University of Pisa

Engineering Ph.D. School "Leonardo da Vinci "

Ph.D. Program in Automation, Robotics, Bioengineering

SSD ING/INF- 04 - Automatica

Information-driven cooperative approaches for AUVs seabed surveying

Ph.D. candidate:
Daniele Meucci

Supervisor:
Prof. Andrea Caiti

To Giada, Dario, Noemi and all the people who encourage me.

Daniele

Acknowledgments

I would like to express my gratitude to my advisor Prof. Andrea Caiti for his continuous support during all my Ph.D. studies and research, for having motivated me and having given me the opportunity to perform the doctoral research. Moreover a special thank goes to Dr. Francesco Di Corato for his technical support and constructive discussions.

Thank you.
Pisa, March 5, 2015.

Abstract

This Thesis investigates innovative exploration methods for marine area search with teams of autonomous underwater vehicles (AUVs). In particular, methods for cooperative adaptive motion planning have been developed, general in nature, but in our case applied to the field of marine archaeological search, where the goal is to find remnants or objects resting on, or buried in, the seabed.

The exploration and motion planning problem is divided in two main lines of investigation. The first consists in defining a map of a priori detection probability in accordance with the available information and data over the survey area. Therefore, a refined mathematical method, that uses Parzen windows theory with Gaussian kernels, is developed for building the a priori map. The Renyi's entropy is used as the metric indicating relative information gain.

The second line of investigation instead defines how to compute the optimal waypoints for each AUV when the search mission is in progress. It can be seen as a classical problem of motion planning, which in marine environment usually involves preplanning paths offline before the exploration, either zig-zag or regular lawn-mower transects. The lawn-mower patterns have some failings:

- The AUV may not be able to search in marine areas where the a priori probability is optimum. Hence, the AUV does not move in areas with higher density of objects resting on or buried in the seabed.

- If during the mission some objects are discovered the pre-specified path does not change. Instead, these objects may have influence on a priori information used at the beginning of the mission and therefore a new planning path may be requested.

- The map of a priori detection probability is not updated dynamically with the exploration in progress.

- The AUVs are not able to establish a cooperative communication and localization procedure. Hence, once the vehicle submerges, its location estimate will drift, eventually deviating from the pre-specified paths.

These failures are ridden out using a new online and adaptive approach to define the AUVs' paths. Therefore, a cooperative distributed algorithm is developed defining the AUVs' waypoints by the minimization of the information entropy over the a priori map. Note that the a-priori map built as previously indicated is naturally suited to this approach. The algorithm is implemented by partitioning the marine area through the Equitable Power Diagrams theory, by potential functions for motion planning and taking into account communication constraints.

The benefits of the proposed algorithms are evaluated within the application field of underwater archaeology. In particular, a

performance metric has been defined in terms of relicts found in a fixed time, time to complete the mission, number of relicts found and area explored for relicts found. The Tuscan Archipelago database, kindly made available to us by the Tuscan Superintendence on Cultural Heritage, has given the ground information to apply the investigated algorithms. Simulations results are summarized to show the effectiveness of the novel proposed exploration method. While the performance results are tied to the application domain chosen, it is clear that the methodology and approaches proposed can also be used for other search and rescue applications.

Contents

CONTENTS

List of Figures

LIST OF FIGURES

LIST OF FIGURES

Chapter 1

Preface

1.1 Motivation

In the near future autonomous underwater systems technology will compete against its ground and aerial robotics counterparts. Groups of autonomous agents, fixed or mobile, will be used to perform ocean monitoring and explorative operation for oceanographic, biological and geophysical purposes. There are a number of advantages in considering a team of small autonomous underwater vehicles (AUVs) for specific operations instead of a single, large, more powerful AUV:

- the team will not be affected as a whole by the malfunctioning of a single vehicle, or at least the system performance will degrade gently;
- scale economies may be gained in vehicle production;
- launch and recovery issues are less relevant for small vehicles;

- overall mission time may be minimized, with consequent cost savings.

Nevertheless a team of mobile autonomous vehicles needs to be deployed over regions, to move in a synchronized manner or to assume a specified pattern.

Multi agents systems introduce numerous advantages for application in area mapping, environmental surveying and underwater surveillance system but need to address the problems of motion coordination, communication and localization. The motion coordination can be inspired by the behaviour of self organized systems, from interactions among molecules and cells to the ecology of animal groups. Flocks of birds and schools of fish travel in formation and act as one unit, allowing these animals to defend themself against predators, protect their territories and explore new areas. A multi agents system is like a group of animals that move together. However, the motion cooperation may be intended in several ways, from simply having more vehicles pursuing different pre-planned missions in different areas, to interaction among the vehicles throughout the mission (as in our case), to strict formation control, the strongest form of cooperation.

Historically, in marine environment the motion cooperation was intended as in the first way and therefore the agents seabed surveys were using primitive heuristic planning methods, such zig-zag or lawn-mower paths. Such an approach has different limitations:

- Plans are not adaptive to the situation where the vehicle does not exactly follow the prescribed path either because it is infeasible or due to unexpected external disturbances.

- Once the vehicle submerges, its location estimate will drift, eventually deviating from the pre-specified path.

- Although some events may require to adjust the pre-specified paths, as the mission progresses, it is not possible to change the paths.

These issues are overcome when the motion coordination is intended as interaction among the vehicles throughout the mission. Therefore several benefits in terms of efficiency and robustness are achieved. The advantages can be exploited in a multi agent system of AUVs to survey the seabed. However with the benefits come additional costs in terms of complexity of algorithms required to control and coordinate multiple vehicles that have limited on-board computation capabilities. Communicating among vehicles in water is limited due to lack of available bandwidth, slow throughput rate, and unreliability of the acoustic channel.

1.2 Thesis Overview and Contributions

This thesis focuses on the problem of cooperative search within the marine environment, with specific application to the case of objects resting on the seabed in marine areas of archaeological interest.

The investigated method is: 1) cooperative in the sense that the availability of the multi agents system is exploited by letting each vehicle explores a sub-region of the search area, 2) adaptive in the sense that plans can be updated based on sensor data that are gathered during the mission and 3) online in the sense that the vehicles'

waypoints are computed combining the current mission state with the a priori detection information.

The determination of maps of a priori percentage density of expected findings is in itself a problem; in this thesis it is proposed to exploit the theory of refined Parzen windows, applied to the available a priori information. Within the chosen application field (i.e., marine archaeology) the approach is particularly valuable since it can deal with information at different degrees of reliability. Clearly, this may be the case not only of archaeological findings but also of other important application fields, as those related to seabed dumpings, hazard waste, etc.

In the algorithm proposed, the a priori density map is updated within the mission taking into account the sensor data gathered throughout the exploration. The method has significant advantages for teams of AUVs employed in ocean monitoring and explorative operations for oceanographic, biological and geophysical purposes:

- The a priori and a posteriori density maps design a more accurate representation of likelihood to gather relevant data.

- The vehicles move over the survey area taking into account communication and localization constraints.

- The mission is completed more efficiently in terms of relicts found, mission time employed, area explored.

1.3 Organization of the Thesis

The thesis is organized as follows: Chapter 2 is devoted to describe the background and literature review. Specifically within the chapter, the sections introduce the information theory (Section 2.1), Voronoi and Equitable Power diagrams (Section 2.2), path planning (Section 2.3), side-looking sensors (Section 2.4) and AUV communication-localization (Section 2.5). Chapter 3 describes the method to consistently represent the prior knowledge and the consequent determination of the prior density map over a marine area of archaeological interest. Chapter 4 presents the core of the proposed Cooperative Algorithm for seabed surveying with a team of AUVs. Chapter 5 presents a novel Multi-RRT* algorithm. Finally, the simulation scenario, together with the performance indicators for the assessment of the algorithms for seabed surveying in above Chapter proposed and the results of numerical simulations are presented in Chapter 6.

Chapter 2

Background

This thesis treats many several research domains to propose a novel exploration method for marine area search. Therefore, this chapter is dedicated to present the background of the most important areas of research examined, mainly relating to:

- Information theory, the Shannon information and the Renyi's entropy.
- Voronoi diagram and its generalization the Equitable Power diagram.
- Robotics motion planning, with focus on the sampling-based motion planning philosophy.
- Side Scan Sonar (SSS) and Multi Beam Echo Sounders (MBES), the side-looking sensors which are used as payload for the AUVs.
- Underwater Acoustic Communication and Localization

2.1 Information Theory

The information theory [1] has been developed by Claude Shannon in 1948 by his seminal work, ‘‘A Mathematical Theory of Communication’’ [58] . Originally the theory was founded to find the fundamental limits of data compression, communication and storage, and it has now broadened to diverse fields [55], [20]. Indeed, it has resolved two fundamental questions in communication theory: what is the ultimate data compression (answer: the entropy), and what is the ultimate transmission rate of communication (answer: the channel capacity). Currently, information theory has fundamental contributions to make in statistical physics (thermodynamics), computer science (Kolmogorov complexity or algorithmic complexity), statistical inference (Occam's Razor: "The simplest explanation is best"), and to probability and statistics (error exponents for optimal hypothesis testing and estimation). Figure 2.1 describes the relationship of information theory to other fields. As the figure suggests, information theory intersects physics (statistical mechanics), mathematics (probability theory), electrical engineering (communication theory), and computer science (algorithmic complexity).

2.1.1 Shannon Information

The *Entropy* is a quantity with many proprieties that defines the notion of what a measure of information should be. In more detail, Shannon has defined the *Entropy* as a measure of the uncertainty of

[1] The review presented in this section is based on the book by Thomas M. Cover and Joy A. Thomas [21].

2.1 Information Theory

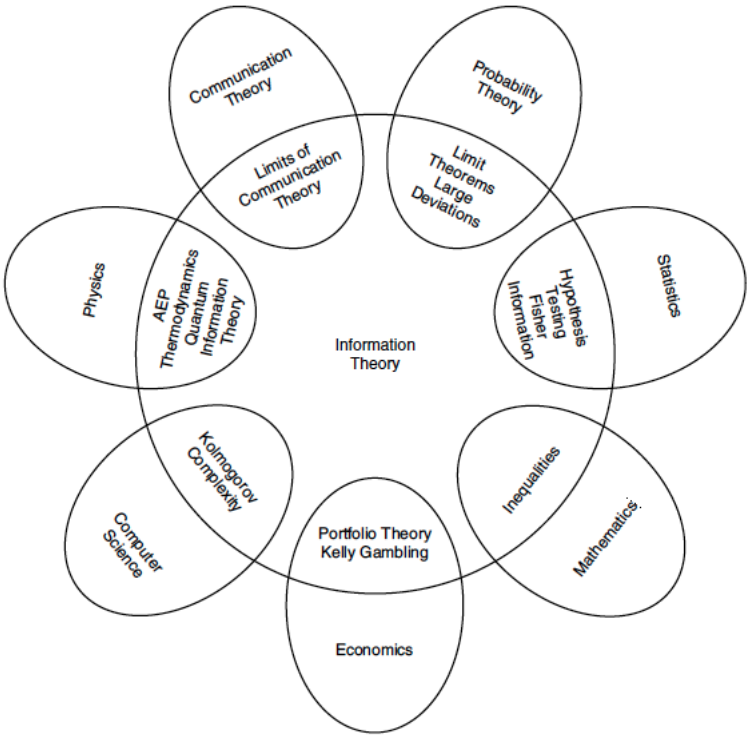


Figure 2.1: *Information theory. (Figure appears in [21])*

a random variable.

Let X be a discrete random variable with alphabet $\mathcal{X}$ and probability mass function $p(x) = \Pr\{X = x\}$, $x \in \mathcal{X}$. The Shannon entropy $H(X)$ of a discrete random variable X is defined as:

$$H(X) = -\sum_{x \in \chi} p(x) \log p(x) \tag{2.1}$$

or similarly for continuous random variable X, with probability density function $p(x)$, is defined as:

$$H(X) = -E[\log p(x)] = -\int p(x) \log p(x) dx \tag{2.2}$$

The log is to the base 2 and entropy is expressed in bits. The base of the logarithm can also be in e measuring the entropy in *nats*, but in that case it is specified with $H_e(X)$. Note that entropy is a functional of the distribution of X. It does not depend on the actual values taken by the random variable X, but only on the probabilities.

As an example consider:

$$X = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1 - p \end{cases} \tag{2.3}$$

Therefore the Shannon entropy is:

$$H(X) = -p \log p - (1-p) \log(1-p) = H(p) \tag{2.4}$$

$H(X)$ can be represented as a function of p, in this case. Figure 2.2 shows the graph of the function $H(p)$ and it illustrates some of the basic properties of entropy. The $H(p)$ is a concave function of the distribution and equals 0 when $p = 0$ or 1. This makes sense,

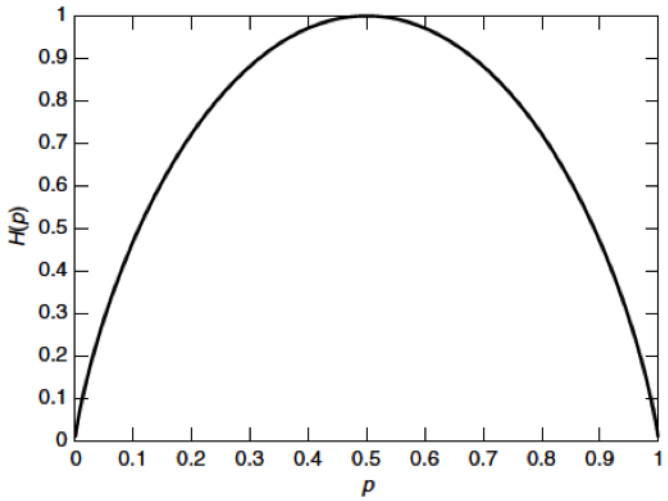


Figure 2.2: *Shannon entropy*

because when $p = 0$ or 1, the variable is not random and there is no uncertainty. Similarly, the uncertainty is maximum when $p = \frac{1}{2}$, which also corresponds to the maximum value of the entropy. The definition of entropy can also be extended through the *joint entropy* for two random variables X and Y.

