initial position, parameterized by $k$. In other words, the shortest tour may or may not visit the circles' intersection depending on the angle of intersection, regardless of the position of the starting point. Numerical examples illustrating Propositions 1 and 2 are shown in Figs . 5.5 and 5.6, respectively.

### 5.2.2 General Case: Asymmetric Configurations

The symmetric circles results in the previous subsection motivate the theoretical results on the angle of intersection of the two circles in the TSPN solution. This section considers the general cases when $\psi$ is not necessarily zero but in which the angle of intersection can take any value, or, more precisely, $\psi \in [-\pi, \pi]$. The results proven in this section are utilized to develop efficient TSPN solutions for $n > 2$ in the remainder of the chapter. The numerical results in Section 5.4 demonstrate the efficiency of the proposed solution for TSPNs commonly arising in directional sensor planning problems and IMVP solutions.

#### Two Intersecting Circles with Angle of Intersection $\varphi$, where $\frac{\psi}{2} < \varphi \leq \pi$

When the two intersecting circles satisfy $\frac{\pi}{2} < \varphi \leq \pi$, the initial position determines the optimal tour. In particular, whether the starting point is included in two intersection cones decides the exact TSPN solution. The *intersection cones*, denoted by $\mathcal{K}^+$ and $\mathcal{K}^-$, are defined as a set of transversals that are in between the two line transversals that passes each circle center and the intersection of two circles' boundaries, as shown in Fig. 5.7.

Let $\mathbf{x}^+$ and $\mathbf{x}^-$ denote the two intersecting points of the two circles' boundaries.

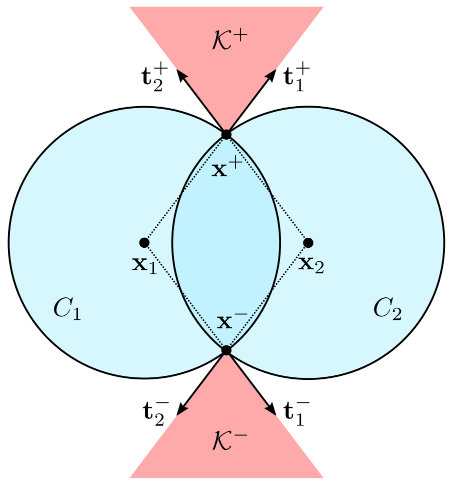



Figure 5.7: Intersection cones constructed by two intersecting unit circles

Then, the unit vectors from $\mathbf{x}_1$ and $\mathbf{x}_2$ to $\mathbf{x}^+$ are denoted by $\mathbf{t}_1^+$ and $\mathbf{t}_2^+$, respectively, and defined as

$$\mathbf{t}_1^+ = \frac{\mathbf{x}^+ - \mathbf{x}_1}{\|\mathbf{x}^+ - \mathbf{x}_1\|};\ \mathbf{t}_2^+ = \frac{\mathbf{x}^+ - \mathbf{x}_2}{\|\mathbf{x}^+ - \mathbf{x}_2\|} \tag{5.14}$$

These two unit vectors with respect to $\mathbf{x}_+$ are referred to as the lower and upper vectors of the intersection cone $\mathcal{K}^+$. The intersecting point $\mathbf{x}_+$ corresponds to the apex of the intersection cone $\mathcal{K}^+$. Similarly, let unit vectors $\mathbf{t}_1^-$ and $\mathbf{t}_2^-$ with respect to $\mathbf{x}_-$ be defined as

$$\mathbf{t}_1^- = \frac{\mathbf{x}^- - \mathbf{x}_1}{\|\mathbf{x}^- - \mathbf{x}_1\|};\ \mathbf{t}_2^- = \frac{\mathbf{x}^- - \mathbf{x}_2}{\|\mathbf{x}^- - \mathbf{x}_2\|} \tag{5.15}$$

Then, the two unit vectors $\mathbf{t}_1^-$ and $\mathbf{t}_2^-$ are referred to as the lower and upper vectors of the intersection cone $\mathcal{K}^-$. The intersecting point $\mathbf{x}_-$ corresponds to the apex of the intersection cone $\mathcal{K}^-$. The intersection cones for the two intersecting TSPN circles are defined as

$$\mathcal{K}^+ = \text{cone}(\mathbf{t}_1^+, \mathbf{t}_2^+, \mathbf{x}^+) \triangleq \{\alpha_1\mathbf{t}_1^+ + \alpha_2\mathbf{t}_2^+ + \mathbf{x}^+ \mid \alpha_1, \alpha_2 \in \mathbb{R}_+\} \tag{5.16}$$

$$\mathcal{K}^- = \text{cone}(\mathbf{t}_1^-, \mathbf{t}_2^-, \mathbf{x}^-) \triangleq \{\alpha_1\mathbf{t}_1^- + \alpha_2\mathbf{t}_2^- + \mathbf{x}^- \mid \alpha_1, \alpha_2 \in \mathbb{R}_+\} \tag{5.17}$$

The exact TSPN solution is then found based on the position of $\mathbf{p}_0$ relative to $\mathcal{K}^+ \cup \mathcal{K}^-$, which is summarized in the following Propositions 3 and 4. Example simulation results for Propositions 3 and 4 are included in Fig. 5.8 and 5.9, respectively. To summarize, the exact TSPN solution visits the intersection of two circle's boundaries, $\mathbf{x}_+$ or $\mathbf{x}_-$ , if $\mathbf{p}_0 \in \mathcal{K}^+ \cup \mathcal{K}^-$. Otherwise, the exact TSPN solution aligns with the line segment that connects $\mathbf{p}_0$ and either of $\mathbf{x}_1$ or $\mathbf{x}_2$, whichever is farther from $\mathbf{p}_0$.

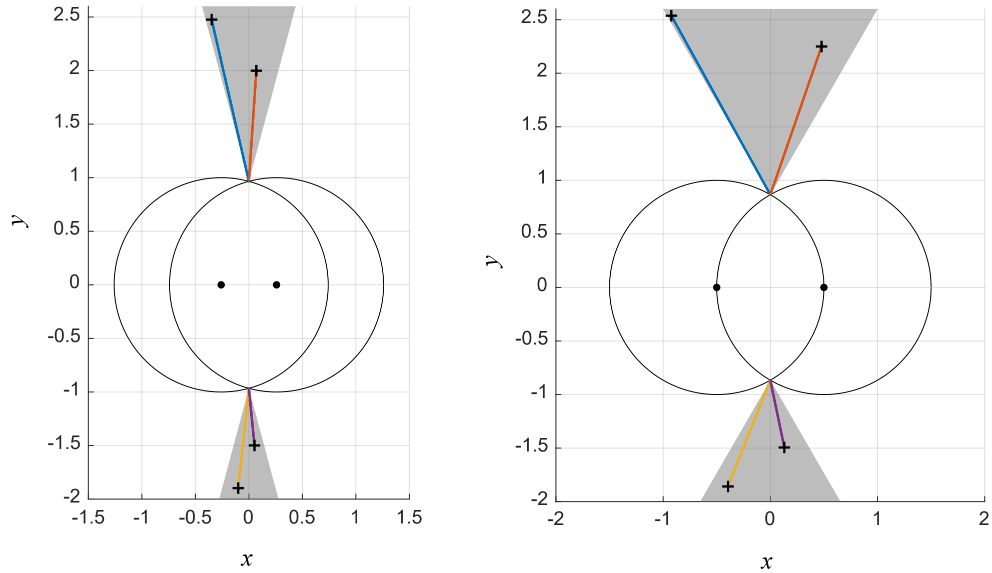
**Proposition 3** *If the starting point $\mathbf{p}_0$ satisfies $\mathbf{p}_0 \in \mathcal{K}^+ \cup \mathcal{K}^-$, the shortest TSPN tour visits $\mathbf{x}^+$ or $\mathbf{x}^-$, and thus, simply finding the closest point inside $C_1 \cap C_2$ from $\mathbf{p}_0$ results in the exact TSPN solution.*

**Proposition 4** *If the starting point $\mathbf{p}_0$ satisfies $\mathbf{p}_0 \notin \mathcal{K}^+ \cup \mathcal{K}^-$, the shortest TSPN tour visits the waypoints $\mathbf{p}_1 \in C_1$ and $\mathbf{p}_2 \in C_2$ such that either $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{x}_1$ are collinear or $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{x}_2$ are collinear. In other words, the exact TSPN tour can be computed by finding the closest point from $\mathbf{p}_0$ to either $C_1$ or $C_1$, whichever is farther from $\mathbf{p}_0$.*

#### Two Intersecting Circles with Angle of Intersection $\varphi$, where $0 \leq \varphi \leq \frac{\pi}{2}$

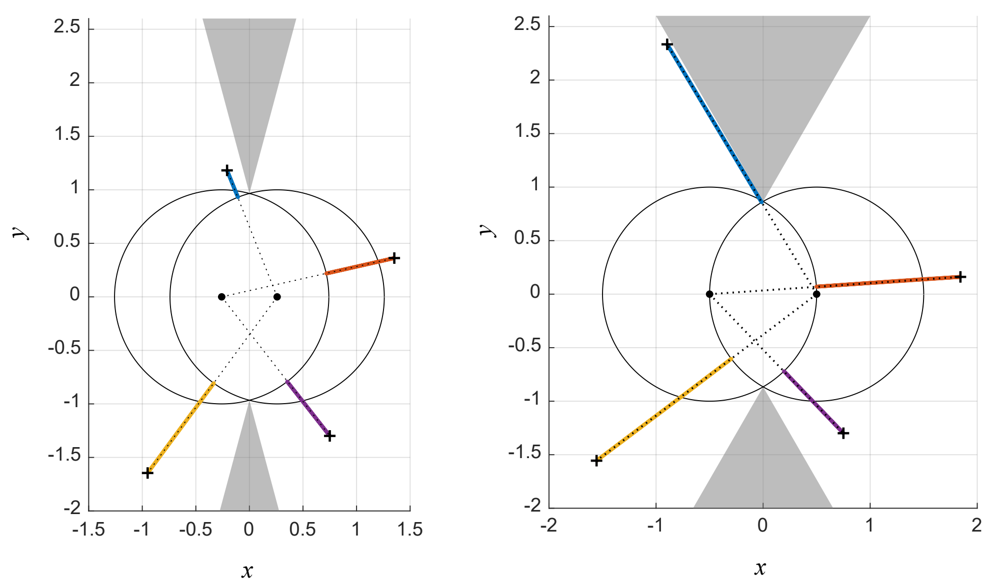
When the two intersecting circles satisfy $0 \leq \varphi \leq \frac{\pi}{2}$, the initial position determines the optimal tour. In particular, whether the starting point is included in two tangential cones decides the exact TSPN solution. The *tangential cones*, denoted by $\mathcal{T}^+$ and $\mathcal{T}^-$, are defined as a set of transversals that are in between the two tangential lines from $\mathbf{x}_1$ to $C_2$ and from $\mathbf{x}_2$ to $C_1$, as shown in Fig. 5.10.

Let $\mathbf{u}^+$ and $\mathbf{u}^-$ denote the intersections of the two tangent lines to each circle




(a) (b)

Figure 5.8: Simulation results of Proposition 3 (a) $\varphi = \frac{\pi}{6}$ and (b) $\varphi = \frac{\pi}{3}$




(a) (b)

Figure 5.9: Simulation results of Proposition 4 when (a) $\varphi = \frac{\pi}{6}$ and (b) $\varphi = \frac{\pi}{3}$

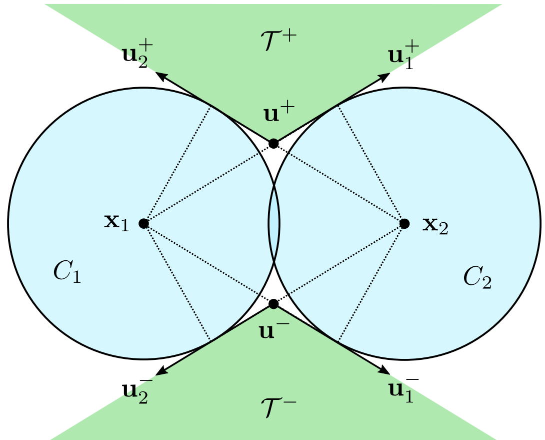



Figure 5.10: Tangential cones constructed by two intersecting unit circles

$C_1$ and $C_2$ from $\mathbf{x}_2$ and $\mathbf{x}_1$ on the two sides of the circles. Then, two unit vectors, denoted by $\mathbf{u}_1^+$ and $\mathbf{u}_2^+$, are defined as the unit vectors representing lower and upper bounds of the tangential cones.

$$\mathbf{u}_1^+ = \frac{\mathbf{u}^+ - \mathbf{x}_1}{\|\mathbf{u}^+ - \mathbf{x}_1\|};\ \mathbf{u}_2^+ = \frac{\mathbf{u}^+ - \mathbf{x}_2}{\|\mathbf{u}^+ - \mathbf{x}_2\|} \tag{5.18}$$

These two unit vectors with respect to $\mathbf{u}^+$ are referred to as lower and upper vectors for the tangential cone $\mathcal{T}^+$. Similarly, unit vectors $\mathbf{u}_1^-$ and $\mathbf{u}_2^-$ are defined as

$$\mathbf{u}_1^- = \frac{\mathbf{u}^- - \mathbf{x}_1}{\|\mathbf{u}^- - \mathbf{x}_1\|};\ \mathbf{u}_2^- = \frac{\mathbf{u}^- - \mathbf{x}_2}{\|\mathbf{u}^- - \mathbf{x}_2\|} \tag{5.19}$$

These two unit vectors with respect to $\mathbf{u}^-$ are referred to as lower and upper vectors for the tangential cone $\mathcal{T}^-$. Two tangential cones $\mathcal{T}^+$ and $\mathcal{T}^-$ are defined as

$$\mathcal{T}^+ = \text{cone}(\mathbf{u}_1^+, \mathbf{u}_2^+, \mathbf{u}^+) \triangleq \{\alpha_1 \mathbf{t}_1^+ + \alpha_2 \mathbf{u}_2^+ + \mathbf{u}^+ \mid \alpha_1, \alpha_2 \in \mathbb{R}_+\} \tag{5.20}$$

$$\mathcal{T}^- = \text{cone}(\mathbf{u}_1^-, \mathbf{u}_2^-, \mathbf{u}^-) \triangleq \{\alpha_1 \mathbf{u}_1^- + \alpha_2 \mathbf{u}_2^- + \mathbf{u}^- \mid \alpha_1, \alpha_2 \in \mathbb{R}_+\} \tag{5.21}$$

The exact TSPN solution is then found based on the position of $\mathbf{p}_0$ relative to $\mathcal{T}^+ \cup \mathcal{T}^-$, which is summarized in the following Propositions 5 and 6. Example simulation results for Propositions 5 and 6 are shown in Fig. 5.11 and 5.12, respectively. To summarize, the exact TSPN solution does not visit the intersection of two circles but visits each circle via two different waypoints if $\mathbf{p}_0 \in \mathcal{T}^+ \cup \mathcal{T}^-$. Otherwise, the exact TSPN solution aligns with the line segment that connects $\mathbf{p}_0$ and either of $\mathbf{x}_1$ or $\mathbf{x}_2$, whichever is farther from $\mathbf{p}_0$.

**Proposition 5** *If the starting point $\mathbf{p}_0$ satisfies $\mathbf{p}_0 \in \mathcal{T}^+ \cup \mathcal{T}^-$, the shortest TSPN tour visits two different waypoints $\mathbf{p}_1 \in C_1$ and $\mathbf{p}_2 \in C_2$ such taht $\mathbf{p}_1, \mathbf{p}_2 \notin C_1 \cap C_2$.*

**Proposition 6** *If the starting point $\mathbf{p}_0$ satisfies $\mathbf{p}_0 \notin \mathcal{T}^+ \cup \mathcal{T}^-$, the shortest TSPN tour visits the waypoints $\mathbf{p}_1 \in C_1$ and $\mathbf{p}_2 \in C_2$ such that either $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{x}_1$ are collinear or $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{x}_2$ are collinear. In other words, the exact TSPN tour can be computed by finding the closest point from $\mathbf{p}_0$ to either $C_1$ or $C_1$, whichever is farther from $\mathbf{p}_0$.*

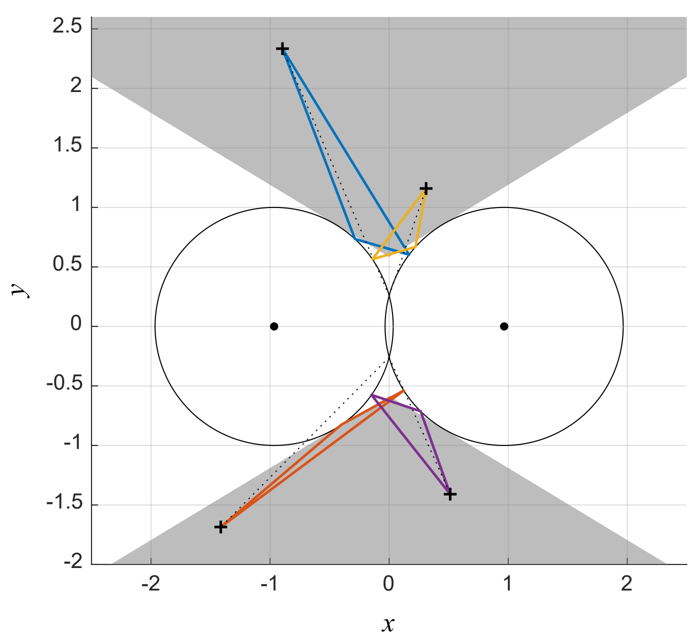



Figure 5.11: Simulation results of Proposition 5 when $\varphi = \frac{\pi}{6}$

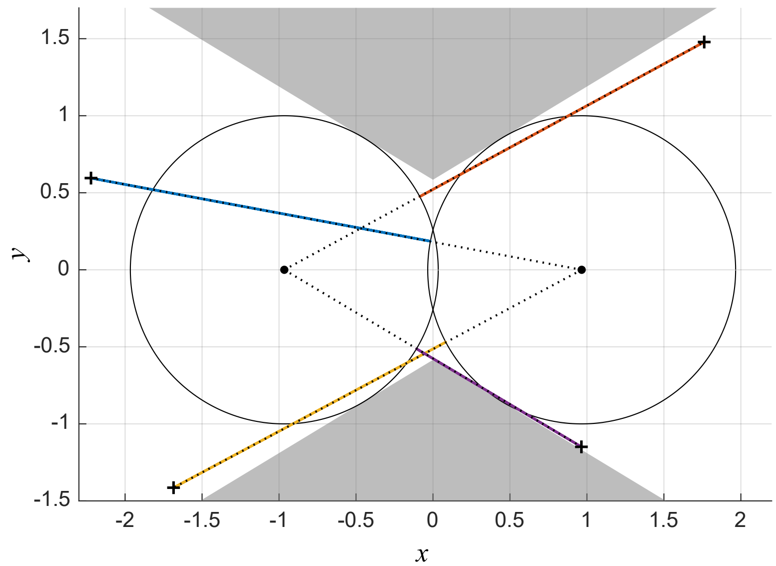


Figure 5.12: Simulation results of Proposition 6 when $\varphi = \frac{\pi}{6}$

### 5.2.3 Fermat's Principle Application to TSPN Solution

This subsection introduces the proof of Proposition 3 and 5 based on Fermat's principle. In the proof of Proposition 3 and 5, it is necessary to show whether the shortest TSPN tour visits the intersection of two circles. Based on Fermat's principle, which says the light takes the path of least time, if the TSPN tour does not visits the intersection of the two circle's boundaries, the exact TSPN solution corresponds to the light ray path that travels from $\mathbf{p}_0$, reflects at $\mathbf{p}_1 \in C_1$ and $\mathbf{p}_2 \in C_2$ such that $\mathbf{p}_1, \mathbf{p}_2 \notin C_1 \cap C_2$, and then travels back to $\mathbf{p}_0$, following the law of reflection (Fig. 5.13). Thus, if such light ray-like TSPN tour does not exist, the exact TSPN solution visits the intersecting point of the two circle's boundaries. Proposition 4 and 6 are skipped as the proof is intuitive.

In the following proofs, the TSPN circles, starting point, and waypoints are represented in a Cartesian coordinate system, as shown in Fig. 5.3. Without loss of generality, the circle centers $\mathbf{x}_1$ and $\mathbf{x}_2$ are located at $A(-p, 0)$ and $B(p, 0)$,

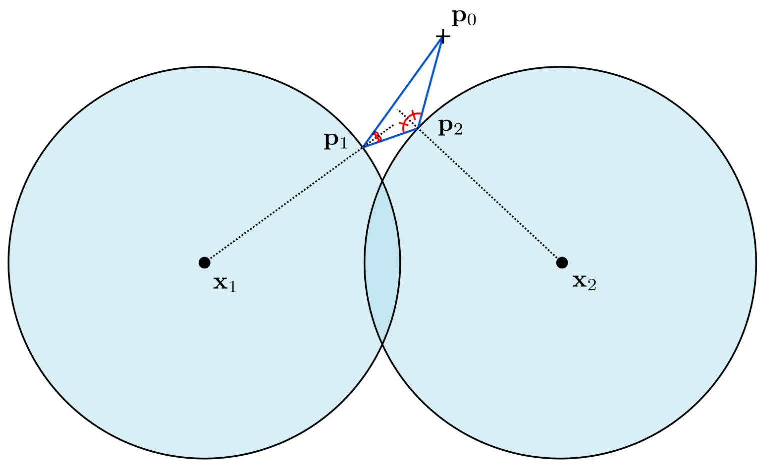



Figure 5.13: Fermat's principle used to prove Propositions 3 and 5.

where $0 \leq p \leq r$ and $r$ is the circle radius. The starting point is represented by $D(k \sin \psi, k \cos \psi)$, where $-\pi \leq \psi \leq \pi$ and $k > 0$, similarly to Fig. 5.3. Let $\theta_1$ and $\theta_2$ denote the angles of $(\mathbf{p}_1 - \mathbf{x}_1)$ and $(\mathbf{p}_2 - \mathbf{x}_2)$ with respect to the $x$-axis defined in counterclockwise and clockwise, respectively. Then, $\mathbf{p}_1$ is located at $E(-p + r \cos \theta_1, r \sin \theta_1)$, and $\mathbf{p}_2$ is located at $F(p - r \cos \theta_1, r \sin \theta_1)$, and $0 \leq \theta_1, \theta_2 \leq 2\pi$.

**Proof of Proposition 3**

The proposition 3 is proven by showing that there does not exist a TSPN tour that corresponds to the light ray path that travels from $\mathbf{p}_0$, reflects at $\mathbf{p}_1 \in C_1$ and $\mathbf{p}_2 \in C_2$ such that $\mathbf{p}_1, \mathbf{p}_2 \notin C_1 \cap C_2$, and then travels back to $\mathbf{p}_0$. Assume that $\mathbf{p}_1, \mathbf{p}_2 \notin C_1 \cap C_2$. Then, there exist $\theta_1, \theta_2 \in (\frac{\varphi}{2}, \frac{\pi}{2})$ such that the TSPN tour consisting of $\overline{DE}, \overline{EF}$ and $\overline{FD}$ follows the law of reflection at $E$ and $F$. Without loss of generality, let $\theta_1 \leq \theta_2$. Let $\overrightarrow{EE'}$ and $\overrightarrow{FF'}$ represent the reflected light rays at $E$ and $F$, respectively.

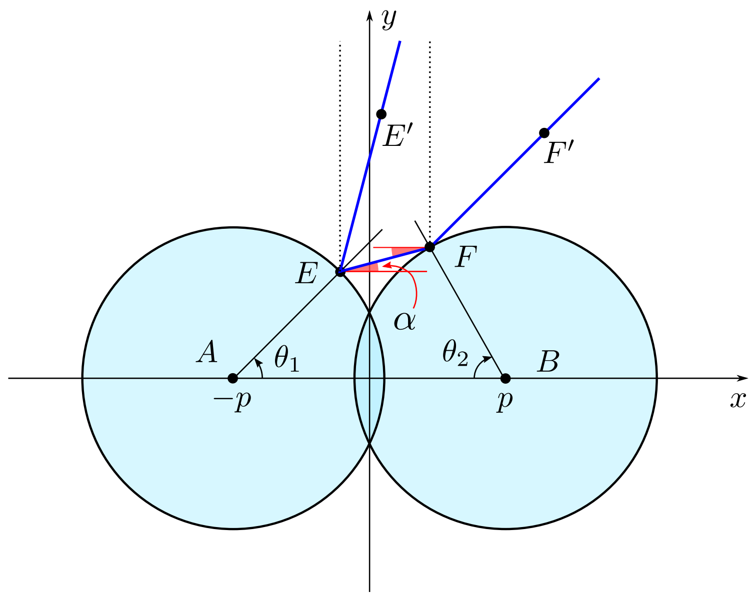



Figure 5.14: TSPN on two intersecting unit circles denoted in Cartesian coordinate for the case in Proposition 3

Because $E(-p + r\cos\theta_1, r\sin\theta_1)$ and $F(p - r\cos\theta_2, r\sin\theta_2)$, the slope of $\overline{EF}$ is

$$\tan\alpha = \frac{2p - r\cos\theta_1 - r\cos\theta_2}{r\sin\theta_2 - r\sin\theta_1} \tag{5.22}$$

and $0 \leq \alpha < \frac{\pi}{2}$ since $\theta_1 \leq \theta_2$. This angle is illustrated by red sector in Fig. 5.14. Then,

$$\angle E'EF = 2(\theta_1 - \alpha) \tag{5.23}$$

$$\angle F'FE = 2(\theta_2 + \alpha) \tag{5.24}$$

by the law of reflection.

The starting point of the shortest TSPN tour is the intersection of two rays $\overrightarrow{EE'}$ and $\overrightarrow{FF'}$. Therefore,

$$\angle E'EF + \angle F'FE < \pi \tag{5.25}$$

by triangle postulate. Equation (5.25) is equivalent to

$$\theta_1 + \theta_2 < \frac{\pi}{2} \tag{5.26}$$

However, by the assumption at the beginning, $\frac{\varphi}{2} < \theta_1 < \frac{\pi}{2}$ and $\frac{\varphi}{2} < \theta_2 < \frac{\pi}{2}$. Thus,

$$\frac{\pi}{2} < \varphi < \theta_1 + \theta_2 \tag{5.27}$$

Thus, by contradiction in equations (5.26) and (5.27), we prove that $\mathbf{p}_1, \mathbf{p}_2 \in C_1 \cap C_2$. Then it is obvious that the shortest TSPN tour satisfies $\mathbf{p}_1 = \mathbf{p}_2 \in \partial C_1 \cap \partial C_2$.

$\square$

**Proof of Proposition 4**

Proposition 4 is proven by showing that the waypoints of the shortest TSPN tour, $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$, are collinear with $\mathbf{x}_1$ or $\mathbf{x}_2$. Let

$$f(\theta_1, \theta_2) = \overline{DE} + \overline{EF} + \overline{FD} \tag{5.28}$$

denote the length of one TSPN tour, not necessarily the shortest. Then,

$$\overline{DE} \geq \|\mathbf{x}_1 - \mathbf{p}_0\| - r \tag{5.29}$$

and the equality holds when $\mathbf{x}_1$, $\mathbf{p}_0$, and $\mathbf{p}_1$ are collinear. From triangle inequality,

$$\overline{DE} \leq \overline{EF} + \overline{FD} \tag{5.30}$$

and the equality holds when $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$ are collinear. Therefore,

$$f(\theta_1, \theta_2) \geq 2\overline{DE} \geq 2(\|\mathbf{x}_1 - \mathbf{p}_0\| - r) \tag{5.31}$$

The equality holds when $\mathbf{x}_1$, $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$ are collinear. Therefore, the shortest TSPN tour satisfies that $\mathbf{x}_1$, $\mathbf{p}_0$, $\mathbf{p}_1$, and $\mathbf{p}_2$ are collinear.

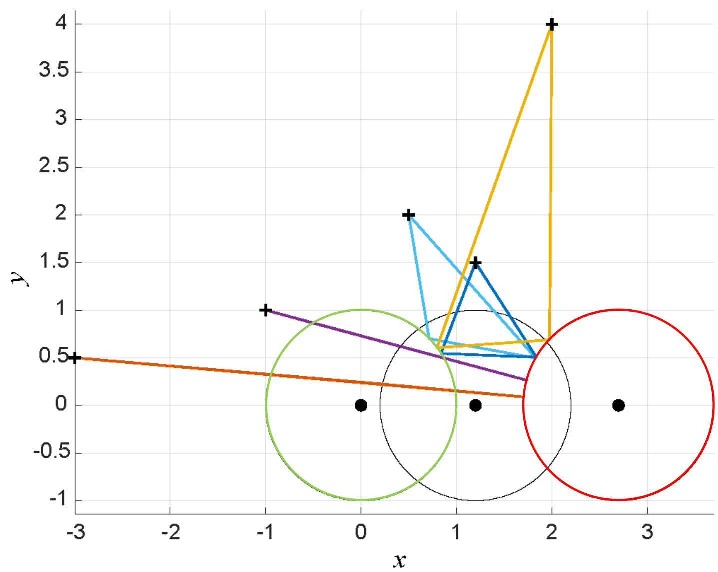
$\square$

Note that the proof of Proposition 6 is omitted as the proof is identical to the one of Proposition 4. The proof of Proposition 5 is omitted as the proposition corresponds to the Fermat's principle.

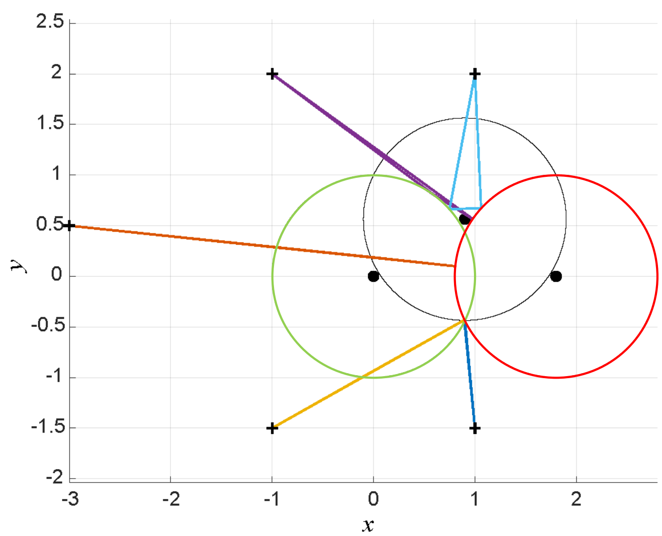
## 5.3 Physics-Inspired Solution: LKH-Geometric Algorithm

The theoretical results in this section are inspired by the observation that the shortest TSPN tour resembles the light ray path that, traveling from a starting point, reflects on circular mirrors corresponding to the TSPN unit circles and, then, travels back to the starting point. Figure 5.15(a) includes some illustrative examples of exact TSPN solutions with starting points at $(-3, 0.5)$ and $(-1, 1)$ corresponds to the light rays reflected by a circular mirror corresponding to the unit circle centered at $(2.7, 0)$ (red circle). The other exact TSPN solutions in Fig. 5.15(a) corresponds to the light rays that is reflected by two circles centered at $(2.7, 0)$ and $(0, 0)$ (red and green circles). Note that the exact solution that resembles a light ray path is reflected to a subset of the given circles as circular mirrors, while the light ray passes through the other circles. It is also observed that, when such a light ray path does not exist, the exact TSPN solution visits at least one intersection of the given circles' boundaries. The examples to support this conjecture are shown in Figures 5.9 and 5.15(b).

Unfortunately, computing a light ray path that starts from a given point, re-




(a)




(b)

Figure 5.15: Exact TSPN solutions from various initial points (denoted by black crosses) are shown in two different circle configurations

flects to some circular mirrors, and then travels back to the exact starting point is computationally intractable due to the characteristics of chaotic scattering [72]. Therefore, this dissertation develops a computational geometry approach that computes a light like path by means of coverage cones proven to contain the line transversals of family of intersecting circles. In the context of this dissertation, a coverage cone can be interpreted as a set of light ray from its apex, and the intersection of a light ray and circle's boundary can be regarded as the point of reflection. While this coverage concept can find the exact position inside each given circle that the TSPN visits, the order of visiting circles can be found using existing TSP algorithms. In this dissertation, LKH algorithm [52] is used to compute the TSP tour of visiting each circle center from the given starting point because of its robustness and performance on computing (near-)optimal solution.

### 5.3.1 $k$-Coverage Cone

A coverage cone is defined for each circular neighborhood with respect to a given apex such that any line that is a subset of the coverage cone transverses the neighborhood [4]. When multiple non-disjoint circles exist, the lines that traverse their intersections can be proven to lie inside a so-called $k$-coverage cone obtained by simple manipulations of unit vectors [4, 31]. When multiple Given an apex $\mathbf{p_0} \in \mathbb{R}^2$, let $K(C_i, \mathbf{p}_0)$ denote the coverage for a circle $C_i$ centered at $\mathbf{x}_i \in \mathbb{R}^2$ with a radius $r_i = 1$, for $i = 1, ..., n$. The position of the center of the Circle $C_i$ relative to the apex $\mathbf{p}_0$ can be expressed by

$$\mathbf{v}_i = \mathbf{x}_i - \mathbf{p}_0 \tag{5.32}$$

as illustrated in Fig. 5.16. Let $\theta_i$ denote the half of the apex angle of $K(C_i, \mathbf{p}_0)$.