The *joint entropy* $H(X, Y)$ of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ is defined as:

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y) \tag{2.5}$$

which can also be expressed as

$$H(X, Y) = -E \log p(X, Y) \tag{2.6}$$

Now we defines the *conditional entropy* of a random variable given another as the expected value of the entropies of the condi-

tional distributions, averaged over the conditioning random variable. If $(X, Y) \sim p(x, y)$, the *conditional entropy* $H(Y \mid X)$ is defined as:

$$\begin{aligned} H(Y \mid X) &= \sum_{x \in \mathcal{X}} p(x) H(Y \mid X = x) \\ &= -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y \mid x) \log p(y \mid x) \\ &= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y \mid x) \\ &= -E \log p(Y \mid X) \end{aligned} \tag{2.7}$$

The entropy of a pair of random variables is the entropy of one plus the conditional entropy of the other, as proved in the following theorem.

Theorem 1

$$H(X, Y) = H(X) + H(Y \mid X) \tag{2.8}$$

Another two related concepts regarding the entropy are defined: relative entropy and mutual information. The relative entropy is a measure of the distance between two distributions. Therefore, given two distributions p and q the *relative entropy* $D(p \parallel q)$ measures the inefficiency of assuming that the distribution is q when the true distribution is p. For example, if we knew the true distribution p of the random variable, we could construct a code with average description length $H(p)$. If, instead, we used the code for a distribution q, we would need $H(p) + D(p \parallel q)$ bits on the average to describe the random variable. Instead, the mutual information is a measure of

the amount of information that one random variable contains about another random variable. It is the reduction in the uncertainty of one random variable due to the knowledge of the other.

The *relative entropy* between two probability mass functions $p(x)$ and $q(x)$ is defined as

$$D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = E_p \log \frac{p(X)}{q(X)} \tag{2.9}$$

where is used the convention that $0 \log \frac{0}{0} = 0$ and the convention (based on continuity arguments) that $0 \log \frac{0}{q} = 0$ and $p \log \frac{p}{0} = \infty$. Thus, if there is any symbol $x \in \mathcal{X}$ such that $p(x) > 0$ and $q(x) = 0$, then $D(p \parallel q) = \infty$. The relative entropy is always nonnegative and is zero if and only if $p = q$. However, it is not a true distance between distributions since it is not symmetric and does not satisfy the triangle inequality. Nonetheless, it is often useful to think of relative entropy as a "distance" between distributions.

Given two random variables X and Y with a joint probability mass function $p(x, y)$ and marginal probability mass functions $p(x)$ and $p(y)$, the *mutual information* $I(X;Y)$ is the relative entropy between the joint distribution and the product distribution $p(x)p(y)$:

$$\begin{aligned} I(X;\, Y) &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \\ &= D(p(x, y) \parallel p(x)p(y)) \\ &= E_{p(x,y)} \log \frac{p(X, Y)}{p(X)p(Y)} \end{aligned} \tag{2.10}$$

The mutual information $I(X;\, Y)$ can be rewritten in function of

entropy.

$$\begin{aligned} I(X;\, Y) &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\ &= \sum_{x,y} p(x,y) \log \frac{p(x \mid y)}{p(x)} \\ &= -\sum_{x,y} p(x,y) \log p(x) + \sum_{x,y} p(x,y) \log p(x \mid y) \\ &= -\sum_{x} p(x) \log p(x) - \big(-\sum_{x,y} p(x,y) \log p(x \mid y) \big) \\ &= H(X) – H(X \mid Y) \end{aligned} \tag{2.11}$$

Thus, the mutual information $I(X;\, Y)$ is the reduction in the uncertainty of X due to the knowledge of Y.

By symmetry, it also follows that

$$I(X;\, Y) = H(Y) – H(Y \mid X) \tag{2.12}$$

Thus, X says as much about Y as Y says about X.

Since $H(X,Y) = H(X) + H(Y \mid X)$ the mutual information is simplified as:

$$I(X;\, Y) = H(X) + H(Y) – H(X,\, Y) \tag{2.13}$$

Figure 2.3 shows the relationship between entropy $H(X), H(Y)$ and mutual information $I(X;\, Y)$ of a pair of random variables X and Y.

Although the concept of Shannon entropy was originally developed to represent communication over a noisy channel, it readily extends to exploration, which is closely related to coverage.

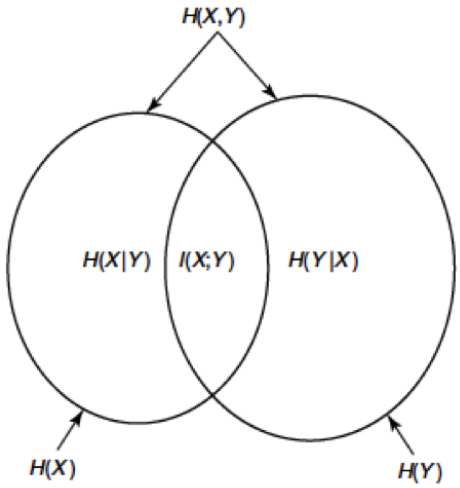


Figure 2.3: *Relationship between entropy and mutual information. (Figure appears in [21])*

2.1.2 Renyi's Entropy

The Hungarian mathematician Alfred Renyi [56] proposed in the 60s a new information measure generalizing the Shannon entropy. The Renyi's entropy for a discrete random variable X with possible outcomes $1, 2, \ldots, n$ and corresponding probabilities $p_i = P(X = i), \quad i = 1, \ldots, n$ is defined as:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \Big(\sum_{i=1}^{n} p_i^\alpha \Big) \quad \alpha > 0 \;\; \alpha \neq 1, \tag{2.14}$$

Similarly for continuous random variable X, with probability density function $p(x)$, the Renyi's entropy is defined as:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \Big(\int p^\alpha(x) dx \Big) \quad \alpha > 0 \;\; \alpha \neq 1, \tag{2.15}$$

As is easily seen with following limiting:

$$\lim_{\alpha \to 1} H_{alpha}(X) = -\sum_{i=1}^{n} p_i \log p_i \tag{2.16}$$

the Shannon's entropy is a special case of the Renyi's entropy.

The Renyi's entropy has the following properties:

- $H_\alpha(X)$ is nonnegative: $H_\alpha(X) \geq 0$.
- $H_\alpha(X)$ is decisive: $H_\alpha(0,\, 1) \;=\; H_\alpha(1,\, 0)$.
- For $\alpha \leq 1$ Renyi's entropy is concave. For $\alpha > 1$ Renyi's entropy in not pure convex nor pure concave. It looses concavity for $\alpha > \alpha^* > 1$ where α^* depends on N and obeys the relation $\alpha^* \leq 1 + \frac{\ln(4)}{\ln(N-1)}$.
- Since $\frac{\alpha-1}{\alpha} H_\alpha(X) \leq \frac{\beta-1}{\beta} H_\beta(X)$ for $\alpha \leq \beta$, $(\alpha - 1)H_\alpha(X)$ is a concave function of X.
- $H_\alpha(X)$ is bounded, continuous and non-increasing function of α.
- Renyi's entropies for different α are correlated.
- The following is a simple but not very sharp bound on Shannon entropy $(H_S(X))$ of any probability mass function $H_2(X) \leq H_S(X) \leq \ln N + \frac{1}{N} - \exp(-H_2(X))$.
- $H_Z(X)$ with $z = \alpha + jw$ is analytic in all the complex plane except the negative real axis.

In this thesis the Renyi's entropy will be used with $\alpha = 2$, called the quadratic entropy. The quadratic entropy for discrete random variable is defined as:

$$H_2(X) = -\log\big(\sum_{i=1}^{n} p_i^2\big) \tag{2.17}$$

Instead, the Renyi's quadratic entropy is defined for continuous random variable as:

$$H_2(X) = -\log\big(\int p^2(x)dx\big) \tag{2.18}$$

2.1.3 Parzen theory

The Renyi's quadratic entropy is defined using the pdf for continuous random variable or the pmf for discrete random variable. The pdf can be estimated by Parzen density estimation method [50] starting from a samples set, which is the realization of the random variable. The Parzen's fundamental idea is to place a kernel function over a samples set and sum with proper normalization.

$$\widehat{p}_X(x) = \frac{1}{M\,h}\sum_{i=1}^{M}\kappa\left(\frac{x - x_i}{h}\right) \tag{2.19}$$

where κ is the kernel function. A kernel function κ satisfies the following properties:

$$\begin{aligned} \sup_{\Re} \mid \kappa(x) \mid &< \infty \\ \int_{\Re} \mid \kappa(x) \mid dx &< \infty \\ \lim_{x \to \infty} \mid x\kappa(x) \mid &= 0 \\ \kappa(x) \geq 0, \quad \int_{\Re} \kappa(x)dx &= 1 \end{aligned} \tag{2.20}$$

Some examples of kernel functions are given in Table 2.1.

$\kappa(x)$
$\frac{1}{2}, \quad \mid x \mid \leq 1$ $0, \quad \mid x \mid \geq 1$
$1- \mid x \mid, \quad \mid x \mid \leq 1$ $0, \quad \mid x \mid \geq 1$
$\frac{4}{3} - 8x^2 + 8 \mid x \mid^3, \quad \mid x \mid < \frac{1}{2}$ $\frac{8}{3}(1- \mid x \mid)^3, \quad \frac{1}{2} \leq \mid x \mid \leq 1$ $0, \quad \mid y \mid > 1$
$\frac{\exp -\frac{1}{2}x^2}{\sqrt{2\pi}}$
$\frac{1}{2} \exp - \mid x \mid$
$(\frac{1}{\pi})(1 + x^2)^{-1}$
$\frac{1}{2\pi}\left(\frac{\sin(x/2)}{x/2}\right)^2$

Table 2.1: *Summary of kernel functions*

The kernel width h, or Bandwidth, in 2.19 remains to define after a kernel function is chosen. The optimal Bandwidth h is a crucial problem because with large h the estimated density will oversmooth and mask the structure of the data while with small h the estimated density will be spiky and very hard to interpret. The statistical approach leads to determine the bandwidth through the minimization of mean integrated squared error (MISE) [59].

$$h_{MISE} = \text{argmin}\left\{E\Big[\int(\widehat{p}_X(x) – p(x))^2 dx\Big]\right\} \tag{2.21}$$

For example, the h^* for Gaussian kernels has been proposed [59] for estimating normally distributed data with unit covariance

$$h^* = 1.06 \cdot \sigma \cdot M^{-\frac{1}{5}} \tag{2.22}$$

where, σ is the sample standard deviation and M is the quantity of data. However using a robust measure of the spread instead of the sample variance, and reducing the coefficient 1.06 to better cope with multimodal densities, a better result can be obtained. The optimal bandwidth then becomes

$$h^* = 0.9 \cdot \min(\sigma, \text{IQR}/1.34) \cdot M^{-\frac{1}{5}} \tag{2.23}$$

where, the interquartile range (IQR), is the difference between the 75th percentile (Q3) and the 25th percentile (Q1), of the generated data sample. A percentile rank is the proportion of samples in a distribution that a specific sample is greater than or equal to.

The Parzen theory can also be extended to p-dimensional sample data obtaining a multivariate kernel density estimator. Therefore,

given a set of M $p-$dimensional data x_n, $n = 1, \ldots, N$, the multivariate kernel density estimator, with the kernel function κ and a kernel width h, is defined as:

$$\widehat{p}_X(x) = \frac{1}{M\,h^p} \sum_{i=1}^{M} \kappa\left(\frac{1}{h}(x–x_i)\right) \tag{2.24}$$

where, M is the samples and h is the kernel variance often called Bandwidth. A popular choice for the kernel is the Gaussian pdf, which is a symmetric kernel with its value smoothly decaying away from the kernel center.

$$\kappa(x) = \frac{1}{\sqrt{2\pi}^p} \exp\left(-\frac{1}{2}(x^T\,x)\right) \tag{2.25}$$

In this thesis the multivariate kernel density estimator 2.24 with Gaussian kernel function 2.25 will define the a priori density map over the test case data of Tuscan Archipelago archaeological remnants.

2.2 Voronoi diagram and Equitable Power diagram

Ukrainian mathematician Georgy Fedosievych Voronyi (or Voronoy) has defined the general $n-$dimensional *Voronoi diagrams* in 1908 [27], [23]. Even though originally the Voronoi diagram dates back to Descartes in 1644. To date, Voronoi diagrams have been applied in several fields of science and technology, finding numerous practical and theoretical applications [8], [48]. Fejes Tóth basing on the

Voronoi diagram has also introduced a new diagram called Power diagram in 1977 [61] and then Aurenhammer has traced the definition of the power distance to the work of 19th-century mathematicians Edmond Laguerre and Georgy Voronoy in 1987 [7]. Finally, in the last years the Power diagram has been improved in the Equitable Power diagram [53], [54] designing novel algorithms for partitioning.

The Voronoi diagram is a partitioning of a plane into regions according to closeness of a set of points called seeds, sites or generators. In detail, after that the position of the seeds are defined each corresponding region is made up of the whole set of closest points. Instead, the Power diagram is a partitioning of a plane with the sizes of the regions depending on weights assigned to each seed beforehand. Then, the Power diagram can become *Equitable* if the weights are computed to achieve an equitable partition.

In the following the Voronoi diagram and the Power diagram are more formally described. Then the concept of Equitable Power diagram will be introduced.

2.2.1 Voronoi diagram

Assume that $G = (g_1, ..., g_m)$ is an ordered set of m-distinct points (called sites) in the plane $Q \in \Re^2$. For two distinct sites $g_i, g_j \in G$, the *dominance* of g_i over g_j is defined as the subset of the plane being at least as close to g_i as to g_j:

$$\mathsf{dom}(g_i, g_j) = \{ x \in Q \mid \delta(x, g_i) \leq \delta(x, g_j)\} \tag{2.26}$$

where, δ is the Euclidean distance function. Clearly, $\mathsf{dom}(g_i, g_j)$ is a closed half plane bounded by the perpendicular bisector of g_i and

g_j, that is

$$b(g_i,\, g_j) = \{x \in Q \;\mid\; \delta(x, g_i) = \delta(x, g_j)\} \tag{2.27}$$

This bisector separates all points of the plane closer to g_i from those closer to g_j and will be termed the separator of g_i and g_j. Thus the region of a site $g_i \in Q$ is the portion of the plane lying in all of the dominances of g_i over the remaining sites in Q. Formally

$$V_i(G) = \bigcap_{g_j \in G - g_i} \mathsf{dom}(\, g_i,\, g_j) \tag{2.28}$$

Notice each region is a convex polygon since it come from the intersection of $n-1$ half planes. Therefore, the boundary of a region consists of at most $n-1$ *edges* (maximal open straight-line segments) and *vertices* (their endpoints). Each point on an edge is equidistant from exactly two sites, and each vertex is equidistant from at least three. As a consequence, the regions are edge to edge and vertex to vertex forming a polygonal partition of the plane. This partition defines the Voronoi diagram $\mathcal{V}(G) = (\, V_1(G),\, \ldots\, ,\, V_m(G)\,)$ generated by points $G = (g_1, \ldots, g_m)$ (see Figure 2.4).

2.2.2 Power diagram

Basing on the Voronoi diagram the Power diagram is developed. Thus it is a generalization of the Voronoi diagram having strongest similarities with the original diagram. Although some differences exist. For instance, a region might be empty and one site might not be in its own region. Basically, the Power diagram is a partitioning of a plane as the Voronoi diagram where each site has assigned one weigh, that influences the location of the regions' edges.

2.2 Voronoi diagram and Equitable Power diagram

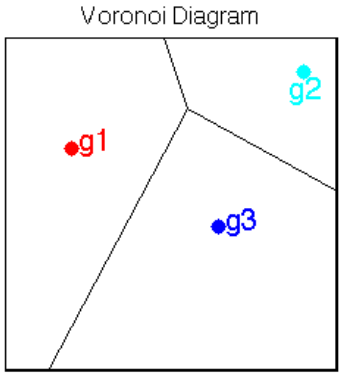


Figure 2.4: *Voronoi diagram for three seeds.*

Assume that $G = (g_1, ..., g_m)$ is an ordered set of m-distinct sites where each one has assigned an scalar weight $w_i \in \Re$. A new ordered set of distinct points G_W and a new distance $\mathsf{d_P}(\cdot)$ called power distance are defined

$$G_W = ((g_1,\ g_1),\ \ldots\ , (g_m,\ w_m)) \tag{2.29}$$

$$\mathsf{d_P}(x, g_i;\ w_i) = \delta(x,\ g_i) - w_i \tag{2.30}$$

Similarly to the Voronoi diagram, each region called *power cell* of the Power diagram $\mathcal{V}(G_W) = (\,V_1(G_W),\ \ldots\ ,\ V_n(G_W)\,)$ generated by *power* points $G_W = ((g_1,\ w_1)\ldots\ , (g_m,\ w_m))$ is defined by

$$V_i(G_W) = \bigcap_{g_j \in G_W - \{(g_i, w_i)\}} \mathsf{dom}(\ (g_i, w_i),\ (g_j, w_j)) \tag{2.31}$$

where the dominance of (g_i, w_i) over (g_j, w_j) is

$$\mathsf{dom}((g_i, w_i), (g_j, w_j)) = \{\ x\ \in Q\ \mid\ \delta(x, g_i) - w_i\ \leq\ \delta(x, g_j) - w_j\} \tag{2.32}$$

and the bisector of (g_i, w_i) and (g_j, w_j) is defined as

$$b\big(\ (g_i, w_i), (g_j, w_j)\ \big) = \{x \in Q \mid\ (g_j - g_i)^T x = \frac{1}{2}(\delta(g_j) - \delta(g_i) + w_i – w_j)\} \tag{2.33}$$

It worth to note that each power cell is a convex set and the bisector $b\big(\ (g_i, w_i), (g_j, w_j)\ \big)$ is a face orthogonal to the line segment $\overline{g_i g_j}$ and passing through the point g^*_{ij} given by

$$g^*_{ij} = \frac{\delta(g_j) - \delta(g_i) + w_i – w_j}{2\delta(g_j, g_i)}(g_j - g_i) \tag{2.34}$$

This last property will be important and crucial to compute the right weights following the prior proprieties requests for an equitable partitioning of the plane. Effectively it means that is possible to arbitrarily move the bisector modifying the weights while still preserving the orthogonality constraint. Figure 2.5 shows an example of Power diagram for three power points.

2.2.3 Equitable Power Diagram

Starting from a Power diagram and computed an equitable partitioning of the plane, the Equitable Power diagram is achieved. This diagram allow to regulate the applied resource allocation among *customers* balancing the assigned workload to each *resource* [53], [54]. *Customers* and *resources* depend on the application field. For instance, in surveillance and exploration missions the customers are specific points of interest to be surveyed over a region that is the *workspace* and the resources are the available agents (i.e. vehicles with sensors); in the transportation and distribution applications the

2.2 Voronoi diagram and Equitable Power diagram

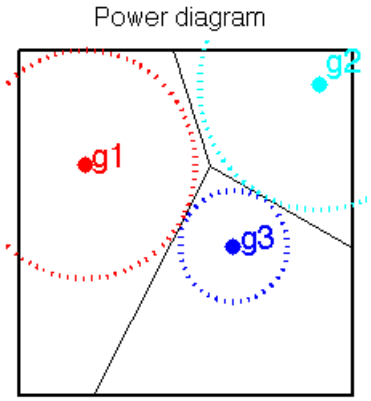


Figure 2.5: *Power diagram for three power points.*

customers are people demanding some service (e.g., utility repair) or goods and in logistics tasks they could be troops in the battlefield.

Recently the *equitable partitioning policy* has been even proposed in multi-agent systems to guarantee a workload-balancing among the agents, which refer resources [9], [10], [18], [41]. A partitioning policy is an algorithm that partitions a given workspace Q into m-subregions basing on the knowledge of the position and the other information of the m available agents. Then, each i-th agent is assigned to subregion Q_i and it gives services to customers in Q_i. Finally, modeling the *workload* over the space Q through a measure function $\lambda : Q \rightarrow \Re$, the *workload* for a region Q_i is $\lambda_{Q_i} = \int_{Q_i} \lambda(x)dx$ and a *equitable partition* of Q is $\lambda_{Q_i} = \lambda_{Q_j}$ for all $i, j \in (1, \ldots, m)$ or in equivalent mode $\lambda_{Q_i} = \lambda_Q/m$, for all i. Here only the concept of Equitable Power diagram has been presented but in chapter 4 an partitioning algorithm will be describe in more

detail because it will be a main step of the cooperative distributed algorithm for AUVs seabed survey, which is the core of the thesis. Notice the use of the Equitable Power diagram will allow to intend the proposed algorithm to be *cooperative*: each agent will negotiate with the others which portion of the neighboring area to explore.

2.3 Robotics motion planning

In robotics the motion planning and the trajectory planning are algorithms that define the sequence of robot configurations to achieve the high-level specifications of tasks [2].

In case of marine systems, the paths are represented by a set of waypoints to be visited. These waypoints are stored in a database and used for generation of trajectory or a path for moving underwater vehicle to follow. Moreover, weather routing, obstacle avoidance and mission planning can be incorporated in the design of waypoints. Waypoint database is generated using different criteria according to information on: mission, environmental data, geographical data, obstacles, collision avoidance and feasibility.

- Mission: the vehicle moves from the starting point to the goal point via the waypoints.
- Environmental data: information about wind, waves, currents are used to define the waypoints for obtaining optimal routes.
- Geographical data: geographical information such as the position of islands and shallow waters are included.

[2]The review presented here is based on the book by Steven M. LaValle [39].

- Obstacles: floating constructions and other obstacles are avoided.
- Collision avoidance: the vehicle moves avoiding collision with other vehicles by introducing safety margins on the routes.
- Feasibility: the waypoints are defined in order to the maneuvers are feasible.

Each waypoint can be also defined either one at a time in accordance with the current up-to-date information, or a priori before the motion.

In the next sections will be reported the available techniques in the literature and the most important reviews on path planning.

2.3.1 Problem Formulation

A few fundamental terms are introduced to formulate the path planning problem.

Workspace $\mathcal{W}$: *World* $\mathcal{W}$ is the physical world where the robot exists and it can be chosen by two modes: either a 2D world $\mathcal{W} = \Re^2$ or 3D world $\mathcal{W} = \Re^3$. Generally, the world contains two kinds of entities:

- Obstacles: Portions of the world that are "permanently" occupied, as an example the walls of a building.
- Robots: Bodies that are modelled geometrically and controlled via motion plan.

A method that uses a collection of primitives permits the systematically constructing representations of obstacles and robots. Both obstacles and robots are (closed) subsets of $\mathcal{W}$.

Configuration space ($\mathcal{C}$): The set of configurations that a robot can achieve. By this abstraction the same motion planning algorithm can be used for different problems. In this mode the path planning becomes a search on a space of transformations.
The configuration space $\mathcal{C}$ can be subdivided into free configuration space, $\mathcal{C}_{free}$ and obstacles configuration space, $\mathcal{C}_{obs}$. The configurations in $\mathcal{C}_{obs}$ result in contact with an obstacle, while those in $\mathcal{C}_{free}$ are not. Formally, given world $\mathcal{W}$, a closed obstacle region $\mathcal{O} \subset \mathcal{W}$, closed robot $\mathcal{A}$, and configuration space $\mathcal{C}$. Let $\mathcal{A}(q) \subset \mathcal{W}$ denote the placement of the robot into configuration q. The obstacles configuration space $\mathcal{C}_{obs}$ in $\mathcal{C}$ is defined as:

$$\mathcal{C}_{obs} = \{q \in \mathcal{C} \mid \mathcal{A}(q) \cap \mathcal{O} \neq \emptyset\} \tag{2.35}$$

Instead the free configuration space $\mathcal{C}_{free}$ is an open subset of $\mathcal{C}$:

$$\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs} \tag{2.36}$$

Path ($\tau(s)$): Curve through $\mathcal{C}_{free}$ parameterized by s.

Degrees of Freedom: The minimum number of independent variables required to represent the robot configuration. In this thesis, the focus is on autonomous underwater vehicles AUVs, which require six degrees of freedom for a full characterization: $q = \{x, y, z, \phi, \theta, \psi\}$, where $x,\ y,\ z$ refer to the position in 3D space, and $\phi,\ \theta,\ \psi$ to the Euler angles so the orientation. Moreover the planning for an AUV can be simplified assuming that $\theta = \theta_{ref}$, $\phi = \phi_{ref}$, $z = z_{ref}$, reducing the representation with three degrees of freedom: $q = x, y, \psi$.

Given robot $\mathcal{A}$ and obstacle $\mathcal{O}$ models, C-space $\mathcal{C}$, a start location $q_I \doteq \{x_I, y_I, z_I, \phi_I, \theta_I, \psi_I\} \in \mathcal{C}_{free}$ and a goal location $q_G \doteq \{x_G, y_G, z_G, \phi_G, \theta_G, \psi_G\} \in C_{free}$. According to the fundamental terms

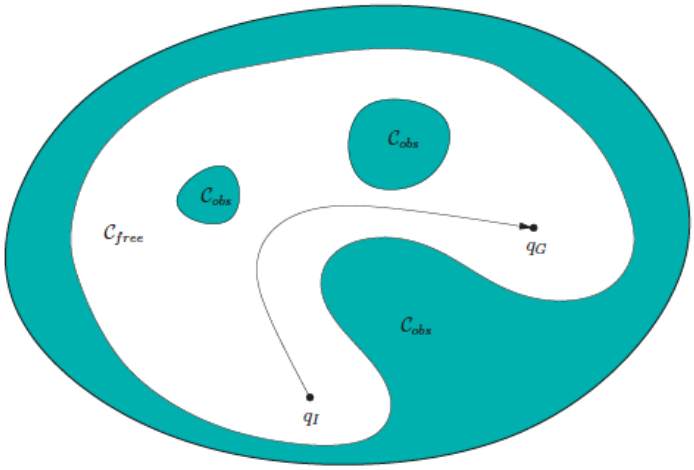


Figure 2.6: *The representation of basic motion planning problem by $\mathcal{C}$-space ideas. (Figure appears in [39])*

the basic motion planning problem is defined as an algorithm that computes a (continuous) *path*, $\tau : [0,1] \rightarrow C_{free}$, such that $\tau(0) = q_I$ and $\tau(1) = q_G$, or correctly report that such a path does not exist. The problem is conceptually illustrated in Figure 2.6.