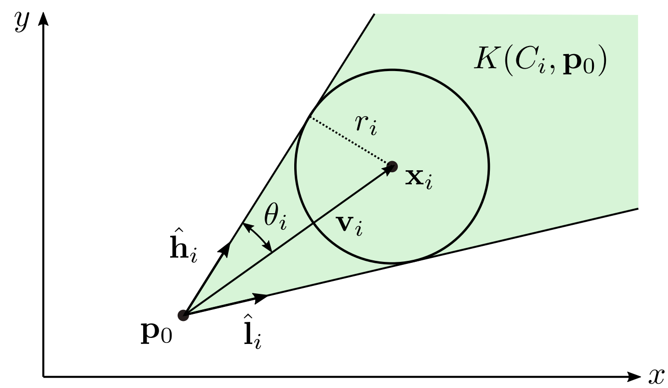



Figure 5.16: Definition of the coverage cone for a given circle $C_i$ centered at $\mathbf{x}_i$ and an apex $\mathbf{p}_0$

Based on the trigonometric relationships,

$$\sin\theta_i = \frac{r_i}{\|\mathbf{v}_i\|} \tag{5.33}$$

$$\cos\theta_i = \frac{\sqrt{\|\mathbf{v}_i\|^2 - r_i^2}}{\|\mathbf{v}_i\|} \tag{5.34}$$

The two unit vectors that define the boundary of $K(C_i, \mathbf{p}_0)$, denoted by $\hat{\mathbf{h}}_i$ and $\hat{\mathbf{l}}_i$, can be expressed using $\mathbf{v}_i$ and $\theta_i$

$$\hat{\mathbf{h}}_i = \begin{bmatrix} \cos\lambda_i \\ \sin\lambda_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \tag{5.35}$$

$$\hat{\mathbf{l}}_i = \begin{bmatrix} \cos\gamma_i \\ \sin\gamma_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \tag{5.36}$$

Therefore, the coverage cone of $C_i$ with respect to the apex $\mathbf{p}_0$ can be defined as

$$\begin{aligned} K(C_i, \mathbf{p}_0) &= \mathrm{cone}(\hat{\mathbf{l}}_i, \hat{\mathbf{h}}_i) \\ &= \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = c_1\hat{\mathbf{l}}_i + c_2\hat{\mathbf{h}}_i,\ c_1, c_2 \geq 0\} \end{aligned} \tag{5.37}$$

For $k$ circles, namely $\mathcal{S} = \{C_1, ..., C_k\}$, if there exists a region such that all coverage cones intersect, i.e., $\cap_{i=1}^{k} K(C_i, \mathbf{p}_0) \neq \emptyset$, the $k$-coverage cone can be defined

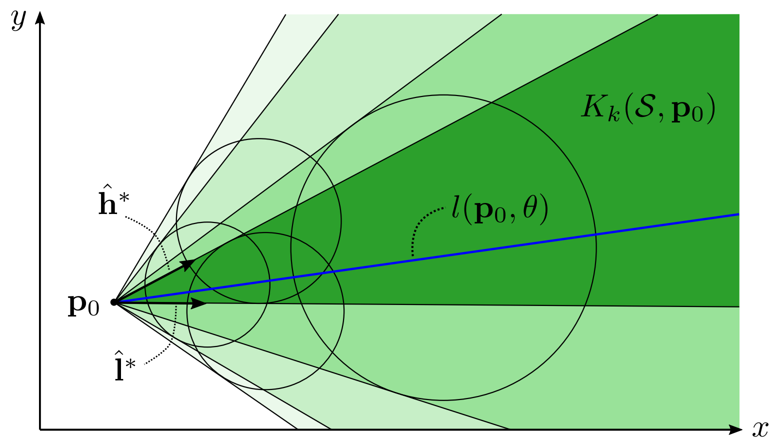



Figure 5.17: A k-coverage cone ($k = 4$) of four circles and an apex $\mathbf{p}_0$

such that any line in the $k$-coverage cone transverses all $k$ circles in $\mathcal{S}$. The $k$-coverage cone can be defined as *ordering* vectors. Taken from [4], the order of two vectors, $\mathbf{u}_i \in \mathbb{R}^2$ and $\mathbf{u}_j \in \mathbb{R}^2$, can be defined as follows. The two vectors are said to be ordered according to the $xy$-frame (denoted by $\mathbf{u}_i \preceq \mathbf{u}_j$) when these vectors are translated to make their origins coincide and $\mathbf{u}_i$ is rotated through the smallest angle possible to meet $\mathbf{u}_j$, where this rotation is in the same direction as the orientation of the $xy$-frame. Based on this notation, let us define two vectors, $\hat{\mathbf{h}}^* = \hat{\mathbf{h}}_\imath$ and $\hat{\mathbf{l}}^* = \hat{\mathbf{l}}_\jmath$, with $\imath, \jmath \in \{1, ..., k\}$ such that $\hat{\mathbf{h}}_\imath \preceq \hat{\mathbf{h}}_i$ and $\hat{\mathbf{l}}_\jmath \succeq \hat{\mathbf{l}}_i$ for $\forall i \in \{1, ..., k\}$. Then, the $k$-coverage cone for $\mathcal{S} = \{C_1, ..., C_k\}$ exists if and only if $\hat{\mathbf{l}}^* \preceq \hat{\mathbf{h}}^*$. Provided that $\hat{\mathbf{l}}^* \preceq \hat{\mathbf{h}}^*$, the $k$-coverage cone for a set of neighborhoods $\mathcal{S}$ with an apex $\mathbf{p}_0$ is defined as

$$K_k(\mathcal{S}, \mathbf{p}_0) = \mathrm{cone}(\hat{\mathbf{l}}^*, \hat{\mathbf{h}}^*) \tag{5.38}$$

as illustrated in Fig. 5.17. Any line in the $k$-coverage cone is a line transversal to the $k$ neighborhoods and defined as follows. Defining a unit vector

$$\hat{\mathbf{t}}(\theta) = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{5.39}$$

a line transversal in $K_k(\mathcal{S}, \mathbf{p}_0)$ is defined as

$$l(\mathbf{p}_0, \theta) = \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{x} = \mathbf{p}_0 + d\hat{\mathbf{t}}(\theta),\ d \geq 0\} \tag{5.40}$$

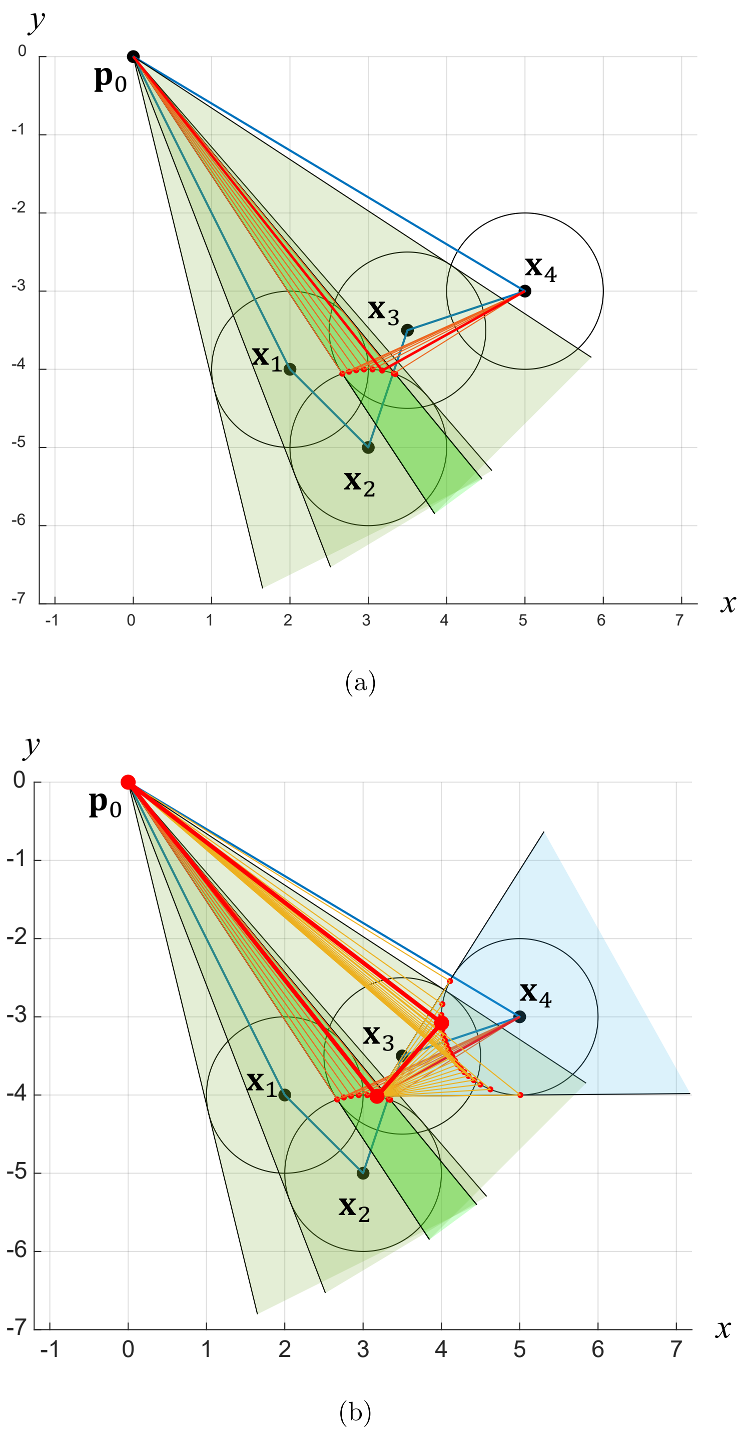
where $\theta \in (\gamma^*, \lambda^*)$, such that $\hat{\mathbf{l}}^* = [\cos\lambda^* \ \sin\lambda^*]^T$ and $\hat{\mathbf{h}}^* = [\cos\gamma^* \ \sin\gamma^*]^T$.

### 5.3.2 LKH-Geometric Algorithm

This dissertation proposes a novel solution to TSPN on uniform circles, namely LKH-Geometric, which computes the shortest TSPN tour using LKH algorithm and $k$-coverage cone. In a nutshell, the proposed LKH-Geometric finds the order of circles to visit using the existing LKH algorithm, and then, computes the exact waypoint inside each circle using $k$-coverage cone. The detailed description on the LKH-Geometric follows in this subsection.

As a first step, the LKH algorithm is used to compute a TSP tour of visiting all circle centers, $\mathbf{x}_1, ..., \mathbf{x}_n$, from the given starting point $\mathbf{p}_0$. The computed sequence of circles to visit is denoted by $\Sigma_{LKH} = (\sigma_1, ..., \sigma_n)$, where $\sigma_i \in \{1, ..., n\}$, $\sigma_i \neq \sigma_j$ for $i \neq j$, and $\mathbf{x}_{\sigma_i} \in \{\mathbf{x}_1, ..., \mathbf{x}_n\}$. In Fig. 5.18(a), for the given four circles centered at $\mathbf{x}_1, ..., \mathbf{x}_4$, the TSP tour computed by LKH algorithm is illustrated in blue color, and the sequence is $\Sigma_{LKH} = (1, 2, 3, 4)$.

After computing the sequence of visiting circles, $\Sigma_{LKH}$, the exact waypoint inside each circle is computed using $k$-coverage cone. The resulting TSPN tour is represented by a sequence of waypoints denoted by $P = (\mathbf{p}_0, \mathbf{p}_{\sigma_1}, ..., \mathbf{p}_{\sigma_n}, \mathbf{p}_0)$. Note that it is possible to have $\mathbf{p}_{\sigma_i} = \mathbf{p}_{\sigma_j}$ for some $\sigma_i$ and $\sigma_j$ when that waypoint lets the TSPN tour to visit both $C_{\sigma_i}$ and $C_{\sigma_j}$. After initializing the sequence of waypoints by $P = (\mathbf{p}_0)$, the LKH-Geometric repeats the following steps until all the circles are visited.




(a)

(b)

Figure 5.18: Diagrams illustrating detailed steps in LKH-Geometric algorithm: blue line indicates TSP tour computed by LKH; shaded cones illustrate $k$-coverage cone computation; thick red line and dots denote TSPN tour and chosen waypoints, respectively

First, from the current waypoint $\mathbf{p}_{\sigma_i}$, $k$-coverage cone is computed for $\{C_{\sigma_{i+1}}, ..., C_{\sigma_{i+k}}\}$ with the maximum $k$ value such that $k$-coverage cone exists. This $k$-coverage cone is denoted by $K_k(\{C_{\sigma_{i+1}}, ..., C_{\sigma_{i+k}}\}, \mathbf{p}_{\sigma_i})$ following the notation (5.38). In Fig. 5.18(a), the $k$-coverage cone is constructed with $k = 3$ for $\{C_1, C_2, C_3\}$ from the current waypoint $\mathbf{p}_0$. In Fig. 5.18(b), the $k$-coverage cone is constructed with $k = 1$ for $\{C_4\}$ from the current waypoint $\mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_3$ on $\partial C_2$.

Next, Once the $k$-coverage is computed, a line transversal defined in (5.40) is used to compute the next waypoint. A line transversal $l(\mathbf{p}_{\sigma_i}, \theta)$ in $K_k(\{C_{\sigma_{i+1}}, ..., C_{\sigma_{i+k}}\}, \mathbf{p}_{\sigma_i})$ is chosen such that $|\mathbf{p}_{\sigma_i} - \mathbf{p}^*| + |\mathbf{p}^* - \mathbf{x}_{\sigma_{i+k+1}}|$ is minimized, where $\mathbf{p}^*$ is a point on $l(\mathbf{p}_{\sigma_i}, \theta)$ such that a line segment from $\mathbf{p}_{\sigma_i}$ to $\mathbf{p}^*$ is the minimum-length segment to visit all the circles in $\{C_{\sigma_{i+1}}, ..., C_{\sigma_{i+k}}\}$. Following the definition of $P$, $\mathbf{x}_{\sigma_{n+1}}$ is considered as $\mathbf{p}_0$. In Fig. 5.18(a), several line transversals are illustrated in thin ornage lines, and the one with $\mathbf{p}^*$ is illustrated in a thicker red line. This step of finding the next waypoint is inspired by physics, although the resultant TSPN tour is not identical to the light ray path. Then, the next waypoints become $\mathbf{p}_{\sigma_{i+1}} = ... = \mathbf{p}_{\sigma_{i+k}} = \mathbf{p}^*$. The description of LKH-Geometric is summarized in Algorithm 2.

## 5.4 Numerical Experiment Results on LKH-Geometric

The LKH-Geometric algorithm is tested for comparison on the benchmark dataset from TSPLIB [67], which a library of TSP instances. The TSPN instances are generated based on the dataset from TSPLIB, by considering the given TSP points as circle centers and choosing a uniform radius based on the overlap ratio similarly to [28]. A starting point is chosen from the given circle centers in each TSP instance

Algorithm 2: LKH-Geometric

**Require:** circle centers $\mathbf{x}_1, ..., \mathbf{x}_n$, uniform radius $r$, starting point $\mathbf{p}_0$

**Ensure:** a sequence of waypoints consisting a TSPN tour $P$

(1) Compute a sequence of a TSP tour $\Sigma_{LKH} = (\sigma_1, ..., \sigma_n)$ that visits the circle centers $\mathbf{x}_1, ..., \mathbf{x}_n$ from $\mathbf{p}_0$

(2) Initialize $P \leftarrow \mathbf{p}_0$ and $i \leftarrow 0$; Set $\mathbf{p}_{\sigma_0} = \mathbf{p}_{\sigma_{n+1}} = \mathbf{p}_0$

**while** $i < n$ **do**

(3-a) Find a $k$-coverage cone for circles $C_{\sigma_{i+1}}, ..., C_{\sigma_{i+k}}$ with maximum $k$ such that $K_k(\{C_{\sigma_{i+1}}, ..., C_{\sigma_{i+k}}\}, \mathbf{p}_{\sigma_i})$ exists

(3-b) Find a waypoint $\mathbf{p}^*$ on a line transversal $l(\mathbf{p}_{\sigma_i}, \theta)$ inside $K_k(\{C_{\sigma_{i+1}}, ..., C_{\sigma_{i+k}}\}, \mathbf{p}_{\sigma_i})$ such that $|\mathbf{p}_{\sigma_i} - \mathbf{p}^*| + |\mathbf{p}^* - \mathbf{x}_{\sigma_{i+k+1}}|$ is minimized with respect to $\theta$.

(3-c) $P \leftarrow \mathbf{p}_{\sigma_{i+1}}, ..., \mathbf{p}_{\sigma_{i+k}}$, where $\mathbf{p}_{\sigma_{i+1}} = ... = \mathbf{p}_{\sigma_{i+k}} = \mathbf{p}^*$; $i \leftarrow i + k$

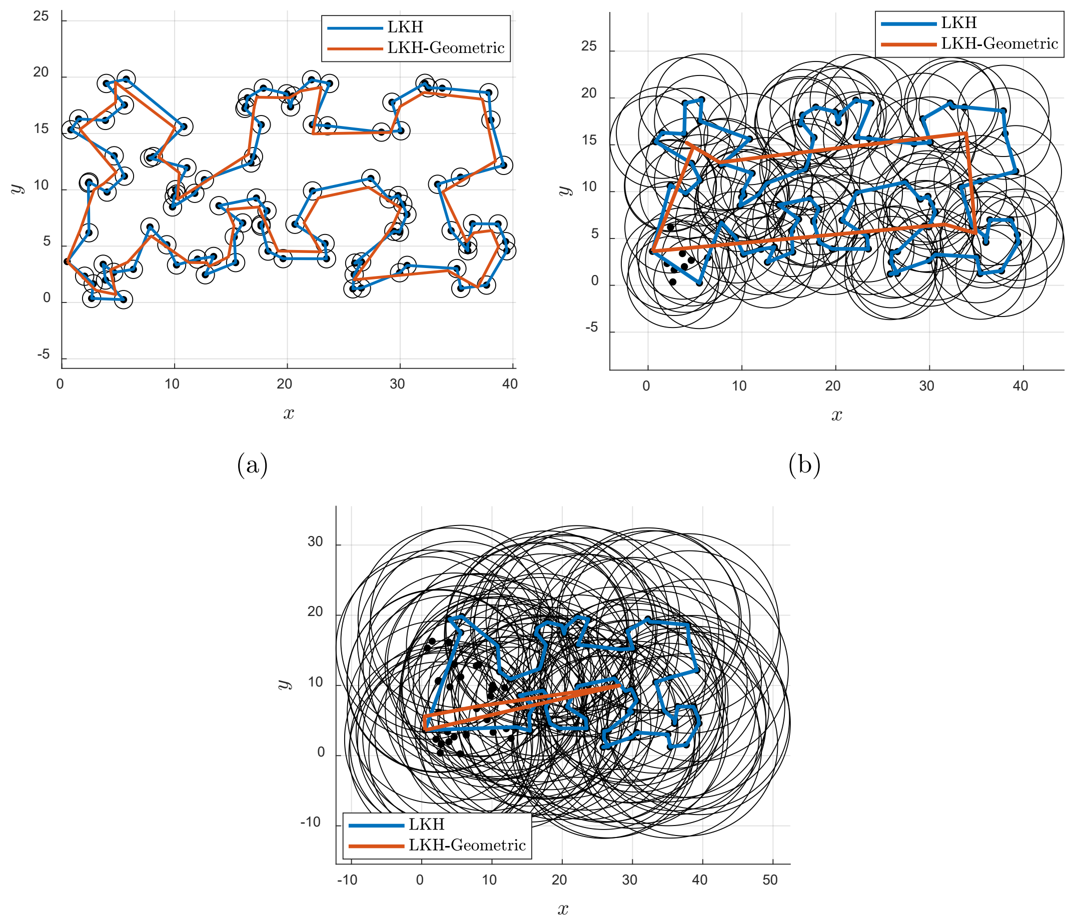
**end while**

**return** $P$

and defined as a circle with a zero radius.

The *overlap ratio*, denoted by $R$, is defined as the ratio of the uniform radius to the length of the smallest square containing all $n$ disks [55]. A set of circles with a smaller overlap ratio tends to have less intersection of circles compared to the one with a larger overlap ratio. In Fig. 5.19, the TSPN solution computed by LKH-Geometric is compared with the TSP tour computed by LKH. When the overlap ratio is small, $R = 0.02$ (Fig. 5.19(a)), the TSPN solution doest not deviate significantly from the TSP solution computed for circle centers, because there is not significant room to choose a waypoint that can visit multiple circles at once. However, as the overlap ratio increases to $R = 0.1$ and $R = 0.3$ (Figures

5.19(b) and 5.19(c)), the TSPN solution from LKH-Geometric reduces the tour length from TSP solution from LKH by visitng multiple circles though a single pass, which is computed from a transversal inside $k$-coverage cone.




(a) (b)

(c)

Figure 5.19: The TSPN solution from LKH-Geometric and TSP solution from LKH algorithms for the dataset 'kroD100' with (a) small overlap ratio $R = 0.02$ (b) moderate overlap ratio $R = 0.1$ and (c) high overlap ratio $R = 0.3$

The performance of the proposed LKH-Geometric algorithm, in terms of TSP tour length, is analyzed by comparing the length of the TSP tour that visits all of the circle centers obtained by LKH. Since the TSP tour computed by LKH is used as a reference, its length is denoted by $L_{ref}$. Let $L$ denote the length of the TSPN tour computed by LKH-Geometric approach. Then, by $L$, the percent improvement of LKH-Geometric over LKH is given by, can be defined as

$$\gamma = \frac{(L_{ref} - L)}{L_{ref}} \times 100[\%] \tag{5.41}$$

The percent improvement is computed for three overlap ratios using the datasets 'kroD100', 'rat195', 'lin318', 'rd400', 'pcb442', 'd493', in which the total number of circles is equal to $n = 100$, $n = 195$, $n = 318$, $n = 400$, $n = 442$, $n = 493$, respectively. For every test case consisting of different dataset from TSPLIB and overlap ratio, the improvement was computed by taking average of 100 simulation results. However, the standard deviation was not plotted because of the low order of mangitude, which was smaller than $10^{-15}$. The standard deviation is small due to the high robustness of LKH algorithm, because the steps (2) to (3-c) in Algorithm 2, which computes the waypoints using $k$-coverage cone is deterministic.

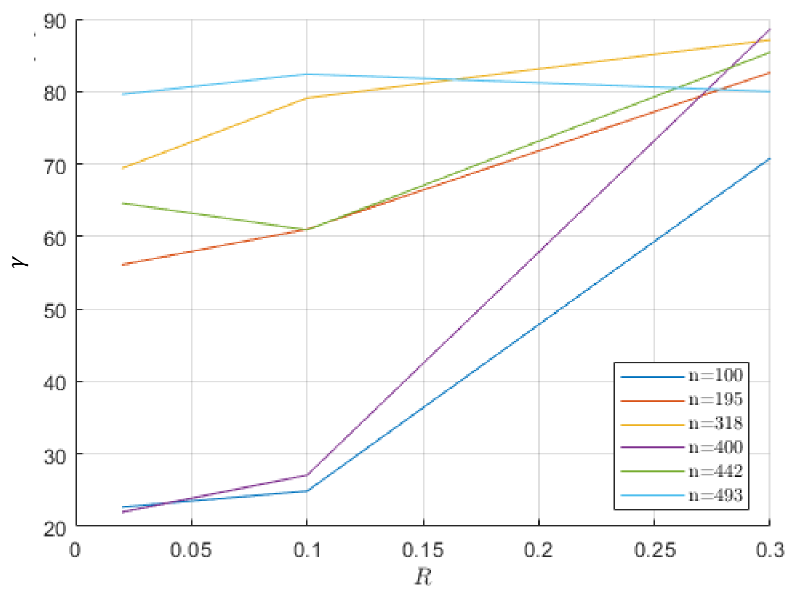



Figure 5.20: The percent improvement $\gamma$ of LKH-Geometric over LKH plotted for different overlap ratio, $R$, and number of neighborhoods, $n$.

In Fig. 5.20, the percent improvement of LKH-Geometric over LKH tends to increase as the overlap ratio increases. This is because, as illustrated in Fig. 5.19, LKH-Geometric can choose waypoints such taht the TSPN tour can visit multiple circles via single pass. The percent improvement is less dependent on the number of circles, because LKH-Geometric chooses the waypoints using $k$-coverage cone, which varies depending on the circle's configuration, especially on how much the circles overlap.

Finally, the computation time afforded by the LKH-Geometric algorithm is compared with the state-of-the-art solver known as Growing Self-Organizing Array (GSOA) [28]. The computational complexity of GSOA is bounded by $O(n^2)$, and the time complexity of LKH is also $O(n^2)$, as also illustrated in Fig. 5.21 [28, 39]. The steps (2) to (3-c) in Algorithm 2 are characterized by a time complexity of order of $O(n \log n)$, because the coverage cone has to be sorted in waypoint planning. Although the dominant term is $O(n^2)$, the total computation time of LKH-Geometric remains lower than that of GSOA. The advantage of the LKH-Geometric over GSOA is that, by using the coverage cones, a TSPN solution similar to GSOA can be obtained without the use of any prior data, also is required by TSPN methods based on unsupervised learning or evolutionary algorithms.

In future work, the LKH-Geometric solution will be further improved and, then, extended to non-uniform circular neighborhoods that are either equal to or are contained by the neighborhoods. By solving the TSPN path planning formulation in Problem 2, the most efficient directional sensor path solution to the IMVP Problem 2 is found. The effectiveness of the IMVP solutions compared to state-of-the-art planners is demonstrated in the next chapter using both simulated and real UUVs engaged in underwater target classification.

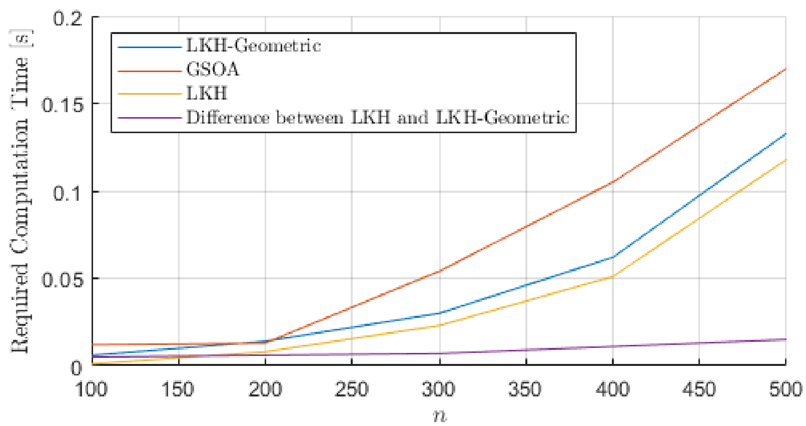



Figure 5.21: Computation time complexity of LKH-Geometric compared with GSOA and LKH.

# CHAPTER 6
## SIMULATION RESULTS AND SEA TEST DEMONSTRATION

The novel IMVP approach presented in this dissertation is first demonstrated on an integrated physics-based simulation dataset of a UUV-based side-scan sonar developed by Dr. Isaacs at NSWC. In this simulation, the dynamics of the UUV are modeled based on the REMUS 100 autonomous underwater vehicle using six degrees-of-freedom nonlinear equations of motion [33]. A pair of side-scan sonar sensors mounted on the UUV are simulated by generating images obtained from the sensor FOV by beamforming the time domain signals using $\omega - k$ beamforming [41]. Other beamforming techniques, such as time-delay and chirp scaling, can also potentially be utilized [37]. As can be seen in Fig. 6.2, objects of interest exhibit strong highlights with varying shadows depths that, while not necessarily unique to objects of interest, provide information about the object features and class. After the image is generated by traveling along a straight line, $l = 3$ m, the ATR algorithm described in Chapter 3, taken from [85], is used to classify objects that have been detected in sonar imagery and to distinguish them from clutter and sea-floor ripples.

The simulated sonar FOV geometry is characterized by the minimum and maximum ranges $r_{min} = 15$ m and $r_{max} = 150$ m, respectively. Once the UUV trajectory is planned by the IMVP approach, the UUV motion is controlled by a proportional-integral-derivative (PID) controller that determines the UUV stern angle, rudder angle, and propeller revolution per minute (RPM) for accurate trajectory following. For simplicity, in this dissertation it is assumed that the UUV position and the target information are provided relative to the inertial frame $\mathcal{F}_{\mathcal{W}}$, inside an ROI $\mathcal{W} = [-L, L] \times [-L, L] \times [0, H]$, where $L = 1200$ m and $H = 50$

Table 6.1: Integrated UUV-based sonar simulation variables and respective ranges.

| Symbol | Nodes | Domain |
|---|---|---|
| $Y_i$ | Target classification | $\mathcal{Y} = \{0, 1\}$ |
| $X_{i1}$ | Target shape feature | $\mathcal{X}_1 = \{\text{sphere}, \text{cylinder}\}$ |
| $X_{i2}$ | Target volume feature defined as the cube root of target volume [m] | $\mathcal{X}_2 = \{[0, 0.16), [0.16, 0.30), [0.30, 1.1), [1.1, 1.7]\}$ |
| $\mathbf{q}_{T_i}$ | Target state | $\mathbb{R}^2 \times \mathbb{S}^1$ |
| $\mathbf{q}$ | UUV configuration | $\mathcal{C}$ |
| $\Lambda_i$ | Relative UUV configuration at time with respect to the $i$th target | $\mathcal{V}_i$ |
| $\hat{Y}_i$ | Estimated target classification | $\mathcal{Y}$ |
| $\hat{X}_{i1}$ | Estimated target shape features | $\mathcal{X}_1$ |
| $\hat{X}_{i2}$ | Estimated target volume feature | $\mathcal{X}_2$ |

m. A target field is generated by sampling a database of underwater objects with the characteristics summarized in Table 6.1 and by distributing the objects in the ROI randomly and uniformly, or in random clusters that replicate real-world object fields. Each underwater object may be classified as a target of interest (TOI), $y_i = 1$, or clutter, $y_i = 0$, based on its features. As shown in Table 6.1, target features available in the sonar simulation are shape ($X_{i1}$) and volume ($X_{i2}$), i.e., $X_i = \{X_{i1}, X_{i2}\}$.

From the target features estimated from the sonar imagery, denoted by $Z_i = \{\hat{X}_{i1}, \hat{X}_{i2}\}$, the class of the $i^{th}$ object, $\hat{Y}_i$, is inferred using the measurement model provided by the joint PMF in Section 2, learned from a training database of 260 objects using the ATR approach in [85]. A different database comprised of 260 objects, randomly sampled from the simulation database and not included in the training database, is then used to generate the target fields for the simulated ROI and, subsequently, for testing the path planning algorithms presented in this dissertation. Three classification sets of increasing difficulty are used in this dissertation. The first classification set, labeled as Set A, contains objects that can be classified as TOIs based solely on their volume. The second classification set, labeled as Set B, contains objects that can be classified as TOIs based on both their volume and their shape. The third classification set, labeled as Set C, consists of the same objects as Set B but is characterized by harsher simulated environmental conditions.

The IMVP solutions are first demonstrated using two high-fidelity simulation environments and, then, tested on a real UUV swimming in the Saint Andrew Bay area in Panama City, FL, as explained in the following sections.

## 6.1 IMVP Results

The IMVP approach developed in this dissertation is tested on a variety of target fields and compared to the state-of-the-art multi-view planning methods known as multiple aspect coverage (MAC) and clustered MAC (CMAC) [6–8, 61, 70]. Because the objects' locations and features used for classification all influence the UUV-based sensor performance, the IMVP approach is demonstrated first by considering different object layouts (Section 6.1.1) and, then, different classification sets (Section 6.1.2) using the simulation environment described in Section 6. The computational complexity of the IMVP solution algorithm proposed in Section 4.3 is analyzed in Section 4.4.

The IMVP performance is evaluated based on the following metrics: (*i*) the travel time ($T$) required to classify all targets with a minimum confidence level $\varepsilon_{CL}$; (*ii*) the total number of contacts ($N$) per travel time; and, (*iii*) the average confidence level of the targets of interest classified along the path. Unlike existing multi-view planning methods, which take into account only the location of the targets, IMVP seeks to minimize the travel time and images processed by the sensor by selecting only the most informative views. In order to demonstrate that the desired classification confidence is met by the IMVP planner, the actual CL of all TOIs ($y_i = 1$) in the ROI is evaluated by the ATR algorithm (Chapter 3) and, then, averaged obtaining the following performance metric

$$\bar{c}_T = \frac{1}{n'} \sum_{\{i \ | \ y_i = 1\}} c(Y_i; M_i(K)) \tag{6.1}$$

Because the CL threshold ($\varepsilon_{CL}$) is only required for TOIs, the average is taken over the total number of TOIs in the region ($n'$), not including clutter.