The motion planning problem, above formulated, is addressed through two main philosophies, *sampling-based motion planning* and *combinatorial motion planning*. In sampling-based motion planning the main idea is to avoid the explicit construction of $\mathcal{C}_{obs}$ and conducting a search that probes the $\mathcal{C}$-space with a sampling scheme by a collision detection module as a "black box". This approach allows to develop planning algorithms independent from the particular geometric models. Instead, the combinatorial motion planning philosophy finds paths through the continuous configuration space

without resorting to approximations. This approach is alternatively referred to as *exact* algorithm, due to this last property. Therefore it is in contrast to the sampling-based motion planning algorithm.

2.3.2 Combinatorial motion planning

The combinatorial approach develops elegant and efficient algorithms for many special classes. These algorithms are complete and do not depend on approximations, so if a solution exist they find one, otherwise they report failure. They also provide theoretical upper bounds on the time needed to solve motion planning problems.

In the combinatorial approach before to define an algorithm is constructed a roadmap along the way to solving queries. *Roadmap* is a topological graph G that maps in $\mathcal{C}_{free}$ with two conditions:

- **Accessibility:** from anywhere in C_{free} it is trivial to compute a path that reaches at least one point along any edge in G.
- **Connectivity-preserving:** If there exists a path through C_{free} from q_I to q_G, then there must also exist one that travels through G.

A topological graph is a graph where, every vertex q corresponds to a point in $\mathcal{C}_{free}$ (topological space) and every edge corresponds to a continuous, injective (one-to-one) function, $\tau : [0, 1] \rightarrow \mathcal{C}_{free}$. The image of τ connects the point in $\mathcal{C}_{free}$ that corresponds to the endpoints (vertices) of the edge. Therefore the roadmap $G(V, E)$ represent a topological graph where V is a set of robot configurations (vertices) and E (edges) is the set of paths that map into C_{free}.
A roadmap is obtained in two modes, either derived from a cell

decomposition of $\mathcal{C}_{free}$ or directly constructed without considering the cells. In the first case, a cell decomposition algorithm partitions the free configuration space C_{free} into a finite set of regions, called cells, satisfying three proprieties:

- Computing a path from one point to another inside of a cell must be trivially easy, i.e. if every cell is convex, then any pair of points in a cell can be connected by a line segment.

- Adjacency information for the cells can be easily extracted to build the roadmap.

- For a given q_I and q_G, it should be efficient to determine which cells contain them.

Thus, the motion planning problem is reduced to a graph search problem. In the second case, instead the partition in cell is not done and the roadmap is constructed directly.

Several cell decomposition algorithms and two approaches to construct a roadmap directly (called *shortest path* and *maximum clearance*) are defined in the literature, see the review in [12], [39], [22], [24], [19], [37]. For the sake of clarity, Figure 2.7 depicts some examples of cell decompositions and Figures 2.8 show roadmaps constructed directly.

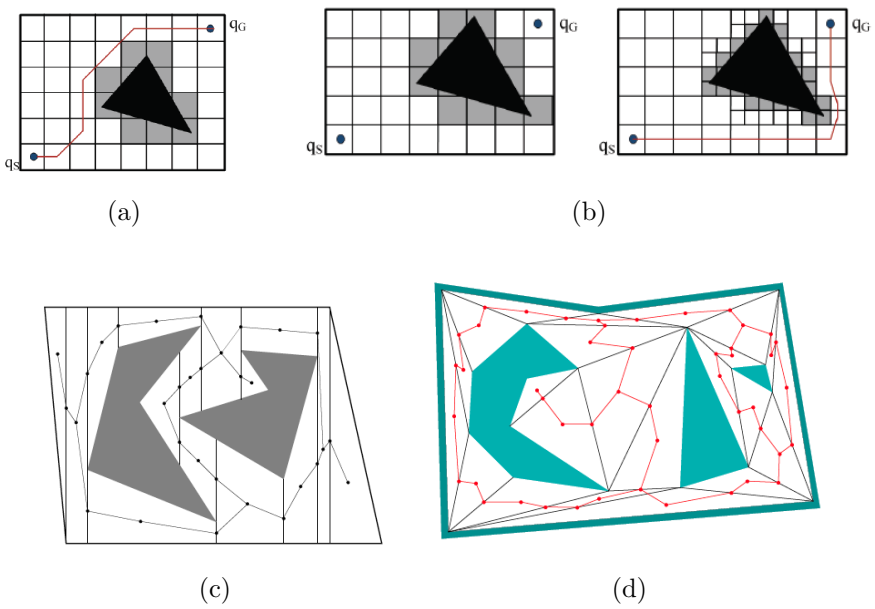
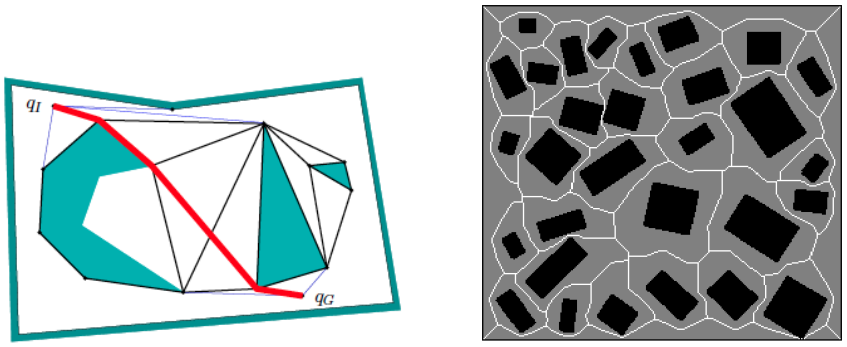


Figure 2.7: *(a) The approximate cell decomposition and the path found into the cells not marked. (b) Example of the MIXED cell subdivision (c) The vertical cell decomposition method and the roadmap derived. (d) A triangulation decomposition of $\mathcal{C}_{free}$ and the roadmap obtained.*



(a) (b)

Figure 2.8: *(a) The shortest path roadmap and the shortest path between $q_I - q_G$ obtained. (b) The maximum-clearance roadmap built by a generalized Voronoi diagram (GVD) (Figure appears in [51]). Each point in the GVD is equidistant from at least the two nearest obstacles.*

2.3.3 Sampling-based motion planning

Before detailing the incremental sampling and searching methods the general sampling concepts and the collision detection module that they rely on, are defined.

Sampling based motion planning algorithms extract a countable number of samples from the state space $\mathcal{C}$ which is uncountably infinite. If the algorithm runs forever, this may be countably infinite, but in practice it terminates after a finite number of samples. The type of sequence with which the samples are chosen to be sampled the state space $\mathcal{C}$ influences the performance of sampling based planning algorithms and this mismatch between the infinite sampling sequence and the uncountable $\mathcal{C}$-space leads to the concept of dense-

ness and it motivates careful consideration of sampling techniques.

Denseness Given U and V two subsets of a topological spaces, the set U is *dense* in V if the closure of U is V, $\mathrm{cl}(U) = V$. For example $(0, 1)$ is dense in $[0, 1]$, $\mathbb{Q}$ is dense in $\mathbb{R}$.

Random sequence is probably dense Simply, a dense sequence is obtained by choosing points at random. The purpose is to have a dense sequence in probability of samples in $\mathcal{C}-$space. For instance, given $\mathcal{C} = [0, 1]$, let $l = [a, b] \subset \mathcal{C}$ with $b - a = l$. Consider a sequence of k independent random samples, the probability that no one of the samples falls into I is $p = (1 - l)^k$. When the number of samples tends to infinity the probability p tends to 0. Hence, the probability that any nonzero length interval contains no point converges to zero. In other words, the infinite sequence of samples is dense in $\mathcal{C}$ with probability 1.

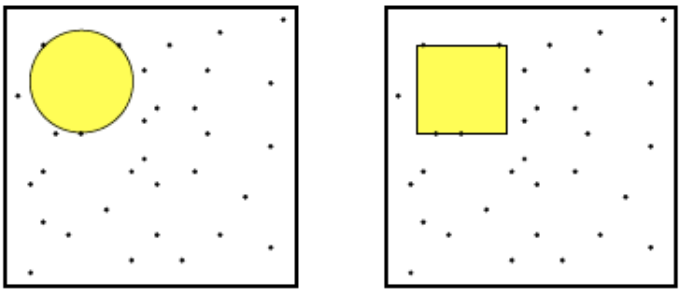
Random Samples The goal is to generate uniform random samples, i.e. to determine a uniform probability density function on the $\mathcal{C}-$space. The random sampling is the easiest of all sampling methods for the $\mathcal{C}-$space because it often consists of Cartesian product. If a uniform sample is taken from X_1 and X_2 the uniformity is obtained also for $X_1 \times X_2$. Therefore, for example given 5 robots with translational movements in $[0, 1]^2$ the $\mathcal{C}-$space is $\mathcal{C} = [0, 1]^{10}$.

Low-dispersion Sampling The Low-dispersion sampling is an alternative to random sampling. It places samples in a way that makes the largest uncovered area be as small as possible by generalizing of the idea of grid resolution. For a grid, the resolution can be increased by decreasing the step size of each axis. A possible extension of this concept is the criterion of *dispersion*:

Definition 1. *In a metric space (X, ρ) the dispersion of a finite set*

P of samples is

$$\delta(P) = \sup_{x \in X} \{\min_{p \in P} \{\rho(x, p)\}\}$$



(a) L_2 dispersion (b) L_∞ dispersion

Figure 2.9: *Interpretation of the dispersion definition with two different metrics. (Figure appears in [39])*

Once the sampling component has been defined, before to construct the solution to the original motion-planning problem is necessary to check if a configuration is in collision. Therefore a collision checking module, providing information about feasibility of candidate trajectory, is crucial since it will also take the largest amount of time in the planning algorithm. Even though it is often treated as a black box, it is important to study its inner workings to understand the information it provides and its associated computational cost. Several collision detection algorithms exist for different applications.

In case of $2D$ convex robot and convex obstacles where the model of $\mathcal{C}_{obs}$ can be determined, is used a *logical predicate* $\phi : \mathcal{C} \rightarrow T, F$ with $T = true$ and $F = false$, where if $q \in \mathcal{C}_{obs}$ then $\phi(q) = T$

otherwise $\phi(q) = F$. The logical predicate is easily implemented based on the available model. However, it is not sufficient in some cases, for example the logical predicate is a boolean function and it does not provide any information on how far the robot is from the obstacle. Hence, in this case a *distance function* $d : \mathcal{C} \to [0, +\infty)$ between q and the closed point in $\mathcal{O}$ is preferred. Moreover, the collision detection for a robot with m links and an obstacle set $\mathcal{O}$ with k connected components is more difficult then previous cases. It is faced with a two-phase approach. In the first phase defined *broad* the computation for bodies that are far away from each other is avoided. A bounding–box is placed around the objects and overlapping between bounding–boxes is easily checked. In the second *narrow* phase instead individual pairs of probably closer bodies are checked. Finally, for the case of nonconvex bodies the collision detection is done with a *hierarchical* approach which decomposes a body into a set of bounding boxes (see Figure 2.10). Such boxes may be as tight as possible around the part of the body or may be as simple as possible so that intersection test is easy.

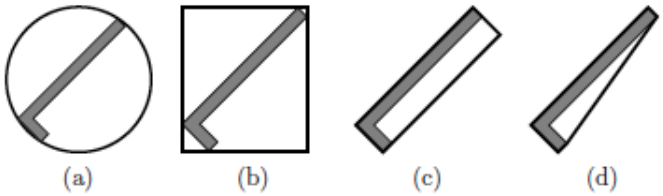


Figure 2.10: *Four different kinds of bounding regions: (a) sphere, (b) axis-aligned bounding box, (c) oriented bounding box, and (d) convex hull. (Figure appears in [39])*

2.3 Robotics motion planning

Motion planning algorithm also require that an entire path is in $\mathcal{C}_{free}$. The check cannot be done point-by-point because it would require an uncountably infinite number of collision detection tests. Hence, suppose a parametrization of the path $\tau : [0,1] \rightarrow \mathcal{C}$ a possible solution, for checking if $\tau([0,1]) \subset \mathcal{C}_{free}$, is to determine a resolution $\triangle_q$ in $\mathcal{C}_{free}$ and call the collision checker only on the samples. The resolution $\triangle_q$ induce a step size $t_2 - t_1$ where $t_1, t_2 \in [0,1]$ and $\rho(\tau(t_1), \tau(t_2)) \leq \triangle_q$ where $\rho(\cdot)$ is a metric on $\mathcal{C}$. Note that if the $\triangle_q$ is too small an high computational times is obtained and on the other hand, if it too large some miss collisions may happened. Therefore the choice of the resolution in $[0,1]$ may lead to a no efficient resolution in $\mathcal{C}$ or to a collisions missing.