The IMVP classification performance is also evaluated by assessing classification accuracy (CA), false alarm ratio (FA), and missed detection ratio (MD). CA, also known as true positive rate, is defined as the ratio of the number of correctly classified TOIs over total number of TOIs ($n'$). FA or false alarm ratio is defined as the total number of objects incorrectly classified as TOI over the total number of objects ($n$), and the MD or false negative rate is defined as the total number of TOIs incorrectly classified as clutter over $n$. The classification accuracy per travel time, referred to here as *classification efficiency*, is also evaluated and denoted by $\eta = \text{CA}/T$.

In the following subsections, the IMVP performance is demonstrated for a variety of target fields characterized by different layouts (Section 6.1.1) and classification features (Section 6.1.2). In every case study, the IMVP performance is compared to the MAC algorithm, which plans the shortest multi-view path to cover every object using a fixed pre-planned number of aspect angles, such that every object is detected at least once from each aspect angle [6–8, 61]. The MAC path may be inefficient for sparse object layouts, requiring the UUV-based sensor to travel long times without observing any objects [6–8, 61]. The modification proposed in [70], known as CMAC, overcomes this limitation by designing the path based on the size of object clusters that may occur in applications with man-made TOIs [64, 71]. Objects are first grouped in clusters by using density-based spatial clustering of applications with noise (DBSCAN) method and, then, the shortest path between clusters is found, typically reducing travel time compared to MAC solutions.

### 6.1.1 Influence of Object Location on IMVP Performance

Previous multi-view planning studies showed that path performance depends strongly on the object layout [6–8, 61, 70]. In particular, algorithms that perform well for objects uniformly distributed spatially, at random, over the ROI may not perform adequately when objects that are laid out into clusters, and viceversa. In this subsection, the features of the IMVP algorithm are demonstrated using two case studies with relatively small target fields obtained by sampling the same classification Set B to obtain $n = 12$ underwater objects, with $n' = 4$ TOIs. The integrated UUV-based sonar and ATR simulation (Section 6) is then used to evaluate all performance metrics after the UUV's trajectory is executed. The paths computed by the IMVP algorithm are simulated using high-fidelity AVA simulator to demonstrate that the path is executable under the UUV dynamics constraints for the sea test (Section 6.3).

In the first case study, the target field is generated by placing objects in the ROI

Table 6.2: Path Planning and Classification Performance Comparison for a Uniformly Sampled Object Distribution

| Performance Metric | Algorithm | | | Improvement[1] |
|---|---|---|---|---|
| | IMVP | MAC | CMAC | |
| Travel time, $T$ [hr] | 1.56 | 2.50 | 2.49 | **37.4** [%] |
| Contacts per unit time, $N/T$ [min$^{-1}$] | 0.47 | 0.22 | 0.28 | **68** [%] |
| Classification efficiency, $\eta$ [hr$^{-1}$] | 0.48 | 0.20 | 0.29 | **66** [%] |
| Average TOIs CL | 0.98 | 0.97 | 0.98 | 0.11 [%] |

by randomly sampling underwater objects from the classification Set B (Section 6) and, then, by placing them in $\mathcal{W}$ at a position and orientation obtained by sampling a uniform distribution defined over $\mathcal{W} \times \mathbb{S}^1$. A representative example of IMVP sensor path is plotted in Fig. 6.1(a). By leveraging prior sensor measurements, or evidence $E_i(0)$ ($i = 1, \ldots, n$), the IMVP path is able to minimize distance traveled between multiple swaths per target, as well as to decide and plan the number of swaths based on the target examined. When the MAC and CMAC algorithms are applied to the same target field, the resulting paths are as shown in Figs. 6.1(b) and 6.1(c), respectively. It can be seen that these existing algorithms plan the number of swaths *a priori* and equally for all targets, only based on their locations. As a result, the IMVP approach developed in this dissertation significantly reduces the travel time, while achieving the same required CL for the TOIs (Table 6.2) and the same classification performance in terms of CA, MD, and FA. This is because, while reducing the travel time by approximately 37% compared to the best existing algorithm, the IMVP approach uses prior target information to obtain a large number of high-quality object images (Fig 6.2), as demonstrated by the number of contacts and classification accuracy per unit time (Table 6.2).

[1] Percent improvement over best existing algorithm

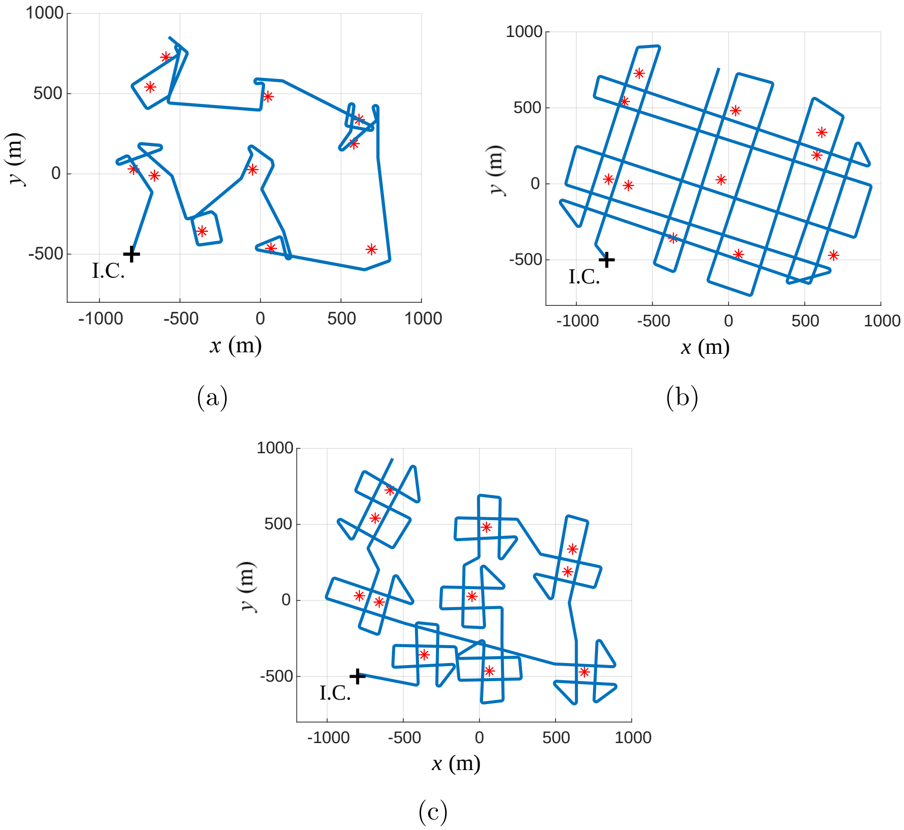
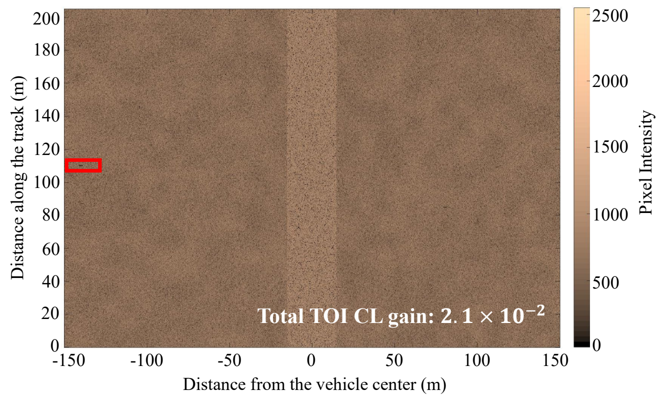
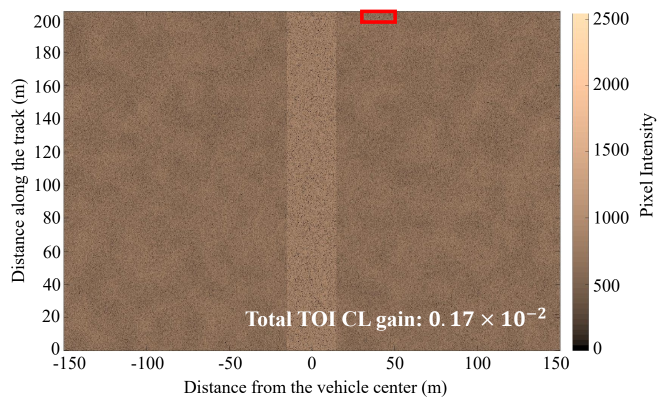



Figure 6.1: Path planning results obtained by the (a) IMVP, (b) MAC, and (c) CMAC algorithms for a representative example of target field with classification features drawn from Set B and object locations sampled from a uniform distribution (red stars), where the initial condition (I.C.) of the UUV-based sonar is denoted by the black cross.

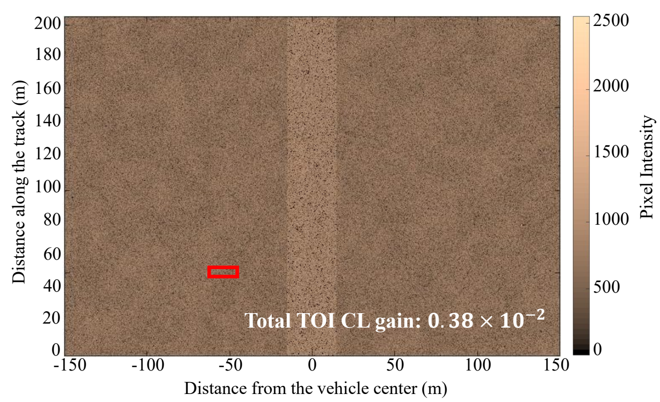
In the second case study, the target field is generated by placing objects in clusters, after randomly sampling underwater objects from the classification Set B (Section 6), using a uniformly sampled object orientation. Object clusters typically present themselves in applications with man-made TOIs [64, 71] and offer the opportunity to view many objects in a single swatch, provided the optimal aspect angle is planned for the UUV-based sonar. A representative target field with 3 clusters, shown in Fig. 6.3, is used to compare the trajectories generated by the IMVP, MAC, and CMAC algorithms. As shown by the performance metrics




(a)




(b)




(c)

Figure 6.2: Sonar images obtained by the (a) IMVP, (b) MAC, and (c) CMAC algorithms around the coordinate $x = -400$ (m) and $y = -350$ (m) from the uniform distribution in Fig. 6.1 and afforded total gain in CL.

Table 6.3: Path Planning and Classification Performance Comparison for a Clustered Object Distribution

| Performance Metric | Algorithm | | | Improvement[2] |
|---|---|---|---|---|
| | IMVP | MAC | CMAC | |
| Travel time, $T$ [hr] | 0.990 | 3.03 | 1.84 | **46.2** [%] |
| Contacts per unit time, $N/T$ [min$^{-1}$] | 0.71 | 0.19 | 0.34 | **110** [%] |
| Classification efficiency, $\eta$ [hr$^{-1}$] | 0.93 | 0.31 | 0.52 | **79** [%] |
| Average TOIs CL | 0.97 | 0.95 | 0.96 | 1.0 [%] |

summarized in Table 6.3, the IMVP algorithm obtain images more efficiently by observing multiple targets through a single pass, thus achieving the required TOI CL in less travel time. The CMAC algorithm also exploits the cluster configuration to reduce travel time. However, the IMVP approach is significantly more effective at planning the sensor path that both enables multiple detections and utilizes the most informative aspect angles, as demonstrated both by the number of object contacts and classification performance per unit time (Table 6.3). This is because the IMVP aspect angles take into consideration the geometry of the C-targets as well as the object features to determine the most informative views and, thus, obtain the most informative sonar images (Fig. 6.4). As evidenced by these two representative case studies, the IMVP approach is similarly able to determine the optimal path for different object configurations (uniformly distributed or clustered) because of the systematic geometric construction of the C-targets obtained from the sonar FOV geometry and object locations.

[2]Percent improvement over best existing algorithm

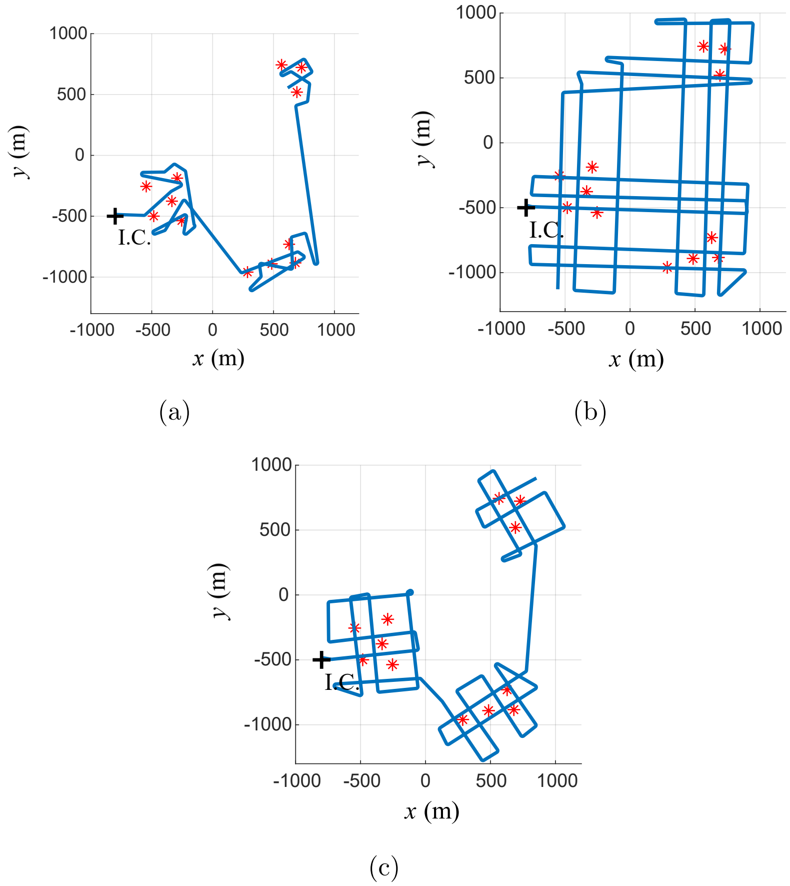
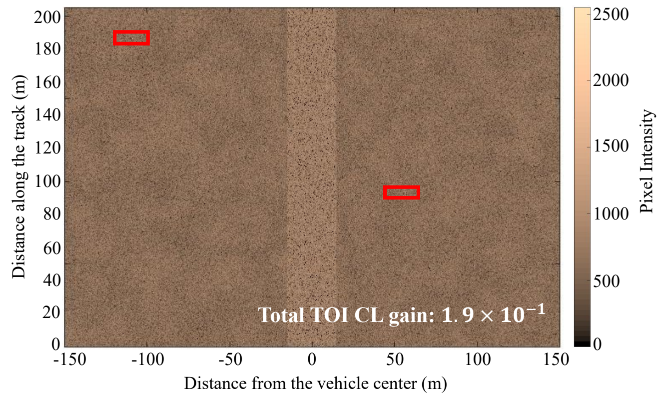


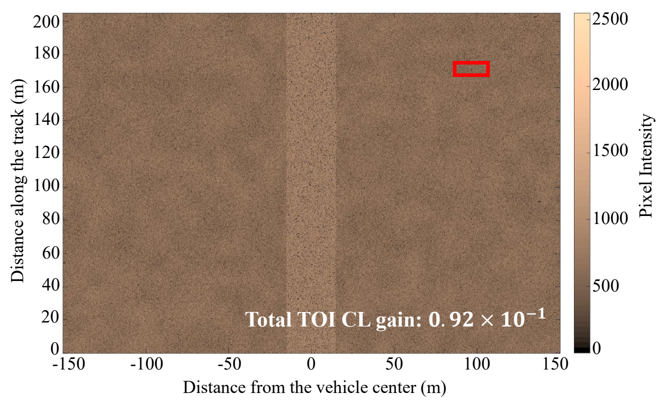

Figure 6.3: Path planning results obtained by the (a) IMVP, (b) MAC, and (c) CMAC algorithms for a representative example of target field with classification features drawn from Set B and clustered object locations (red stars), where the initial condition (I.C.) of the UUV-based sonar is denoted by the black cross.

### 6.1.2 Influence of Classification Features on IMVP Performance

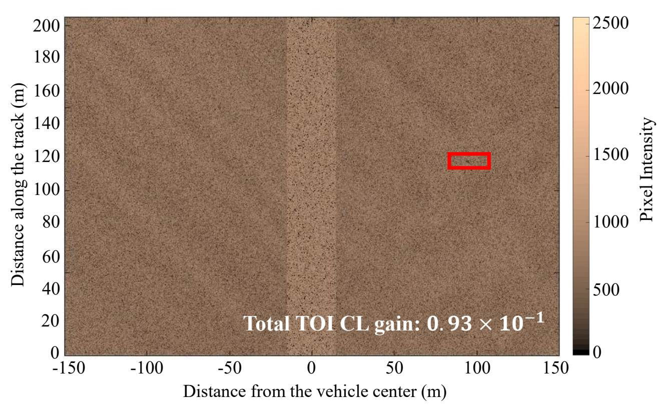
In addition to accounting for the sonar FOV geometry and object location, the IMVP approach also provides a systematic methodology for determining the most valuable views based on prior information about the target features and ATR characteristics. The IMVP ability to adapt the path to the complexity of the




(a)




(b)




(c)

Figure 6.4: Sonar images obtained by the (a) IMVP, (b) MAC, and (c) CMAC algorithms around the coordinate $x = 700$ (m) and $y = -800$(m) from the clustered distribution in Fig. 6.3 and afforded total gain in CL.

classification task is demonstrated by using three object classification databases of increasing complexity, referred to as Sets A, B, and C, described in Section 6. Using the same set of target locations (Fig. 6.5), three target fields are generated by sampling $n = 19$ underwater objects from Sets A, B, and C. The corresponding IMVP trajectories, respectively plotted in Fig. 6.5(a), 6.5(b), 6.5(c), show that the optimal number of views and the shortest path between them highly depend on the target characteristics.

On the other hand, the MAC and CMAC algorithms produce the same identical path for all the three target fields because they only account for the object location (trajectories omitted for brevity). The result is not only a reduced travel time by IMVP but also improved classification efficiency (Table 6.4), particularly for challenging classification features (Set B) and environmental conditions (Set C). This is because the IMVP algorithm determines the minimum number of views and the most informative aspect angles required per object, based on its ECL and estimated features, and then determines the shortest path between them.

Finally, a statistically significant analysis of the performance improvement brought about by the IMVP approach compared to existing algorithms is conducted by generating ten target fields for every classification set (Set A, B, and C). Every classification performance metric is then evaluated by averaging 1,000 trials to obtain both its mean value and standard deviation. In addition to classification efficiency, the actual gain in confidence level per unit time is computed as follows,

$$\beta = \frac{1}{nT} \sum_{i=1}^{n} c(Y_i; M_i(K)) \tag{6.2}$$

to determine how informative are the sonar images obtained by the IMVP ap-

[3] Percent improvement over best existing algorithm

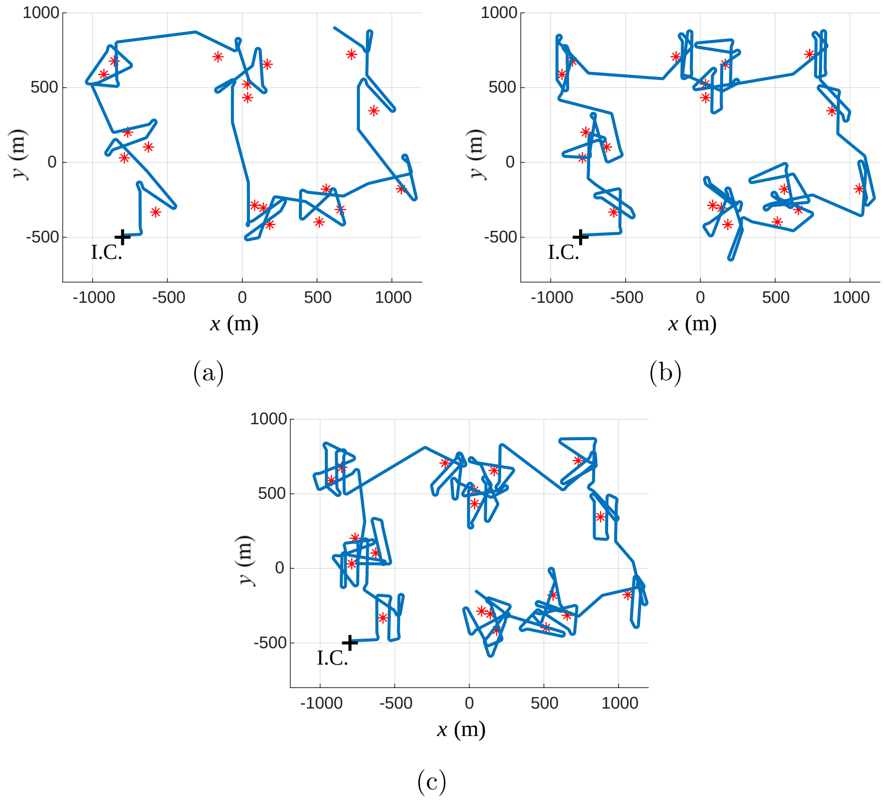



Figure 6.5: IMVP path planning results obtained for a fixed set of object locations (red stars) and different object features sampled from classification Sets A (a), B (b), and C (c), where the initial condition (I.C.) of the UUV-based sonar is denoted by the black cross.

proach. The mean value and standard deviation of the classification efficiency ($\eta$) and of the CL gain per unit time ($\beta$) are plotted in Fig. 6.6 and Fig. 6.7, respectively, for the IMVP approach, as well as for the MAC and CMAC algorithms.

The results in Fig. 6.6 show not only that the IMVP approach achieves a much higher classification efficiency – namely 88% improvement for set A, 49% improvement for set B, and 13% improvement improvement for set C – but also a much smaller standard deviation than that of MAC and CMAC algorithms, indicating that the IMVP performance is not only better but also more robust. Furthermore, the classification results obtained by the IMVP approach also have

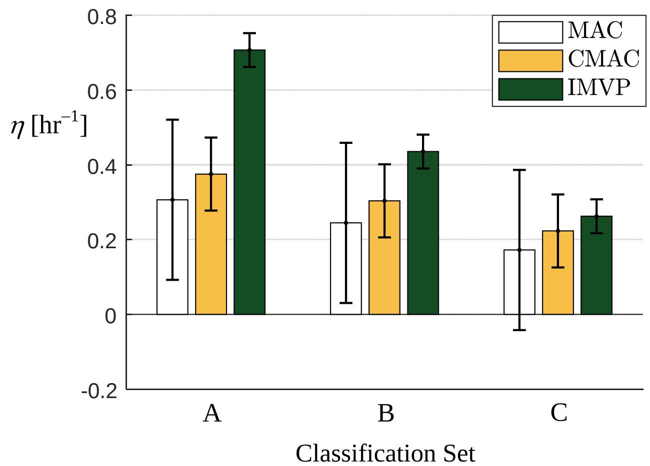



Figure 6.6: Classification efficiency mean value and standard deviation (vertical bars) for IMVP, MAC, and CMAC algorithms.

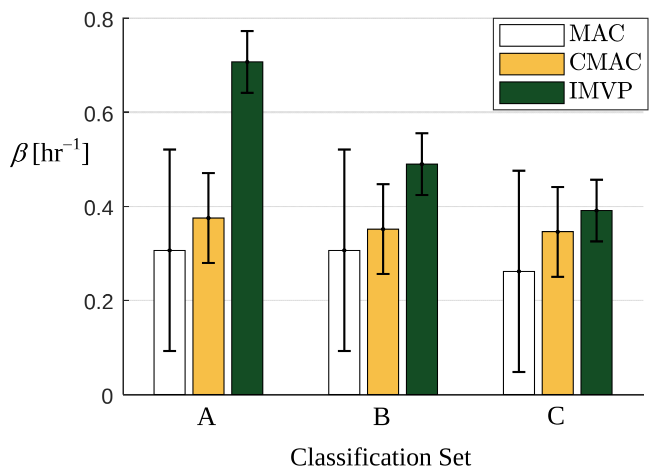



Figure 6.7: Confidence-level gain per unit time and standard deviation (vertical bars) for IMVP, MAC, and CMAC algorithms.

Table 6.4: Path Planning and Classification Performance Comparison for Different Classification Sets

| Performance Metric | IMVP Performance (Improvement[3]) | | |
|---|---|---|---|
| | Set A | Set B | Set C |
| Travel time, $T$ [hr] | 1.67 **(47.7%)** | 2.51 **(21.6%)** | 3.14 **(2.10%)** |
| Contacts per unit time, $N/T$ [min$^{-1}$] | 0.59 **(59.0%)** | 0.80 **(64.2%)** | 0.67 **(79.8%)** |
| Average CL of TOIs | 1.0 (0%) | 0.97 (3.7%) | 0.96 (6.0%) |
| Classification efficiency, $\eta$ [hr$^{-1}$] | 0.60 **(91%)** | 0.37 **(32%)** | 0.23 **(12%)** |
| Classification Accuracy (CA) | 1.0 (0%) | 0.93 (3.7%) | 0.71 (9.7%) |
| False Alarm Ratio (FA) | 0.0 (0%) | 0.038 (9.4%) | 0.27 (1.3%) |
| Missed Detection Ratio (MD) | 0.0 (0%) | 0.071 (30%) | 0.29 (18%) |

higher confidence than those provided by the MAC and CMAC algorithms. In fact, the results in Fig. 6.7 show that the IMVP approach results in a much higher CL gain per unit time – namely 91% improvement for set A, 43% improvement for set B, and 18% improvement for set C – as well as in a much smaller standard deviation than that of MAC and CMAC algorithms, indicating that the CL improvement also is more robust.

## 6.2 Autonomous Vehicle Architecture (AVA)

As part of the sea-test preparation, the IMVP algorithm was implemented as a standalone C++ library and integrated within the Autonomous Vehicle Architecture known as AVA. AVA is a software framework initially developed at the Naval Surface Warfare Center Panama City Division (NSWC PCD) in order to simplify S&T development and reduce the recreation of software year by year for research projects [77]. AVA is structured in three layers to provide a balanced level of individual control: High-level mission and sortie management, intermediate task layer with deliberative planning capabilities, and low-level behavior planner for reactive capabilities. Additionally, AVA has functions that provide replanning through levels of monitors and solvers while also interacting with perception modules such as world models or automated target recognition software. A layout of the framework for AVA is provided in Fig. 6.8.

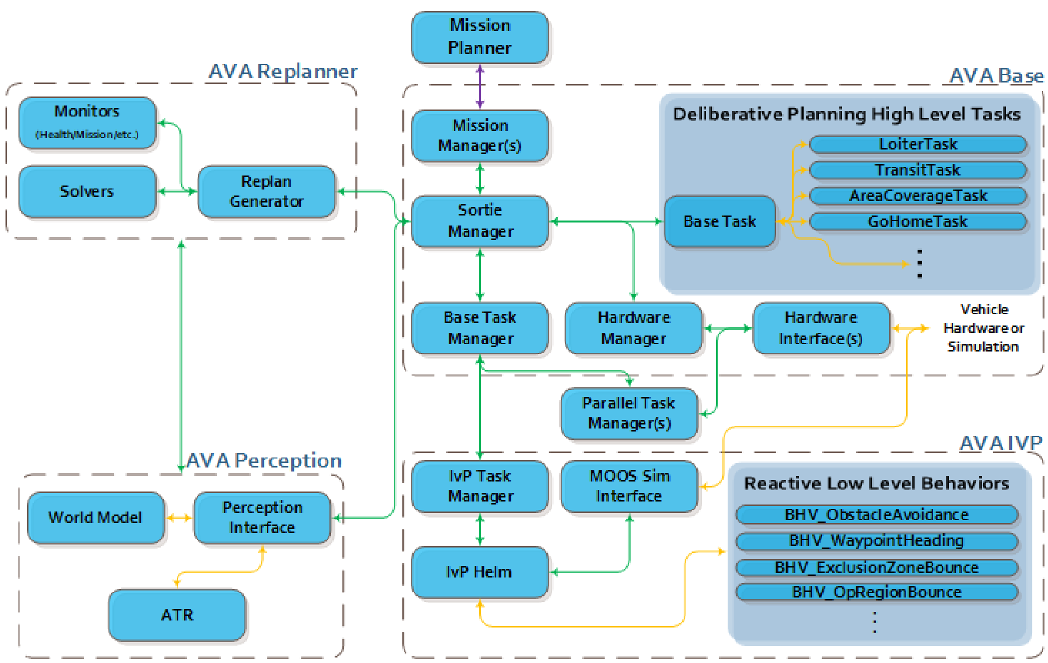



Figure 6.8: Framework for the Autonomous Vehicle Architecture (AVA).

AVA was originally built using the Mission Oriented Operating Suite Interval Programming (MOOS-IvP) environment as a base communication layer while also taking advantage of the low-level behavior components of IvPHelm. Over the past few years, all components of AVA have moved to the Robot Operating System (ROS) 2.0 for the myriad of advantages the new environment provides. Additionally, IvPHelm and other relevant components of MOOS-IvP have also been converted to ROS 2.0 environment under a similarly named ROS-IvP [69]. These components have greatly been improved since their first iteration into the ROS 1.0 environment.

Using the above tools as well as other opens source tools, AVA works to follow the Modular Open System Approach (MOSA) for components by providing a general framework for communicating between components and multiple base classes that will provide general functionality for new components (tasks, behaviors, etc.). Software is made to simplify the addition of new components and have minimal impact to the architecture. The architecture is created to be platform agnostic and has been demonstrated on several unmanned vehicles across multiple domains (undersea, surface, and ground). AVA is also configured to be third party behavior agnostic, having developed multiple interfaces in the past to collaborate and work with software environment such as operating in parallel with other autonomy architectures, such as SeeByte's Neptune, International Partners, etc.

## 6.3 IMVP Sea Test Demonstration

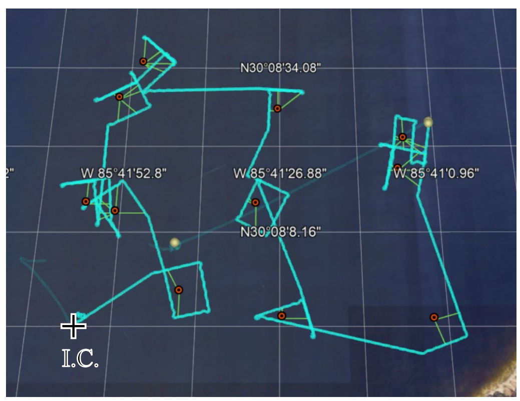
Sea tests were conducted in collaboration with NSWC PCD to demonstrate the feasibility and effectiveness of the IMVP algorithm. The sea trials were performed at Saint Andrew Bay area near Panama City (FL) (Fig. 6.9(a) ). The IMVP planner was first integrated within the AVA architecture (Fig. 6.8) and, then, executed onboard a REMUS 100 for the test case described in Fig. 6.3(a). The REMUS trajectory executed during the sea trial is shown in Fig. 6.9, and the corresponding ATR performance is evaluated using the sonar simulation described in Section 6. The results in Table 6.5 show that the sea-test REMUS trajectory and

Table 6.5: IMVP Algorithm Path Planning and Classification Performance Comparison for the Sea Test and AVA Simulation

| Performance Metric | IMVP Algorithm | |
|---|---|---|
| | Sea Test | AVA Simulation |
| Travel time, $T$ [hr] | 1.15 | 1.20 |
| Contacts per unit time, $N/T$ [min$^{-1}$] | 0.59 | 0.68 |
| Classification efficiency, $\eta$ [hr$^{-1}$] | 0.53 | 0.56 |
| Average TOIs CL | 0.98 | 0.99 |

classification performance are similar or better than those obtained by the AVA simulation. The REMUS speed was maintained at approximately 3 m/s, as in the simulation environment (Section 6). The number of target contacts was, however, reduced from 49 (in AVA simulation) to 40 in the sea test due to disturbance in the yawing motion caused by the sea waves. These missing target contacts result in lower classification accuracy and lower confidence level on some objects. Nevertheless, the target contacts are obtained from all the C-target viewpoint regions that were planned from the IMVP planner. For the targets acquired, the results in Table 6.5 show that the REMUS was able to execute the IMVP path with good accuracy in real undersea environments, thus resulting in similar target classification performance.


(a)




(b)

Figure 6.9: (a) Bird-eye view of the sea tests in the Saint Andrew Bay area in Panama City, FL, and (b) close view of the REMUS IMVP trajectory executed at sea for the target field described in Fig. 13(a), and a vehicle initial condition (I.C.) denoted by the black cross.