Once the sampling component and the collision checking module have been defined, the incremental sampling and searching algorithms are developed. Such algorithms are strikingly similar to the family of search algorithms on discrete optimization. The main difference is that in search algorithm on discrete optimization the edges represent control actions while in incremental sampling and searching algorithm is constructed a topological graph where the edges are path segments. The sampling-based planning algorithms can be synthetized as follows:

1. **Initialization:** consider a graph $\mathcal{G}(V,E)$ with $E = \emptyset$ and V contains at least q_I and q_G (and possibly other points in $\mathcal{C}_{free}$).

2. **Vertex Selection Method (VSM):** Choose vertex $q_{cur} \in V$ to expand the graph.

3. **Local Planning Method (LPM):** For some $q_{new} \in \mathcal{C}_{free}$ try

to construct a path $\tau : [0,1] \rightarrow \mathcal{C}_{free}$ with $\tau(0) = q_{cur}$ and $\tau(1) = q_{new}$. If a collision occurs along τ go to step 2.

4. **Insert edges and nodes to the graph:** Insert τ in E and if not already in V insert q_{new} in V.

5. **Check for a solution:** Check if in $\mathcal{G}$ there is the desired path.

6. **Iterate:** Iterate until a solution is found or termination conditions are met.

A large family of sampling-based algorithms can be described by varying the implementations of steps 2 and 3, where the algorithms are based on the number of search trees. For these algorithms similarity to algorithms for graph explorations can be used unidirectional (single-tree), bidirectional (two-trees) or multi-directional (more than two trees) methods [39]. Bidirectional and multi-directional methods are useful in case of complex spaces with "traps" but are more difficult to manage. A bidirectional method defines the vertex selection method to alternate between trees when selecting vertices while, a multi-directional method selects pair of trees for connection.

Another method that can also be used for the sample and searching scheme is the Grid search algorithm [39]. The basic idea is to discretize each dimension of the $\mathcal{C}$–space obtaining $k-$neighbourhood with $k \geq 1$. The algorithm start searching the closest $1-$neighbourhood (or $k-$neighbourhood) that are in $\mathcal{C}_{free}$.

Arguably, to date the most influential algorithms for incremental sampling and searching are Probabilistic RoadMaps (PRMs) [33], [34] and Rapidly-exploring Random Trees (RRTs) [35], [38], [39]. These two algorithms differ in the way that they construct the graphs

connecting the points sampled randomly from the state space. The PRM is designed such that the roadmap building and the query phases of operation are separate; the entire roadmap is built before a path is found from start to goal. The challenging aspects of this type of algorithm are: 1) determining whether a randomly generated sample lies in the free configuration space at all, and 2) determining whether the edge between two nodes remains in the free configuration space. An implementation of the PRM planner is given in Algorithm 1. Instead the RRT find a path to the goal as samples are generated.

Algorithm 1: Probabilistic RoadMap

Input : Map, $\mathcal{C}$
Output: Probabilistic RoadMap PRM

```
1 Add q_start and q_goal to PRM;
2 while No path exists from q_start to q_goal in PRM do
3     Randomly generate q from C;
4     if q ∈ C_free then
5         Add q to PRM;
6         forall the node n ∈ PRM with n ≠ q do
7             if Good path exists from n to q then
8                 Add edge in PRM between n and q;
```

More details of the PRMs and RRTs algorithms can be found in [32].

2.4 Side-Scan Sonar - Multi Beam Echo Sounders

Underwater applications in area mapping and/or area surveillance are conducted by acoustic looking sensors: either side-scan sonar (SSS) or multi-beam echo sounder (MBES). The images of seabed are generated by SSS and the seabed bathymetry is determined by MBES.

2.4.1 Side-Scan-Sonar

To date, two types of Side Scan Sonar sensors [11], [43], [25] have been developed, either as payload for AUV or to be vessel-towed and installed on a standard Towfish. In the first case the side-scan transducers are on-board the AUV and gather data as the AUV moves forward in rectilinear motion. In the second case instead the side-scan transducers is placed in a "towfish" and pulled by a "tow cable" by a surface vessel, which performs regular lawn-mower transects paths. In both cases the side-scan transducer leaves a narrow channel not scanned directly beneath itself. Figure 2.11 shows an AUV path and corresponding SSS coverage swath and Figure 2.12 shows a vessel with the towfish towed. Side Scan Sonar systems use high frequency acoustic pulses in the range 100-600 kHz, which result in strong reflection from seabed features and rapid attenuation of energy transmitted into the seabed. Therefore, features located either on or above the seabed cause the reflections detected. The interpretation of reflections detected takes two main aspects: detection of vertical offsets and recognition of textural differences.

2.4 Side-Scan Sonar - Multi Beam Echo Sounders

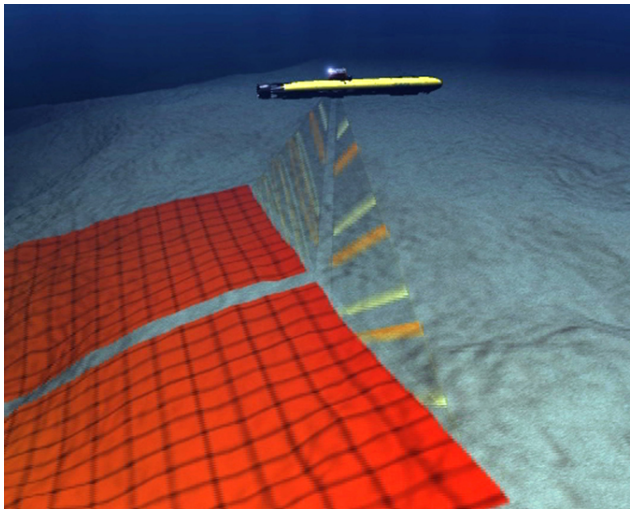


Figure 2.11: *An example of AUV trajectory and corresponding area covered by its SSS.*

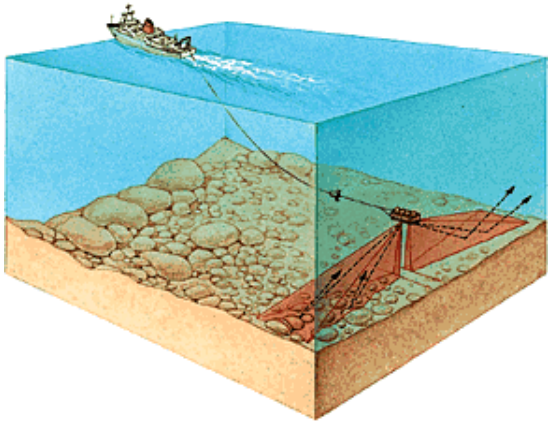


Figure 2.12: *Above the seabed a towfish is towed by a vessel on the surface.*

Vertical offsets are identified by discrete strong reflections (seabed has stepped upward) and shadow zones (seabed has dropped down); combinations of these effects facilitate identification of troughs (e.g. dredge marks, scour, channels) and ridges (sand bars, man-made features). Objects such as mines, buoys, and wrecks also give rise to distinctive, characteristic images. Changes in seabed material grain size and composition result in differing acoustic backscatter patterns. However, in the late 1990s the advent of digital acquisition and accurate positioning systems has developed modern sonars that has used image processing, real-time data mosaic and pattern recognition. Figure 2.13 shows an example of image obtained by SSS.

Finally, the resolution is divided into range (track-perpendicular) and transverse (track-parallel) components. Range resolution is a function of pulse frequency content, although transverse resolution is defined by beam width, determined by transceiver geometry and pulse content. Modern systems may employ either simple pulses or swept frequency ('Chirp') signals, which offer improved resolution at the expense of repetition rates.

2.4.2 Multi Beam Echo Sounders

The bathymetry measurements and the nature of seabed are achieved through two different technologies, beamforming and interferometric or phase discrimination sonars, but both the techniques give the same results.

The multi beam echo sounders collect bathymetric soundings in a swath perpendicular to the ship track by transmitting a broad acous-

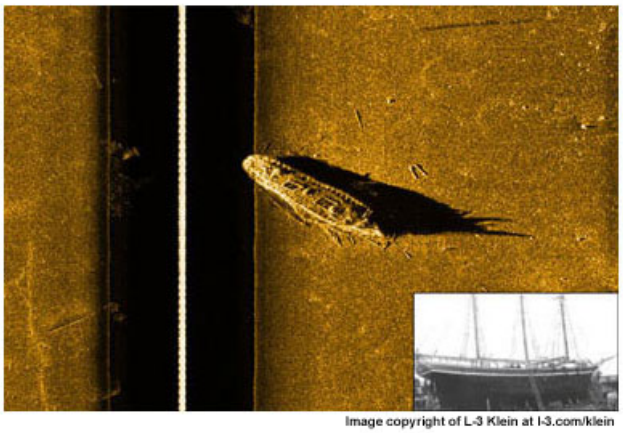


Figure 2.13: *Shipwrecks from Estonia: image obtained with the Klein System 5000 by the Estonian Navy near Tallinn Estonia - Polaris/Raa - The sailing vessel was built in 1917 in Koivisto, Finland. She was 32 meters long, 9 meters wide and the waterline was at 3,6 meters. She was originally named Polaris until the year 1938, when she was sold to Estonia and renamed Raa. She sunk in 1941.*

tic fan shaped pulse from a specially designed transducer across the full swath across-track. In reception it is possible to form multiple narrow receiver beams (beamforming) in the across-track direction (around 1 degree depending on the system). From these narrow beams a two way travel time of the acoustic pulse is established utilizing a bottom detection algorithm. The algorithm requires also an accurate measurement of the sonar's motion relative to a Cartesian coordinate system to determine the transmit and the receive angle for each beam. Typically the measures of motion are: heave, pitch, roll, yaw, and heading. Furthermore, the depth and position of the

return beams can be adjusted if the speed of sound in water is known for the full water column profile [30]. Various transmit frequencies are utilized by different MBES systems depending on the sea floor depth. Low frequencies (12 kHz) systems can collect swath soundings at full ocean depths, many up to 10.000 meters, while high frequency systems (300 kHz and more) collect swath bathymetry in depths of 20 meters or less.

In contrast, the interferometric technique uses the phase content of the sonar signal to measure the angle of a wave front returned from a sonar target. When backscattered sound energy is received back at the transducer, the angle the return ray makes with the transducer is measured. The range in the corresponding direction is calculated from the two-way travel time of the ray. The angle is determined by knowing the spacing between elements within the transducer, the phase difference of the incoming wave front, and the wavelength [42].

2.5 Autonomous Underwater Vehicles Localization

The estimate of autonomous vehicle's absolute position is defined as localization problem. The guidance system and/or the post-processing of data gathered through sensor need measures of absolute position to compute their own outputs.

Most autonomous system rely on radio frequency communications and global positioning system (D)GPS on surface. However, underwater such signals propagate only short distances and acoustic

based sensors and communications perform better. Therefore an Unmanned Underwater Vehicle (UUV) localization system relies on two different techniques, either acoustic transponders and modems or inertial navigation. In the first technique, the localization is based on measuring the time-of-flight (TOF) of signals from acoustic beacons or modems. In the second technique, instead are used accelerometers and gyroscopes for increased accuracy to propagate the current state. Nevertheless, this method has position error growth that is unbounded.

The type of localization systems depend on the characteristics of mission and different systems can be combined improving the performance.

2.5.1 Acoustic Localization

The acoustic positioning systems measure by the time-of-flight TOF of acoustic signals the positions relative to a framework of active sensing elements, which must be deployed prior to operations. These systems are categorized into three broad classes based on the baseline, which is the distance between the active sensing elements [63]:

- Ultra-Short Baseline (USBL), the baseline length is shorter than 10cm. See Figure 2.14b.
- Short Baseline (SBL), the baseline length is between 20mt to 50mt. See Figure 2.14a.
- Long Baseline (LBL), the baseline length is between 100mt to 6000mt. See Figure 2.14c.

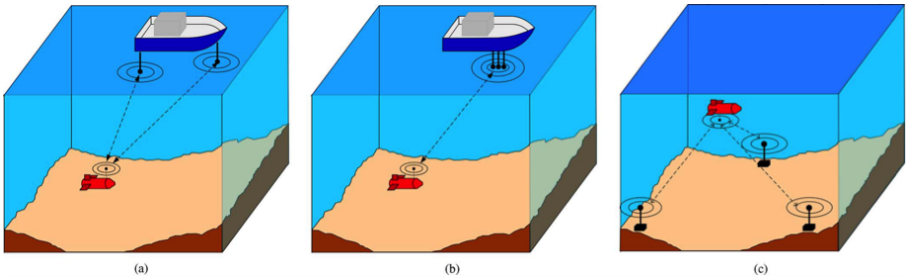


Figure 2.14: *(a) Short baseline SBL (b) Ultra-short baseline USBL (c) Long baseline LBL (Figure appears in [52])*

- Single Fixed Beacon, the localization is performed from only one fixed beacon.
- Acoustic Modem, new techniques are developed using acoustic modems. Beacons no longer have to be stationary.

USBL system measures phase comparison on an arriving ping between individual elements within a multi-element (≥ 3) transducer. This phase comparison is used to determine the bearing from USBL transceiver to a beacon. The transceivers are placed at either end of the ship hull. The USBL system works in pinger, responder, or transponder mode. Any range and bearing (position) derived from a USBL system is with respect to the transceiver mounted to the vessel and as such a USBL system needs a Vertical Reference Unit (VRU), a Gyro, and possibly a surface navigation system to provide a position that is seafloor referenced.
The main advantages of USBL system are:

- Low system complexity and easy tool to use.
- Not need to deploy transponders on the seabed.