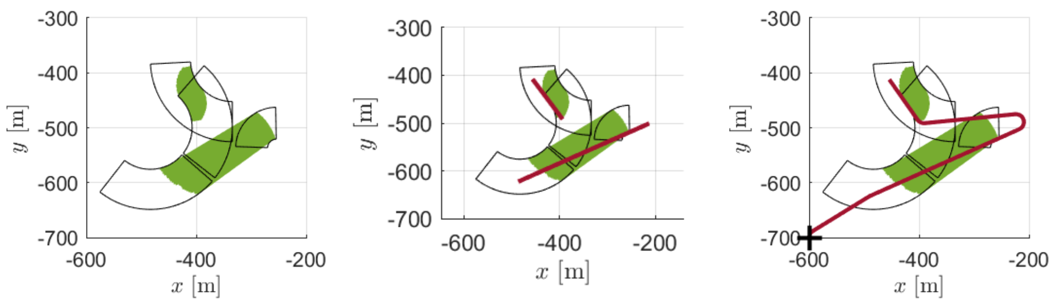
CHAPTER 7

# EXTENSIONS

## 7.1 IMVP Robustness to Navigation Error

The proposed IMVP approach is extended to take into account navigation error in a real-world setting, such as a sea trial. Because the navigation error significantly increases at each turn of a UUV, the IMVP planner is extended to minimize the number of turns of the UUV to attain robustness. Originally, the proposed IMVP algorithm plans a path by sampling waypoints from the neighborhoods intersections. Then, a traveling salesman problem (TSP) solution is used to connect all the sampled waypoints. The final path is constructed by replacing each waypoint with a line segment in order to minimize the sonar image distortion. Thus, the number of turns can increase as the number of waypoints increases. Thus, the IMVP algorithm can miss connecting some collinear waypoints if the waypoints are from the viewpoint regions that are not intersecting.

The IMVP algorithm is modified to find a path that can visit multiple neighborhoods through as many single passes as possible, even when the neighborhoods are disjoint. Intersection of neighborhoods is re-defined such that, if there is any swath that can connect two neighborhoods, two neighborhoods are considered intersecting. The intersection is also re-defined as a set of the line segments that connect two points in each neighborhood at the same vehicle heading angle in configuration space. Then, the same greedy algorithm in the IMVP approach (Algorithm 1) is used to compute the minimal disjoint coverage set by looking at every pair of neighborhoods. An example of this modified IMVP algorithm, namely IMVP-Robust, is described in Fig. 7.1. It is noted that the figure is illustrating each viewpoint region, which is a section of C-targets, from a top-view of the configuration space. Then, with the new-definition of neighborhoods intersections, IMVP-Robust samples swaths from each neighborhoods intersection as shown in Fig. 7.1(b). The path is computed by connecting the sampled swath using greedy TSP algorithm as shown in Fig. 7.1(c).




(a) (b) (c)

Figure 7.1: IMVP-Robust algorithm applied to an example set of neighborhoods (viewpoint regions): (a) minimal disjoint coverage set (shaded regions); (b) sampled swaths; (c) resultant path

This extended IMVP algorithm, or IMVP-Robust, is implemented and compared with existing IMVP approach in two test target fields. In the first test target field, the targets are distributed by forming clusters as shown in Fig. 7.2. The planned paths are then demonstrated through a simulation described in Section 6, and the UUV trajectories are plotted in Fig. 7.2. The number of turns is reduced from 70 in IMVP (Fig. 7.2(a)) to 41 in IMVP-Robust (Fig. 7.2(b)) after applying the modification in neighborhoods intersections, which amounts to a 41% reduction. The classification performance of IMVP-Robust is summarized and compared in Table 7.1. The classification results show that IMVP-Robust results in similar classification performance with IMVP because the same viewpoint regions planned using ECL are visited.

The second test target field consists of underwater objects whose positions are

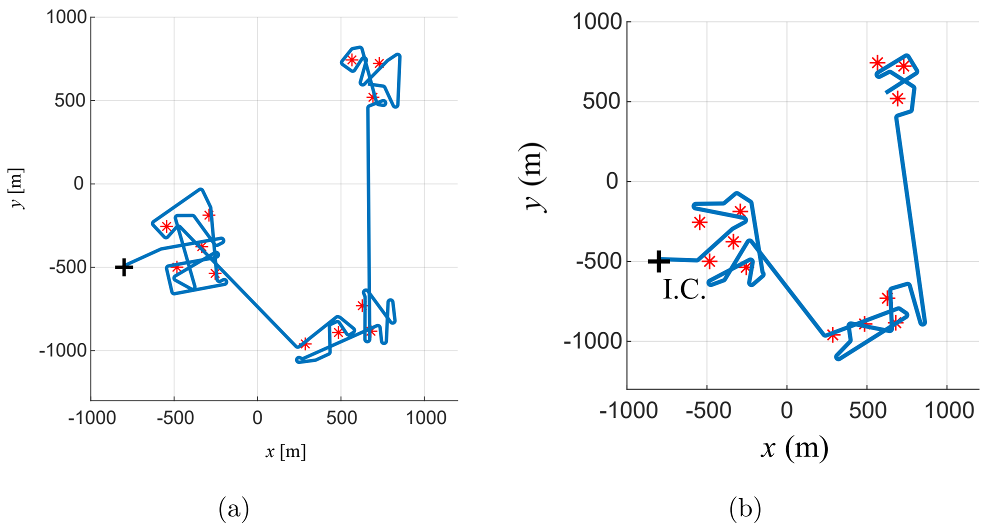
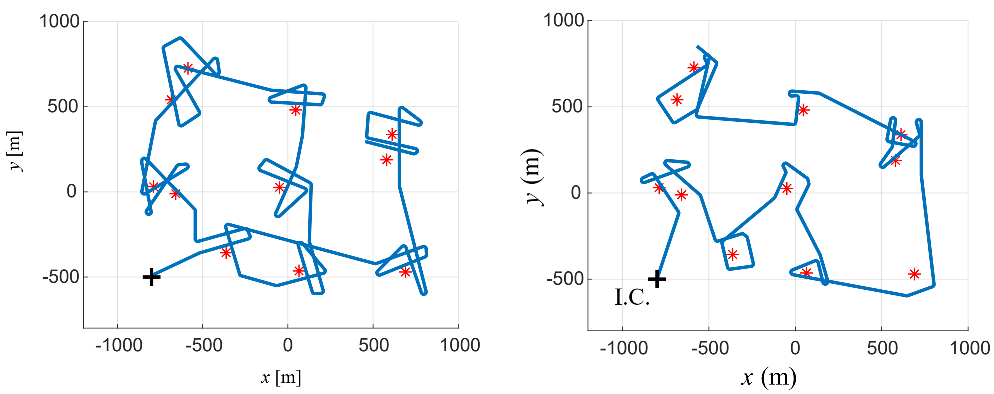



Figure 7.2: Simulated UUV trajectories of (a) IMVP and (b) IMVP-Robust for the objects (red starts) distributed with clusters and the given the initial condition (black cross)

Table 7.1: Path Planning and Classification Performances Comparison of IMVP-Robust for the target field in Fig. 7.2

| Metric | Algorithm | |
|---|---|---|
| | Modified IMVP | IMVP |
| Travel Time [hr] | 1.33 | 1.71 |
| Num. of Images per Time [/min] | 0.69 | 0.67 |
| Num. of Images per Target | 4.58 | 5.75 |
| Average CL of TOIs | 0.94 | 0.95 |
| Classification Accuracy | 0.62 | 0.62 |
| False Alarm Ratio | 0.34 | 0.34 |
| Missed Detection Ratio | 0.38 | 0.38 |

uniformly distributed over the region of interest. The resultant UUV trajectories from IMVP and IMVP-Robust are compared in Fig. 7.3. The number of turns is reduced from 51 in IMVP (Fig. 7.3(a)) to 43 in IMVP-Robust (Fig. 7.3(a)), which amounts to a 16% reduction. This reduction is smaller than the reduction from the first test target field because there is fewer swaths that can connect two waypoints from disjoint neighborhoods. The classification performance of IMVP-Robust is summarized and compared in Table 7.2. The classification results show that IMVP-Robust can also achieve similar classification performance with IMVP.




(a) (b)

Figure 7.3: Simulated UUV trajectories of (a) IMVP and (b) IMVP-Robust for the objects (red starts) distributed uniformly and the given the initial condition (black cross)

## 7.2 IMVP Robustness to Uncertainty in Target Position

The IMVP planner proposed in this dissertation can also be extended to handle the target and localization uncertainty, which is a very common practical issues in real-world setting. The target position uncertainty is considered by utilizing

Table 7.2: Path Planning and Classification Performances Comparison of IMVP-Robust for the target field in Fig. 7.3

| Metric | Algorithm | |
|---|---|---|
| | IMVP-Robust | IMVP |
| Travel Time [hr] | 1.77 | 1.87 |
| Num. of Images per Time [/min] | 0.39 | 0.42 |
| Num. of Images per Target | 3.73 | 4.27 |
| Average CL of TOIs | 0.97 | 0.95 |
| Classification Accuracy | 0.94 | 0.93 |
| False Alarm Ratio | 0.22 | 0.20 |
| Missed Detection Ratio | 0.06 | 0.07 |

the target position probability distribution model in MAC algorithm [6–8], which assumes that the target position has Gaussian noise in 2-dimensional space ($xy$-plane). This extended IMVP planner that takes into account target position uncertainty is referred to as IMVP-Uncertainty. In IMVP-Uncertainty, the target position uncertainty is taken into account by sampling the swaths that passes with the maximum probability of detection, which is defined in [6–8]. The probability of detection is defined as a convolution of sensor profile and target positions' probability distribution over the axis coincides with the side-scan sonar field-of-view geometry, i.e., perpendicular to the UUV heading angle. Therefore, the swath with the highest probability of detection is chosen by considering both sensor profile and the probability distribution of target positions. The length of the swath is also computed based on the probability of detection, which is referred to as swath trimming in MAC algorithm [6–8]. For each chosen swath, the probability of detection is computed with respect to the axis coincides with the swath. The swath is trimmed from its both ends until the integrated probability of detection between the two ends of the swath is larger than the user-chosen threshold.

This extended IMVP algorithm, or IMVP-Uncertainty, is implemented and compared with the IMVP algorithm for a sample target field illustrated in Fig. 7.4. In this comparison, it is assumed that the target position uncertainty is represented by a 2-dimensional Gaussian noise, where the mean values are the actual target positions and standard deviations are set to be 20 (meters) equally with even weighting. The same estimated target positions are input to both IMVP-Uncertainty and the IMVP algorithms in the path planning stage. Then, the actual target position is used in the UUV-sonar simulation (Section 6) to compare how many measurements are still obtained in IMVP-Uncertainty and missed in IMVP algorithm.

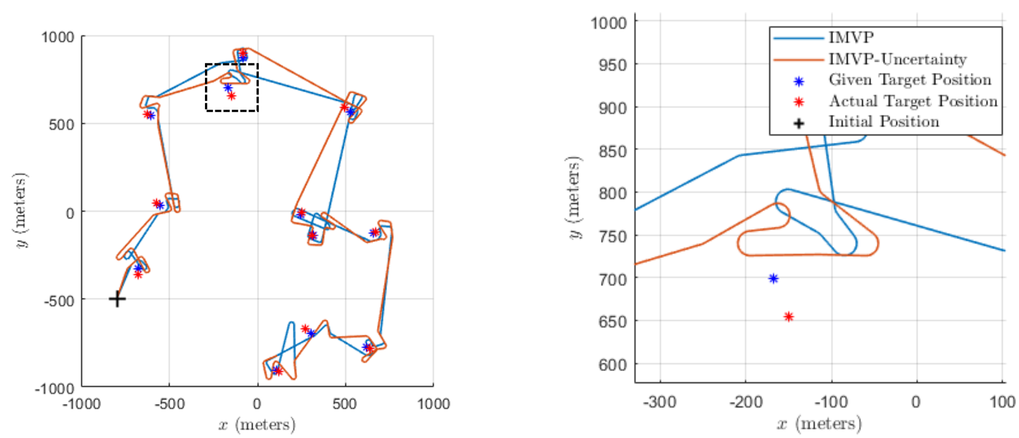



Figure 7.4: Comparison of IMVP-Uncertainty and IMVP: (Left) the whole target field; (Right) Zoomed-in view of the black dotted box.

The paths planned by IMVP-Uncertainty and IMVP are compared in Fig. 7.4. This test case shows that some targets are detected by following the IMVP-Uncertainty path, yet not detected by IMVP path. This result is illustrated in detail in the right subfigure of Fig. 7.4. The number of obtained target contacts and classification performance are compared in Table 7.3 and 7.4, respectively. In both Tables, each value is obtained by averaging 100 trials, and the standard deviation (Std.) is shown in parenthesis. The path planning results show that the IMVP-Uncertainty miss fewer target contacts because the path is planned considering target position distribution. As a result, the results show that this fewer missing target contacts in IMVP-Uncertainty leads to a less reduction of classification performance.

| Algorithm | Expected Number of Contacts | Actual Number of Contacts (Std.) | Difference | Difference in Percentage |
|---|---|---|---|---|
| IMVP-Uncertainty | 44 | 39.4 (3.09) | 4.6 | **10.5%** |
| IMVP | 48 | 34.1 (4.57) | 13.9 | 30.0% |

Table 7.3: Comparison of IMVP and IMVP-Uncertainty on the number of target contacts.

| Metric | Algorithm | Expected Value (Std.) | Actual Value (Std.) | Difference in Percentage |
|---|---|---|---|---|
| TPR | IMVP-Uncertainty | 0.88 (0.015) | 0.85 (0.048) | **3.4** |
| | IMVP | 0.88 (0.016) | 0.78 (0.11) | 11 |
| False Alarm | IMVP-Uncertainty | 0.099 (0.0084) | 0.097 (0.025) | **2.0** |
| | IMVP | 0.070 (0.0070) | 0.11 (0.041) | 57 |
| Missed Detection | IMVP-Uncertainty | 0.12 (0.015) | 0.14 (0.033) | **17** |
| | IMVP | 0.12 (0.016) | 0.22 (0.051) | 83 |

Table 7.4: Comparison of IMVP and IMVP-Robust algorithms on classification performance

CHAPTER 8

# CONCLUSION

This dissertation presents a novel approach to planning sensor measurements and motions in applications that require multiple looks or views per target, such as underwater imaging. The approach, referred to as informative multi-view planning or IMVP, takes into account the sensor field-of-view geometry and the target position and orientation by constructing a so-called C-target in the mobile sensor's configuration space. By this approach, the expected information value of every possible sensor look (or view) of the target can be quantified systematically as a function of the sensor configuration. The IMVP approach is demonstrated on a UUV-based side-scan sonar that must classify multiple targets with a minimum required confidence level. As a result, the information value of C-target regions is represented by the expected confidence level derived from prior sensor measurements and ATR model.

An approximate algorithm for solving the multi-view planning problem, reduced to a generalized traveling salesman problem with neighborhoods, is also presented to leverage intersecting C-target regions and maximize the number of targets detected in sonar images. While the proposed GTSPN solution requires high computation time on computing the neighborhoods intersection, the novel physics-inspired TSPN solution proposed in this dissertation can successfully find a reasonable TSPN tour by considering the neighborhoods intersection using coverage cone. The proposed TSPN solution only adds a computation time complexity of $O(n \log n)$ to existing TSP solvers to find a TSPN tour. The main advantage of this geometric TSPN solution is that no prior data is required to find a solution unlike existing methods based on supervised learning and evolutionary algorithms.

The results obtained from a high-fidelity closed-loop imaging sonar simulation show that IMVP significantly outperforms existing state-of-the-art multi-view planning methods, known as MAC and CMAC algorithms. In fact, IMVP-guided sonar are able to complete multi-target classification tasks with equal or superior classification performance in approximately half the time of existing algorithms. Also, the IMVP approach is shown to adapt the UUV path based on individual target features, the difficulty of classification task, and the configuration of the target field. In real operations, the IMVP method can be reformulated to have a time constraint and to maximize the expected confidence level or information gain on target classification. In this formulation, the IMVP method will vary the number of views for each target to limit the operation time while choosing the most informative viewpoints. When an additional total time constraint is given, the IMVP approach can be modified to limit the number of views by removing the least informative viewpoint. Also, when there is a constant ocean current applied to the vehicle, the IMVP approach can be extended such that the planned viewpoint regions are connected through a minimum-time trajectory that optimizes the operation time considering vehicle dynamics. When there is a significant navigation error or target position uncertainty due to the environmental condition, the IMVP approach can also extended to include some waypoints that the UUV can surface up and update its information or integrate the localization methods that utilizes some known underwater structures. The proposed IMVP approach can also be extended to operate in the online configuration, where the contact reinspection is performed right after the detection. This online approach will plan an additional view based on the expected confidence level until the confidence level reaches a user-chosen threshold. Therefore, the IMVP approach is not only promising for other multi-view sensor applications but also for the development of adaptive planning algorithms.

Future work includes finding an approximation factor of LKH-Geometric solution, sensor path planning under significant target uncertainty, and incorporating vehicle dynamics to sensor path planning. Firstly, the approximation factor of the current LKH-Geometric solution depends on the TSP solution from LKH algorithm. Replacing this LKH with a new geometric TSP solution that can provide the sequence of circles to visit in LKH-Geometric will enable the whole algorithm to have an approximation factor that can be proven geometrically. Secondly, when the target position uncertainty becomes significant, future work on updating the knowledge on the target positions based on the obtained measurements may improve the robustness of current IMVP approach. This future work may incorporate some prior knowledge on the sensor system that collected the target positions in pre-surveying in order to efficiently update and correct the target positions.

Finally, possible next steps include developing an underwater sensor path planning approach that incorporates vehicle dynamics. Depending on the vehicle types (UUV, uncrewed surface vehicle, remotely operated underwater vehicle, etc.) and operation environment (shallow or deep water), the hydrodynamic forces applying to the vehicle can affect on-board sensor performance. This integration may require prediction of ocean environment in for a certain future time horizon and optimize the path in a receding-horizon fashion. By incorporating vehicle's dynamics and external forces from surrounding underwater environment, the sensor path planning approach will enable the system to maintain desired sensor performance in a harsh and dynamic environment.

## BIBLIOGRAPHY

[1] E. Abbena, S. Salamon, and A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*. Textbooks in Mathematics. CRC Press, 2017.

[2] Sergey Alatartsev, Vera Mersheeva, Marcus Augustine, and Frank Ortmeier. On optimizing a sequence of robotic tasks. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 217–223. IEEE, 2013.

[3] Sergey Alatartsev, Sebastian Stellmacher, and Frank Ortmeier. Robotic task sequencing problem: A survey. *Journal of intelligent & robotic systems*, 80(2):279–298, 2015.

[4] Kelli Baumgartner and Silvia Ferrari. A geometric transversal approach to analyzing track coverage in sensor networks. *IEEE Transactions on Computers*, 57(8):1113–1128, 2008.

[5] Kelli AC Baumgartner, Silvia Ferrari, and Anil V Rao. Optimal control of an underwater sensor network for cooperative target tracking. *IEEE Journal of Oceanic Engineering*, 34(4):678–697, 2009.

[6] Matthew J Bays, Apoorva Shende, Daniel J Stilwell, and Signe A Redfield. A solution to the multiple aspect coverage problem. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1531–1537. IEEE, 2011.

[7] Matthew J Bays, Apoorva Shende, Daniel J Stilwell, and Signe A Redfield. Theory and experimental results for the multiple aspect coverage problem. *Ocean Engineering*, 54:51–60, 2012.

[8] Matthew Jason Bays. *Stochastic motion planning for applications in subsea survey and area protection*. PhD thesis, Virginia Tech, 2012.

[9] Israel Becerra, Luis M Valentín-Coronado, Rafael Murrieta-Cid, and Jean-Claude Latombe. Reliable confirmation of an object identity by a mobile robot: A mixed appearance/localization-driven motion approach. *The International Journal of Robotics Research*, 35(10):1207–1233, 2016.

[10] Ioana O Bercea. Improved bounds for the traveling salesman problem with neighborhoods on uniform disks. *arXiv preprint arXiv:1809.07159*, 2018.

[11] Philippe Blondel. *The handbook of sidescan sonar*. Springer Science & Business Media, 2010.

[12] Patrice Brehmer, Thierry Lafont, Stratis Georgakarakos, Erwan Josse, François Gerlotto, and Christophe Collet. Omnidirectional multibeam sonar monitoring: applications in fisheries science. *FISH and Fisheries*, 7(3):165–179, 2006.

[13] Ashish Kumar Budhiraja. *View Point Planning for Inspecting Static and Dynamic Scenes with Multi-Robot Teams*. PhD thesis, Virginia Tech, 2017.

[14] Chenghui Cai and Silvia Ferrari. Comparison of information-theoretic objective functions for decision support in sensor systems. In *2007 American Control Conference*, pages 3559–3564. IEEE, 2007.

[15] Chenghui Cai and Silvia 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.

[16] T-H Hubert Chan and Khaled Elbassioni. A qptas for tsp with fat weakly disjoint neighborhoods in doubling metrics. *Discrete & Computational Geometry*, 46(4):704–723, 2011.

[17] Shi Chang, Jason Isaacs, Bo Fu, Jaejeong Shin, Pingping Zhu, and Silvia Ferrari. Confidence level estimation in multi-target classification problems. In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIII*, volume 10628, page 1062818. International Society for Optics and Photonics, 2018.

[18] Daniel T Cobra, Alan V Oppenheim, and Jules S Jaffe. Geometric distortions in side-scan sonar images: a procedure for their estimation and correction. *IEEE Journal of oceanic engineering*, 17(3):252–268, 1992.

[19] Giovanni Comarela, Kênia Gonçalves, Gisele Lobo Pappa, Jussara Almeida, and Virgílio Almeida. Robot routing in sparse wireless sensor networks with continuous ant colony optimization. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, pages 599–606, 2011.

[20] Michel Couillard, John Fawcett, and Matt Davison. Optimizing constrained search patterns for remote mine-hunting vehicles. *IEEE Journal of Oceanic Engineering*, 37(1):75–84, 2012.

[21] Harold Scott Macdonald Coxeter. *Introduction to geometry.* New York, London, 1961.

[22] Mark de Berg, Joachim Gudmundsson, Matthew J Katz, Christos Levcopoulos, Mark H Overmars, and A Frank van der Stappen. Tsp with neighborhoods of varying size. *Journal of Algorithms*, 57(1):22–36, 2005.

[23] Adrian Dumitrescu and Joseph SB Mitchell. Approximation algorithms for tsp with neighborhoods in the plane. *Journal of Algorithms*, 48(1):135–159, 2003.

[24] Adrian Dumitrescu and Csaba D Tóth. The traveling salesman problem for lines, balls, and planes. *ACM Transactions on Algorithms (TALG)*, 12(3):43, 2016.

[25] Adrian Dumitrescu and Csaba D Tóth. Constant-factor approximation for tsp with disks. In *A Journey Through Discrete Mathematics*, pages 375–390. Springer, 2017.

[26] Khaled Elbassioni, Aleksei V Fishkin, Nabil H Mustafa, and René Sitters. Approximation algorithms for euclidean group tsp. In *International Colloquium on Automata, Languages, and Programming*, pages 1115–1126. Springer, 2005.

[27] Khaled Elbassioni, Aleksei V Fishkin, and René Sitters. Approximation algorithms for the euclidean traveling salesman problem with discrete and continuous neighborhoods. *International Journal of Computational Geometry & Applications*, 19(02):173–193, 2009.

[28] Jan Faigl. Gsoa: Growing self-organizing array-unsupervised learning for the close-enough traveling salesman problem and other routing problems. *Neurocomputing*, 312:120–134, 2018.

[29] Jan Faigl, Petr Váňa, and Jindřiška Deckerová. Fast heuristics for the 3-d multi-goal path planning based on the generalized traveling salesman problem with neighborhoods. *IEEE Robotics and Automation Letters*, 4(3):2439–2446, 2019.

[30] John Fawcett, Vincent Myers, David Hopkin, Anna Crawford, Michel Couillard, and Benoit Zerr. Multiaspect classification of sidescan sonar images: Four different approaches to fusing single-aspect information. *IEEE Journal of Oceanic Engineering*, 35(4):863–876, 2010.

[31] S. Ferrari. Track coverage in sensor networks. In *2006 American Control Conference*, pages 7 pp.–, 2006.

[32] S. Ferrari and T.A. Wettergren. *Information-driven Planning and Control*. Cyber physical systems series. MIT Press, 2021.

[33] Silvia Ferrari and Greg Foderaro. A potential field approach to finding minimum-exposure paths in wireless sensor networks. In *2010 IEEE International Conference on Robotics and Automation*, pages 335–341. IEEE, 2010.

[34] Silvia Ferrari and Alberto Vaghi. Demining sensor modeling and feature-level fusion by bayesian networks. *IEEE Sensors Journal*, 6(2):471–483, 2006.

[35] Daniel Garcia-Castellanos and Umberto Lombardo. Poles of inaccessibility: A calculation algorithm for the remotest places on earth. *Scottish Geographical Journal*, 123(3):227–233, 2007.

[36] Iacopo Gentilini, François Margot, and Kenji Shimada. The travelling salesman problem with neighbourhoods: Minlp solution. *Optimization Methods and Software*, 28(2):364–378, 2013.

[37] Peter T Gough and David W Hawkins. Imaging algorithms for a strip-map synthetic aperture sonar: Minimizing the effects of aperture errors and aperture undersampling. *IEEE Journal of Oceanic Engineering*, 22(1):27–39, 1997.

[38] Roy Edgar Hansen. Introduction to synthetic aperture sonar. In *Sonar systems*. InTech, 2011.

[39] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130, 2000.

[40] Geoffrey A Hollinger, Urbashi Mitra, and Gaurav S Sukhatme. Active classification: Theory and application to underwater inspection. In *Robotics Research*, pages 95–110. Springer, 2017.

[41] Jason C Isaacs. Unsupervised domain transfer of latent dirichlet allocation derived representations from synthetic aperture sonar imagery. In *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XVIII*, volume 8709, page 87090G. International Society for Optics and Photonics, 2013.

[42] Jason C Isaacs. Sonar automatic target recognition for underwater uxo remediation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 134–140, 2015.

[43] Jason C Isaacs and Anuj Srivastava. Geodesic shape distance and integral invariant shape features for automatic target recognition. In *OCEANS 2010 MTS/IEEE SEATTLE*, pages 1–6. IEEE, 2010.

[44] Jason C Isaacs and James D Tucker. Diffusion features for target specific recognition with synthetic aperture sonar raw signals and acoustic color. In *CVPR 2011 WORKSHOPS*, pages 27–32. IEEE, 2011.

[45] Jason C Isaacs and James D Tucker. Signal diffusion features for automatic target recognition in synthetic aperture sonar. In *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pages 461–465. IEEE, 2011.

[46] Jerzy Kocik. A theorem on circle configurations. *arXiv preprint arXiv:0706.0372*, 2007.

[47] Daniel Köhntopp, Benjamin Lehmann, Dieter Kraus, and Andreas Birk. Classification and localization of naval mines with superellipse active contours. *IEEE Journal of Oceanic Engineering*, 2018.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[49] James J Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *ICRA*, pages 3993–3998. Citeseer, 2004.

[50] Jean-Claude Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[51] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[52] Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

[53] 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.

[54] Douglas G Macharet and Mario FM Campos. A survey on routing problems and robotic systems. *Robotica*, 36(12):1781–1803, 2018.

[55] William Kenneth Mennell. *Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, and sequence-dependent team orienteering problem*. University of Maryland, College Park, 2009.

[56] Joseph SB Mitchell. A constant-factor approximation algorithm for tsp with pairwise-disjoint connected neighborhoods in the plane. In *Proceedings of the twenty-sixth annual symposium on Computational geometry*, pages 183–191, 2010.

[57] Joseph SB Mitchell. A ptas for tsp with neighborhoods among fat regions in the plane. *arXiv preprint arXiv:1703.01646*, 2017.

[58] Joseph SB Mitchell et al. Geometric shortest paths and network optimization. *Handbook of computational geometry*, 334:633–702, 2000.

[59] Vincent Myers and John Fawcett. A template matching procedure for automatic target recognition in synthetic aperture sonar imagery. *IEEE Signal Processing Letters*, 17(7):683–686, 2010.

[60] Vincent Myers and David P Williams. Adaptive multiview target classification in synthetic aperture sonar images using a partially observable markov decision process. *IEEE Journal of Oceanic Engineering*, 37(1):45–55, 2012.

[61] Bao Nguyen, Matthew J Bays, Apoorva Shende, and Daniel J Stilwell. An approach to subsea survey for safe naval transit. In *OCEANS'11 MTS/IEEE KONA*, pages 1–6. IEEE, 2011.

[62] Charles E Noon and James C Bean. An efficient transformation of the generalized traveling salesman problem. *INFOR: Information Systems and Operational Research*, 31(1):39–44, 1993.

[63] Narcís Palomeras, Natalia Hurtós, Eduard Vidal, and Marc Carreras. Autonomous exploration of complex underwater environments using a probabilistic next-best-view planner. *IEEE Robotics and Automation Letters*, 4(2):1619–1625, 2019.

[64] Tan Qin, Yonghui Zhao, Guocong Lin, Shufan Hu, Cong An, Dexiang Geng, and Chunfeng Rao. Underwater archaeological investigation using ground

penetrating radar: A case analysis of shanglinhu yue kiln sites (china). *Journal of Applied Geophysics*, 154:11–19, 2018.

[65] Sharmin Rahman, Alberto Quattrini Li, and Ioannis Rekleitis. Sonar visual inertial slam of underwater structures. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[66] Maryam Rahnemoonfar, Abdullah F Rahman, Richard J Kline, and Austin Greene. Automatic seagrass disturbance pattern identification on sonar images. *IEEE Journal of Oceanic Engineering*, 44(1):132–141, 2018.

[67] Gerhard Reinelt. Tsplib—a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[68] Sanem Sariel, Tucker Balch, and Nadia Erdogan. Naval mine countermeasure missions. *IEEE robotics & automation magazine*, 15(1):45–52, 2008.

[69] Mark Snyder, Joshua N Weaver, and Matthew J Bays. Ros-ivp: Porting the interval programming suite into the robot operating system for maritime autonomy. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–6. IEEE, 2016.

[70] Abhishek Sriraman and Matthew J Bays. Efficient reacquire and identify path planning over large areas. In *Oceans-St. John's, 2014*, pages 1–7. IEEE, 2014.

[71] Jason R Stack and Cheryl M Smith. Combining random and data-driven coverage planning for underwater mine detection. In *Oceans 2003. Celebrating the Past... Teaming Toward the Future (IEEE Cat. No. 03CH37492)*, volume 5, pages 2463–2468. IEEE, 2003.

[72] David Sweet, Edward Ott, and James A Yorke. Topology in chaotic scattering. *Nature*, 399(6734):315–316, 1999.

[73] Pratap Tokekar, Joshua Vander Hook, David Mulla, and Volkan Isler. Sensor planning for a symbiotic uav and ugv system for precision agriculture. *IEEE Transactions on Robotics*, 32(6):1498–1511, 2016.

[74] Robert J Urick. *Principles of underwater sound*. Tata McGraw-Hill Education, 1967.

[75] Kevin Vicencio, Brian Davis, and Iacopo Gentilini. Multi-goal path planning based on the generalized traveling salesman problem with neighborhoods. In

*2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2985–2990. IEEE, 2014.

[76] Daniel J Walker and J Brian Alford. Mapping lake sturgeon spawning habitat in the upper tennessee river using side-scan sonar. *North American Journal of Fisheries Management*, 36(5):1097–1105, 2016.

[77] Joshua Weaver, Jim Perkins, and Dan Sternlicht. Advanced autonomy architecture for maritime applications ava. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–8. IEEE, 2016.

[78] Hongchuan Wei, Pingping Zhu, Miao Liu, Jonathan P How, and Silvia Ferrari. Automatic pan–tilt camera control for learning dirichlet process gaussian process (dpgp) mixture models of multiple moving targets. *IEEE Transactions on Automatic Control*, 64(1):159–173, 2018.

[79] David Williams and Johannes Groen. Multi-view target classification in synthetic aperture sonar imagery. In *Proceedings of the International Conference and Exhibition on Underwater Acoustic Measurements*, pages 699–704, 2009.

[80] David P Williams. Fast target detection in synthetic aperture sonar imagery: A new algorithm and large-scale performance analysis. *IEEE Journal of Oceanic Engineering*, 40(1):71–92, 2014.

[81] David P Williams and Samantha Dugelay. Multi-view sas image classification using deep learning. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–9. IEEE, 2016.

[82] Bo Yuan, Maria Orlowska, and Shazia Sadiq. On the optimal robot routing problem in wireless sensor networks. *IEEE transactions on knowledge and data engineering*, 19(9):1252–1261, 2007.

[83] Guoxian Zhang, Silvia Ferrari, and Chenghui Cai. A comparison of information functions and search strategies for sensor planning in target classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(1):2–16, 2012.

[84] 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.

[85] Pingping Zhu, Jason Isaacs, Bo Fu, and Silvia Ferrari. Deep learning feature

extraction for target recognition and classification in underwater sonar images. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pages 2724–2731. IEEE, 2017.