- Good range accuracy with time of flight system.

The disadvantages of USBL system are:

- Detailed calibration of system required.
- Absolute position accuracy depends on additional sensors that are on board of the ship.
- Minimal redundancy.
- Large transceiver/transducer gate valve or pole required with a high degree of repeatability of alignment.

SBL system derives a bearing to a beacon from multiple (≥ 3) surface mounted transceivers. This bearing is derived from the detection of the relative time of arrival as a ping passes each of the transceivers. The transceivers are placed at opposite ends of a ship's hull and so the baseline depends on the size of the support ship. Any range and bearing (position) are based on the transceivers and as such a SBL system needs a Vertical Reference Unit (VRU), a Gyro, and possibly a surface navigation system to provide a position that is seafloor (earth) referenced. A SBL system works in pinger, responder or transponder mode.
The main advantages of SBL system are:

- Low system complexity and easy tool to use.
- Good range accuracy with time of flight system.
- Spatial redundancy built in.
- Not need to deploy transponders on the seabed.

- Small transducers/gate values.

The disadvantages of SBL system are:

- Detailed offshore calibration of system required.
- Large baselines for accuracy in deep water ($\geq$ 40mt) needed.
- Very good dry dock/structure calibration required.
- Absolute position accuracy depends on additional sensors that are on board of the ship.
- $\geq$ 3 transceivers deployment poles/machines needed.

LBL system derives a position basing on an array of transponders deployed on the seabed but over a wide mission area. The localization is based on triangulation of three or more time of flight of acoustic signals to/from the seafloor stations. LBL system works in responder or transponder mode and any range/range position is based on relative or absolute seafloor coordinates. As such the LBL system does not require a VRU or GYRO. Other diverse implementation of LBL system is with GPS intelligent buoys where the beacons are installed at the surface rather than on the seabed. Therefore GPS intelligent buoys system reduces the installation costs and the need for recovery of the beacons.
The advantages of LBL system are:

- Very good position accuracy independent of water depth.
- Observation redundancy.
- Provide high relative accuracy positioning over large areas.

- Small transducer

The disadvantages of LBL system are:

- Complex system requiring expert operators.
- Large arrays of expensive equipment.
- Operational time consumed for deployment/recovery.
- Conventional systems require comprehensive calibration at each deployment.

The major detractor of LBL system is the cost and time required to install the beacons and to geo-reference them. This is reduced by the use of only a single fixed beacon instead of a network of them where, the baseline is synthesized by propagating the ranges from a single beacon forward in time until the next update is received. This technique has been simulated on real world data in [36] and defined "virtual LBL"[49], [6]. For instance, Figure 2.15 shows a visual representation of single beacon navigation. In the figure the vehicle receives three acoustic pings and knows the beacon location a priori. Each time, reception of a ping results in a reduction of uncertainty in the direction of the beacon.

Recent fast technological developments have induced a new localization system based on the acoustic modems. Acoustic modems allow simultaneous communication of small packets and ranging based on TOF. Moreover the receiver can bind its position to a sphere centered on the transmitter if the position of the transmitter is included in the communicated information. So the operations of localization and fixing for beacons prior to the mission is no longer necessary.

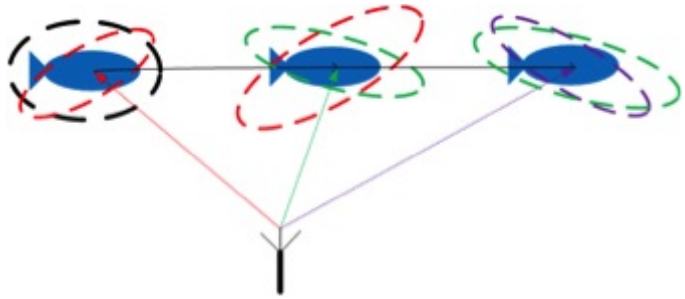


Figure 2.15: *An example of the localization system with single fixed beacon at known location is showed. Uncertainty grows in between updates from the beacon. On reception of an update, uncertainty is reduced in the dimension coinciding with the location of the beacon. (Figure appears in [52])*

The main producers of acoustic modems are: Aquatec, Woods Hole Oceanographic Institute, Teledyne Benthos, TriTech, LinkQuest, Desert Star Systems and EvoLogics. The communication uses either frequency shift keying with frequency hopping (FH-FSK), which is more reliable but provides lower data rates, or variable rate phase-coherent keying (PSK). Some models also include precise pulse-per-second clocks to allow synchronous ranging. The communication channel can be also shared using a TDMA scheme, where each node of the network is allotted a time slot within which to broadcast information. Nevertheless TDMA scheme has the main disadvantage that the total cycle time grows with the number of nodes. At present, the features of commercial modems in optimal conditions are shown in Table 2.2.
The use of acoustic modems provides two important advantages:

- The geo-reference of the beacons prior to starting the mission

Modem	Frequency [kHz]		Bit Rate [bps]	Range [km]
	Center	Bandwidth		
AQUAModem 1000	9.75	4.5	2000	5
MicronModem	22	4	40	0.5
MicroModem (FSK)	25	4	80	2
MicroModem (PSK)	25	5	5388	2
ATM9XX (PSK)	11.5/18.5/24.5	5	2400	6
ATM9XX (MFSK)	11.5/18.5/24.5	5	15360	6
ATM885	18.5	5	15360	0.7
S2CR 48/78	63	30	31200	1
S2CR 40/80	51	26	27700	1
S2CR 18/34	26	16	13900	3.5
S2CR 12/24	18.5	11	9200	6
S2CR 7/17	12	10	6900	8
UWM1000	35.695	17.85	17800	0.35
UWM2000	35.695	17.85	17800	1.5
UWM2000H	35.695	1 7.85	17800	1.5
UWM2200	71.4	35.7	35700	1
UWM3000	10	5	5000	3
UWM3000H	10	5	5000	3
UWM4000	17	8.5	8500	4
UWM10000	10	5	5000	10
SAM-1	37.5	9	154	1000

Table 2.2: *Commercial underwater acoustic modems. FSK, frequency-shift keying; PSK, phase-shift keying.*

is no longer necessary.

- The beacons can move during the missions.

Consequently, time and money are saved and the mission range can be extends as necessary without redeploying the sensor network. Many methods have been recently published that exploit these advantages [45], [46], [62], [65], [66].

2.5.2 Cooperative Localization

In the last ten years the research in AUV localization has been exploded shifting from old technologies, which require pre-deployed and localized assets, towards dynamic multi-agent system approaches that allow rapid deployment by minimal infrastructure. A particular instance is the localization system for a team of AUVs developed and experimented in Thesaurus Project [3], [4], [14]. Three acoustic modems, one USBL and a cooperative localization algorithm make the localization system. Where, the algorithm uses USBL measurements, range and navigation information to reduce errors on AUVs' position. Figure 2.16 depicts the scenario of the developed system. In this cooperative localization system one AUV stays on surface to get GPS and to geo-localize the other vehicles that remain underwater to continue the mission.

2.5.3 Inertial Systems

Two approaches are possible to resolve the navigation problem, one that uses and refers external aids, and second instead completely independents from them. This sub-section explains the first approach.

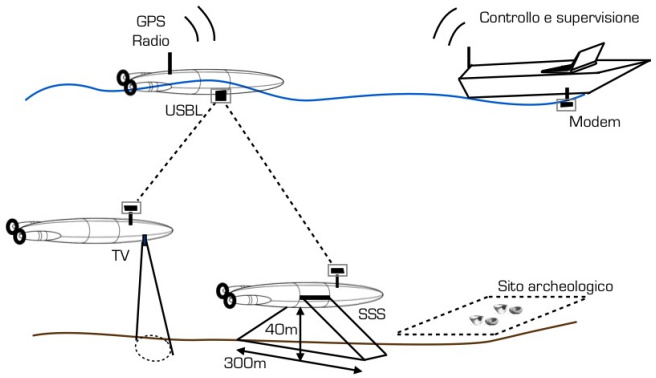


Figure 2.16: *Cooperative localization scenario. (Figure appears in [14])*

Dead-Reckoning is a navigation system not based on external aids. This system does not uses acoustic positioning support from a ship or acoustic transponders but it dead reckons to resolve the localization problem. Dead-Reckoning relies on the continuous updating of the position data derived from inputs of velocity components or speed and heading generated from a known start position. The main disadvantages are that errors are cumulative and the accuracy is largely influenced by the accuracy with which the initial position, velocity and heading are known. Consequently, the position error grows unbounded with distance traveled.

The inertial navigation system aims to improve upon the dead-reckoning pose estimation by integrating measurements from accelerometers and gyroscopes. These sensors can reduce the growth rate of pose estimation error, although it will still grow without bound. The inertial navigation is characterized by the navigation equations that are a set of nonlinear differential equations relating

vehicle's Attitude, Velocity and Position to known/measured inertial quantities. The inertial quantities are gathered by inertial sensors, accelerometers (f_{ib}^b) and gyroscopes (ω_{ib}^b), which represent the inputs of the navigation system. The inertial mechanization state variables can be defined as the angular parameterization Θ of the Direction Cosine Matrix $R_b^n = R_b^n(\Theta)$, which rotates from body (b) to navigation (n) frames, the velocity vector $V_n = [\quad V_n \quad V_e \quad V_d]^T$, expressed in navigation frame, and the position vector $r_e = [\quad \varphi \quad \lambda \quad h \quad]$, composed of the latitude, longitude and altitude of the navigation frame with respect to an Earth-fixed frame (e). Any navigation and Earth-fixed frames can be used. The NED and ECEF reference frames [37] are adopted and without loss of generality, the body frame is assumed coincident with the INU.

Following these assumptions, the continuous-time navigation equations resolved in the NED frame have the following form:

$$\begin{aligned}
\dot{\varphi} &= \frac{V_n}{R_m + h} \\
\dot{\lambda} &= \frac{V_e}{(R_n + h)\cos(\varphi)} \\
\dot{h} &= -V_d \\
\dot{V^n} &= R_b^n f_{ib}^b - (2\omega_{ie}^n + \omega_{en}^n) \wedge V^n + \gamma^n(\varphi) \\
\dot{R_b^n} &= R_b^n(\omega_{ib}^b - R_n^b \omega_{in}^n)\wedge
\end{aligned} \tag{2.37}$$

where ω_{in}^n denotes the *transport rate*, that can be computed as

$$\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n$$

The *transport rate* is as the summation between the NED frame angular velocity ω_{en}^n and the Earth rotation rate ω_{ie}^n, projected onto

the axes of the navigation frame. The angular velocity ω_{en}^n is defined as such velocity needed to make the navigation frame constantly aligned with the Geodetic North-East-Down configuration, while the body travels on the Earth surface. The two velocities of the *transport rate* are included into the navigation Equations 2.37 also, to account for the Coriolis and centripetal acceleration effects. The term $\gamma^n(\varphi)$ denotes the local gravity acceleration, aligned with the vertical axis of the navigation frame, $\gamma^n(\varphi) = [\quad 0 \quad 0 \quad \gamma_{local}^n(\varphi)]^T$. Note that the navigation equations depend on some local constants which are the Earth *WGS84* Datum constants, such as the local $Normal \quad (R_n)$ and $Meridian \quad (R_m)$ Earth radii of curvature, together with $\|\omega_{ie}^n\|$ and the local value of the gravitational acceleration, $\gamma_{local}(\varphi)$. Full derivation of the above equations and the detailed descriptions of the model local constants can be found in several textbooks (see, for instance,[57]).

2.6 Underwater Communication

Localization of underwater vehicle employs the communication if it is worked out by techniques based on either acoustic positioning systems or cooperative systems. Even if the underwater communication and localization are actually two distinct underwater problems, they become linked. For instance, the solution in [14] resolves the localization problem by a communication network. Consequently all the underwater localization systems based on acoustic measurements are affected by the underwater communication's failing.

Underwater communication is still a challenging problem due to

lack of available bandwidth, slow throughput rate, and unreliability of the acoustic channel [2], [16]. It endures many shortcomings [15] such as:

- Small bandwidth, which means that multiple pairs of communicating nodes have to use time division multiple access (TDMA) techniques to share the communication channel.
- Low data rate, which constrains the volume of data that can be transmitted.
- High latency since the speed of sound in water is only 1500m/s (five orders of magnitude slower than electromagnetic signals).
- Variable sound speed due to variable water temperature, density, and salinity.
- Multi-path transmissions due to the presence of an upper (free surface) and lower (sea bottom) boundary coupled with variable sound speed.
- Unreliability, resulting in the need for a communications system designed to handle frequent data loss in transmissions.

The design of distributed cooperative algorithms able to face these failures on communications with a team of AUVs is currently an active new area of research. The main purpose is ridden out the communication's shortcoming making a team of AUVs able to adapt their behaviours in accordance with the variation of acoustic channel [5], [15], [40], [47].

Chapter 3

Priori Density Map

This section defines a mathematical method that provides a map of a priori percentage density of expected findings in accordance with available historical archival data-base, either previous findings or new discoveries, over the survey area. The map is developed by applying Parzen theory with Gaussian kernels to a set of data which are built from the archival data-base keeping into account both the uncertainty and reliability of each datum. This a priori density map drives the choices of vehicles' waypoints in the systematic search strategy that it will be describe in the next chapter.

3.1 Problem Formulation

In the systematic search problem, an object rested on the seabed has its location given by a probability density and a fixed amount of a resource, in terms of energy, time, or money, is allocated to try to find this object. The classical theory centers on the determination

of the search location together with the distribution of search effort within it, to maximize the expected percentage of detections, without spending more than the allocated amount. Therefore, the union of the expected density of detection associate to each object draws the a priori density map over the survey area.

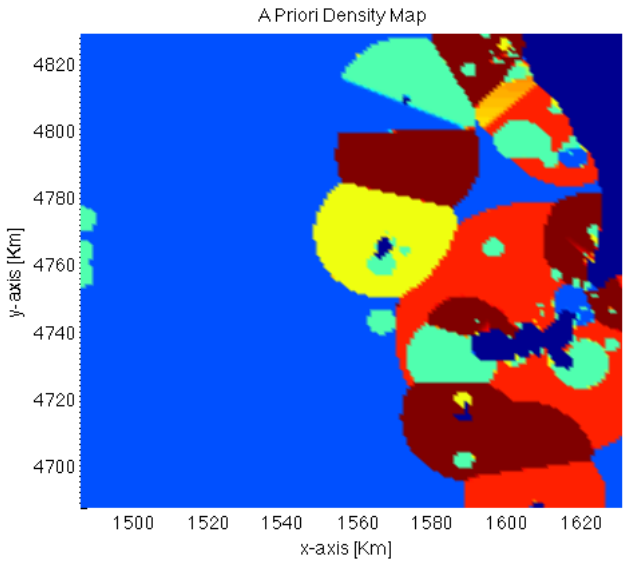


Figure 3.1: *The a priori density map built by the Bayesian approach. Hot colours indicate points with high expected density of objects on the seabed, while cold colours indicate low expected density.*

Historically, the a priori density map was built through the classical Bayesian inference, which allows the organization of available data with associated uncertainties and computation of the Probability Distribution Function (PDF) for target location given these data. Bayesian approach organizes the available priori information

into consistent scenarios, quantifying the uncertainties with probability distributions, weighting the relative likelihood of each scenario, and performing a simulation to produce a prior PDF for the location of the object rested on the seabed. The Bayesian approach doesn't compute a smooth map but a too crude approximation of the a priori percentage density of expected findings, as Figure 3.1) shows. Hence, a refined mathematical method is provided adapting the theory of Parzen windows [50] to account for the qualitative information available from archaeological data-bases. Parzen window theory naturally adapts to, or at the very least is able to cope with, data-base where each entry has a qualitative judgment on the reliability of the indication itself and on the accuracy of the referred position.

3.2 Mathematical Method

In the early '60s it has been proposed by Parzen [50] a mathematical method to estimate the Probability Density Function (PDF) through Gaussian kernel function from a data set given. In this section the Parzen method is refined to draw an a priori density map.

Available a priori information can be formalized by considering a set of data $\mathbb{X} = \{X_i\}$ as points X_i, $i = 1, \ldots, N_{data}$ with some attributes. For modelling purposes, each element in the data set has three attributes to encode the spatial and qualitative information. These attibutes are: the geographical position x_i of the datum X_i, which represents the specific position on the map where that particular finding was discovered/reported, the reliability P_i of the

indication and the *uncertainty* $\sigma_i \propto \frac{1}{\lambda_i}$ of the referred position, being λ_i the accuracy. Each point is thus a triple of attributes

$$X_i(x_i,\ P_i,\ \sigma_i)\ ,\ i = 1,\ \ldots,\ N_{data} \tag{3.1}$$

Note that here the concept of *reliability* is recast into that of *probability*, for which the symbol P is used. This is made in the sense that the more reliable a particular datum is, the more probable it is finding objects of archaeological interest in that specific position. For this reason, the value of this latter attribute is rescaled in the range [0; 1] to formally represent a probability.

The estimation of a priori density map is interested in representing the knowledge encoded in the set of data $\mathbb{X}$, given the single observations X_i, $i = 1$, ..., N_{data}, to finally obtain an a priori description of the environment in terms of percentage density of detections per area. Actually, it is interested in computationally simple estimators of this density, that have the further property to be continuous and differentiable, given that the aim is to employ such distribution in an optimization framework. Non-parametric sample-based estimators, usually adopted in assessing the *probability* density function of a set of given data, can be considered good candidates to be adopted in this particular case also, being able to naturally cope with the spatial and qualitative information embedded in the data X_i. For this aim, a huge literature has been produced; one can refer to the review in [29], for example. What renders our problem particular, is the further description of the entries in the historical database in terms of the reliability of the referred position, which represents a very useful and discriminating characteristic. *De facto*, the method of estimation would like to give much more importance to that indi-

cations which reliability is very high, with respect to the ones that are lesser and lesser reliable, such that the resulting spatial distribution is *less affected* by them. This latter characteristic, validates to employ a non-parametric estimator based on Parzen window theory [50], which seems to naturally adapt to the entire set of information encoded in the data X_i. The remaining part of the section illustrates the process of estimating the whole *prior* spatial distribution of the given set of observations X_i.

Given the intrinsic probabilistic nature of the data $X_i, ; \forall i$, it can be reasonable to associate a random variable $\chi_i \sim p_{\chi_i}$ to each point, which is able to represent the confidence in the knowledge of the *average* position of a specific element and how such position is uncertain, i.e. how it is expected to be *spread* over space. Since no other information about the actual probability distribution p_{χ_i} of such data is generally available, we chose to associate a Gaussian random variable to each point, having the geographical position (north-east) as the mean value and the uncertaintly σ_i as the standard deviation. Given that the accuracy in the referred position (thus the uncertainty σ_i) is defined as a scalar value, the same value of the variance along the two directions is assigned, that is

$$p_{\chi_i} = N(x_i,\ \sigma_i^2 I) \quad i = 1,\ ...,\ N_{data} \tag{3.2}$$

At this stage, the problem is partially modeled, since the variables χ_i fully describe the spatial uncertainty in the corresponding archival datum, however they do not encode the reliability information. This work proposes to cope with such futher attribute in a Monte Carlo fashion. A realization of each Gaussian prior p_{χ_i}, by sampling a certain number of particles $\pi_{ij},\ j = 1,\ ...,\ m_i$ from p_{χ_i}, is obtained.

The number m_i of particles extracted from each prior is proportional to the reliability P_i. More specifically: a total of K_{max} particles are extracted per each random variable when $P_i = 1$, thus:

$$m_i = K_{max} \cdot P_i, \quad i = 1, \ ..., \ N_{data} \tag{3.3}$$

Clearly, there is no a unique criterion for the determination of the K_{max}. Although arguably other choices can be made, this work proposes to select such value based on the desired resolution of the density map. In the next section, some comparisons in the obtained map, with different values of the gain K_{max}, are proposed.

Given the ensemble of particles $\Upsilon_\pi = \{\pi_{ij}\}, \ \forall i, \ j$, the prior density map of findings can be estimated by using the Parzen method [50], with Gaussian kernels

$$\begin{aligned} \hat{p}(\pi) &\doteq \frac{1}{N_\pi} \sum_{\pi_{ij} \in \Upsilon_\pi} \phi(\pi - \pi_{ij}, \Sigma_\pi) \\ &= \frac{1}{N_\pi} \sum_{i=1}^{N_{data}} \sum_{j=1}^{m_i} \phi(\pi - \pi_{ij}, \Sigma_\pi) \end{aligned} \tag{3.4}$$

where $\phi(\cdot)$ is the bivariate Gaussian kernel with mean π_{ij} and covariance matrix Σ_π and $N_\pi = \sum_{i=1}^{N_{data}} m_i$. By an intuitive point of view, Equation 3.4 gives a closed form expression, as a function of the geographical position π, of the (normalized) percentage density of findings per area. For instance, when integrated over a certain area $\mathcal{A}$, the distribution gives the expected *relative percentage* of findings over the area $\mathcal{A}$, that is:

$$\frac{N_\mathcal{A}}{N_{data}} \approx \int_\mathcal{A} \widehat{p}\,(\pi)\, d\pi \tag{3.5}$$

being $N_{\mathcal{A}}$ the expected number of findings on that area and N_{data} the a priori total number of (expected) objects in the environment, Equation 3.1.

For implementation purposes, note that the covariance matrix of the Gaussian kernels Σ_π (often called *bandwidth* matrix), which is the same among the particles, is a free parameter and plays an important role in the estimation task. It exhibits a strong influence on the resulting density estimate and a number of rules do exist in the literature regarding its choice. In particular Wand and Jones [64] identified various alternatives of specifying the bandwidth matrix. However, they showed that using two independent bandwidths, one per each coordinate, is often adequate. This was the choice adopted in this work as well. For the purposes of the problem, we obviously expect to cope with a multimodal density underlying the data set $\mathbb{X}$, given a putative realization of the variables χ_i. For this reason, the optimal selection of the bandwidth for multimodal densities [28], one per each coordinate (north, east) of the particles, is used

$$\Sigma_\pi = \begin{bmatrix} \sigma^2_{\pi,n} & 0 \\ 0 & \sigma^2_{\pi,e} \end{bmatrix} \tag{3.6}$$

being

$$\sigma_{\pi,i} = 0.9 \min\left(\mathrm{STD}_{\pi,i}, \frac{\mathrm{IQR}_{\pi,i}}{1.34}\right) N_\pi^{-1/5},\ i = n, e \tag{3.7}$$

In the previous equation, $\mathrm{STD}_{\pi,i}$ and $\mathrm{IQR}_{\pi,i}$ are respectively the sample-based standard deviation and the interquartile range of the particles set, computed over the i-th coordinate.

Table 3.1 summarizes the algorithm used to generate the prior spatial density map, given the set of points $X_i(x_i,\ P_i,\ \sigma_i)$.

Prior Density Map Generation
Input: historical data set $\mathbb{X} = \{X_i(x_i,\ P_i,\ \sigma_i)\},\ i = 1,\ ...,\ N_{data}$; gain value K_{max}. **Algorithm:** 1) For each datum X_i: a) Define the Normal variable χ_i, by using Eq. 3.2; b) Compute $m_i = K_{max}P_i$, Eq. 3.3; c) Sample m_i particles from Gaussians χ_i; 2) Given the particle set $\{\pi_{ij}\}$, compute the bandwidth matrix, Eq. 3.6; 3) Estimate the density function around the particles set, using the Parzen window method, Eq. 3.4;

Table 3.1: *Summary of the Prior Density Map Generation algorithm*

3.3 Tuscan Archipelago: Prior Density Map

This section applies the Parzen method with Gaussian kernels to build the a priori density map of findings on the Tuscan Archipelago, by a data set originate from the reserved data-base of the Tuscan Superintendence on Cultural Heritage.

3.3 Tuscan Archipelago: Prior Density Map

Archival historical data set is given from the reserved data-base of the Tuscan Superintendence on Cultural Heritage, regional Authority responsible of cultural heritage preservation. This data-base includes both actual findings and the indications as gathered from oral sources (divers, fishermen, etc.); each indication comprises a qualitative judgment on the reliability of the indication itself and on the accuracy of the referred position. For instance, an amphora caught in a fishing net represents a very reliable information (it is actually a *finding*), however the accuracy of the position could be very low; moreover, the unconfirmed indication of an amateur diver may be associated to both low reliability of information itself and low accuracy in the referred position. Figure 3.2 shows the plot of positions of such points localized in the northern part of the Tuscan Archipelago. Note that the red dots represent the position of historical harbors in the area.

Once the data set is given, each element has associated three attributes and the gain value K_{max}. Thus the refined Parzen method, summarized in Table 3.1, generates the a priori density map.
As mentioned in previous section, there is not a unique criterion to choose the gain value K_{max}. Figure 3.3 show some comparisons in the obtained density map with different values of the gain K_{max}. Note that the big gain values generate sharp maps with a lot of small areas with high expected density of objects on the seabed, i.e. where the number of findings is great, and the remaining areas where the number of findings is low have an zero expected density of objects. Instead, small gain values generate smooth maps without isolated areas with high expected density. All the obtained density maps are reasonable and the choice of the gain value K_{max} depends on

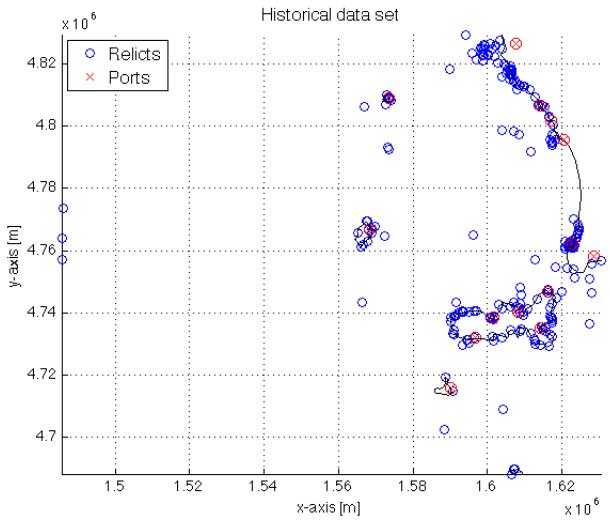


Figure 3.2: *Plot of the historical archival data position in the Tuscan Archipelago.*

the desired resolution of the map. For instance, in the next chapter the map is going to drive the choice of the vehicles' waypoints in a systematic search strategy. Therefore, by an intuitive viewpoint, the intent behind the chosen value is to deal with a smoother map than with a sharper one (see Figure 3.3), in order to force a wider coverage of the exploration area. In effect, one region with low expected percentage of detection could still have some object resting on the seabed. Thus all the areas of the map will have to be explored with different priority. For the systematic search strategy, defined in detail in the next chapter, the empirically optimal gain value is 5, as Figure 3.5 shows. Hot colours correspond to regions with an expected high density of objects of archaeological interest on the

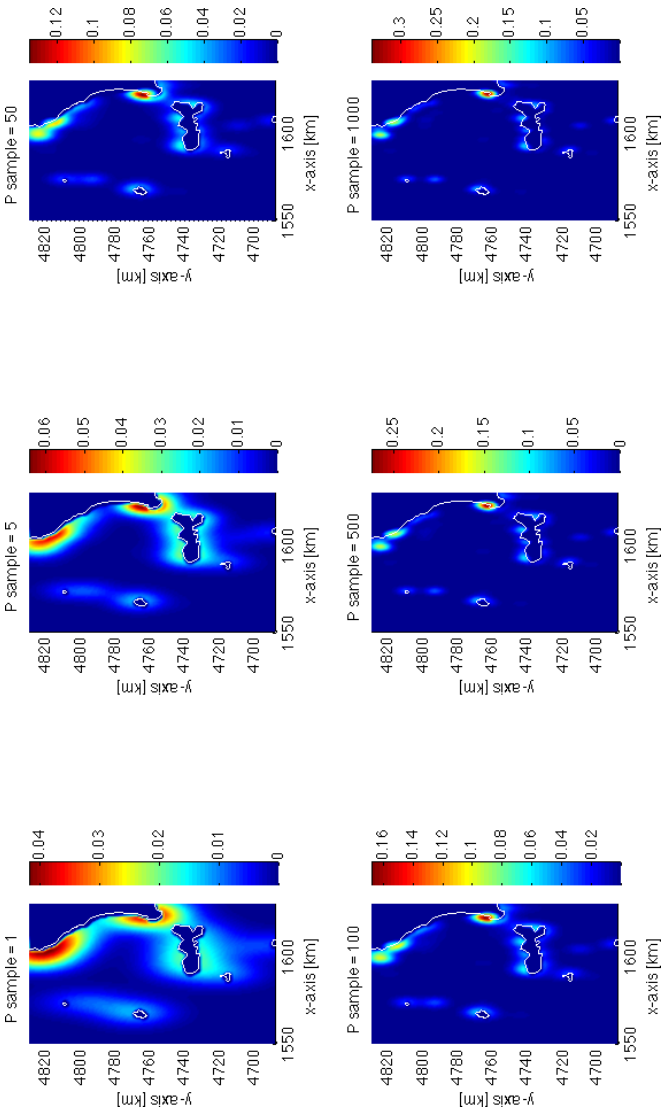
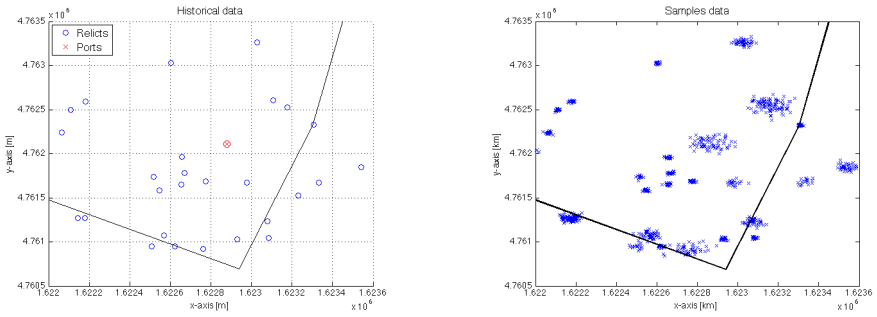


Figure 3.3: *Prior density maps obtained with different choices of the max number K_{max} of samples per Gaussian variables.*

seabed. Clearly, hot colours are localized in the regions where the number of points is greater and they are very close each other. It is worth to note, however, that the value of the resulting map is only in part determined by the spatial density of the points X_i in a certain region – compare Figures 3.2 and 3.5.



(a) (b)

Figure 3.4: *(a) Zoom of the historical data plot. (b) Zoom of the same area showed in (a) with sampled data. Closely clustered data indicate that the point that originated them is highly reliable, hence the position has small variance. The number of generated samples from any individual point in (a) is proportional to the probability of the original point.*

The distinctive characteristic is actually the value of the reliability and accuracy attributes, which are responsible of the number and concentration of the particles in a given region, as Figure 3.4 suggests. Figure 3.4(b) shows the particles sampled from the random variables χ_i, in the same area of Figure 3.4(a), but noticing the difference in the dispersion and cardinality of the clouds of particles, depending on the corresponding qualitative indicators. In this

sense, in Figure 3.5 the area (for example) around the coordinates (1620; 4760) presents an higher value in the density map than the area around the coordinates (1590; 4740), even if the two regions have comparable spatial densities of points X_i, as Figure 3.4 shows. This is due to the fact that in the first region, the corresponding findings were marked with higher values of reliability and accuracy, with respect to the second one, thus resulting in a expected higher density of findings.

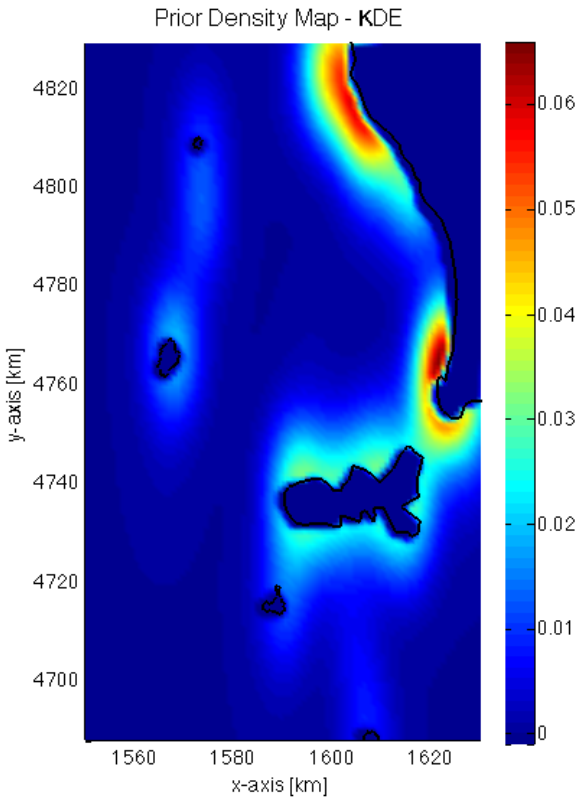


Figure 3.5: *The prior density map built by the kernel density estimator over the Tuscan Archipelago. Hot colors indicate points with high expected density of objects on the seabed, while cold colors indicate low expected density.*

Chapter 4

Cooperative Distributed Algorithm

In the Thesaurus project [3] a team of three Autonomous Underwater Vehicles (AUVs) is considered to explore the Tuscan Archipelago, historical marine area, by a cooperative distributed algorithm. This algorithm implements a systematic search strategy, being cooperative, online, and adaptive. It uses an a priori density maps to compute the AUVs' waypoints. The proposed algorithm is going to overcome the issues of the standard planning methods (zig-zag or lawn-mower paths) exploring the survey marine area more efficiently. The efficiency is evaluated in terms of relicts found in a fixed time, time to complete the mission, number of relicts found and area explored for relicts found. Note that while the original project that inspired this thesis, as well as our simulations and test cases, consider three AUVs, the proposed algorithm is scalable in terms of motion planning. However, the algorithm is subject to increasing

communication delays as the number of AUV increases, since the communication backbone is based on a TDMA MAC (see Section 2.5.1).

4.1 Problem Formulation

Motion coordination for a multi-agents system is intended as the interaction among the vehicles throughout an explorative mission. A cooperative distributed algorithm is proposed resolving the problem of motion coordination to execute the explorative mission, with a team of AUVs, surveying a marine area. In particular, the algorithm has the following properties:

- cooperation: each vehicle explores a sub-region of the total search area.
- adaptation: the plans is updated basing on the sensor data gathered on the mission in progress.
- online computation: the vehicles' waypoints are computed based on the vehicles' locations and the a priori percentage density of expected findings.

Historically, the motion coordination problem for a multi agents system was resolved using systematic methods that involved preplanning paths, either zig-zag or regular lawn-mower transects, pattern. This method has some shortcomings:

- The available vehicles not are used in the best mode to cover the space to the degree required. Some subspaces may need to searched more than once and thus require denser coverage.

- The vehicles are not assigned to sub-regions of the survey area and so they do not take advantage of the workload sharing method.
- The mission employs a lot time to explore all the survey area.
- Although during the mission some objects are discovered, the pre-specified paths not change. Instead, these objects may have influence on a priori information used at the beginning of the mission and therefore a new planning path may be request.
- The map of a priori percentage density of expected findings is not update dynamically with the exploration is in progress.
- The AUVs are not able to allow a cooperative communication and localization. Once the vehicle submerges, its location estimate will drift, resulting different from the pre-specified path.

Therefore the novel cooperative distributed algorithm is proposed to overcome these issues.

The cooperative distributed algorithm take the a priori density map over the survey area as input and search along the routes that maximize the expected percentage of detections. The map of input is calculated by the mathematical method explained in chapter 3. Notice the search is *cooperative*, thus each vehicle needs to negotiate with the others which portion of the neighboring area to search. Once the vehicles have agreed on area subdivision, each one independently selects its waypoints within the area assigned to it. The policy ensures the workload sharing among the vehicles, conditioned over the density map. Each vehicle is thus assigned to a portion of

the area with the same expected density of detections of the other ones in the team. At the same time, it is desired to fulfill given range constraints, in order to keep each agent within communication range and to avoid collisions. With the above assumptions, the proposed cooperative distributed algorithm can be summarized in the following main steps:

- Determination of the vehicles sub-area via Equitable Power Diagrams.
- Definition of the vehicles waypoints via maximization of the information gain, within the assigned area.
- Motion and sampling of the assigned area.
- Update of the density map, in accordance to the actual result of the survey.

These steps represent the algorithm's topics that will be explained in detail in the next sections.

4.2 Equitable Power Diagram: area partition among the vehicles

The Equitable Power Diagrams [53] are particular extension of the Voronoi Diagrams [31], [48], recently proposed as partitioning policy in multi-agents systems. The diagrams allow the regulation of the applied resource allocation among *customers*, to balance the assigned workload to each *resource*. In multi-agent system applications the resources are the available agents and the customers are specific points

of interest over a given *workspace* Q. In the proposed algortithm, the workspace is defined as the finite set of 2-D spatial coordinates – say North and East – of the area to be surveyed. Within the framework of the Equitable Power Diagram, the workspace is partitioned into N_v sub-regions $Q_i \subset Q,\ i = 1,\ \ldots,\ N_v$, one per each agent, parameterized via the distinct position $g_i \in Q_i$ of the agents and the corresponding weights w_i, which encode a certain measure of the workload over that specific region. Such weights de facto define the sub-region's bounds.

For the sake of clarity, here is followed the same notation used in [54], in order to allow the interested reader to easily refer to the cited work for the details of the employed theory, that are omitted for brevity. Therefore, W indicates the set of weights and $G_W(t)$ the set of *power points*, that is

$$G_W(t) = \{(g_1(t)\,,\, w_1(t))\,,\, ...,\, (g_{N_v}(t)\,,\, w_{N_v}(t))\}\,,\, t \geq 0 \qquad (4.1)$$

Moreover, $E(W)$ indicates the local energy function, which maps the set of weights into a positive real number. Such function encodes a measure built over the prior map, defined over the marine area of interest. With this assumption, that is

$$E(W) \doteq \sum_{i=1}^{N_V} \left(\int_{V_i(G_W(t))} \hat{p}(\pi)\ d\pi \right)^{-1} \qquad (4.2)$$

where, $V_i(G_W(t))$ is the *power cell* [54] of the power diagram $\mathcal{V}(G_W(t))$ assigned to $i-$th vehicle at time t and $\hat{p}(\pi)$ is the (whole) prior density function defined over the workspace Q. With the above definitions, the Equitable Power Diagram can thus be computed by using the algorithm defined in [54]. In particular, the partitioning of the

area among the vehicles is made by choosing the set of weights which minimizes the local energy function 4.2. The iterative adaptation law of the $i-$th weight is

$$\dot{w}_i(t) = -\frac{\partial E(W)}{\partial w_i}, \quad t \geq 0 \tag{4.3}$$

being

$$\frac{\partial E(W)}{\partial w_i} = \sum_{j \in N_i} \frac{1}{2(\parallel g_j - g_i \parallel)} \cdot \left(\frac{1}{\hat{p}(\pi)^2_{V_j(G_W(t))}} - \frac{1}{\hat{p}(\pi)^2_{V_i(G_W(t))}} \right) \cdot \int_{\triangle_{ij}} \hat{p}(\pi)\, d\pi \tag{4.4}$$

where, N_i denotes the set of indices of the power neighbors of i-th vehicle and $\triangle_{ij}$ is the bound between i-th vehicle and j-th vehicle. Details of the above development can be found in [54].
The optimization is made in a discrete-time fashion, starting from value zero for $w_i(0), i = 1, ..., N_v$. It is important that all the initial weights have the same value. The optimal solution for the set of weights exists, since the energy function 4.2 has two main proprieties, as defined in [54]:

- it is based on the weights of the vehicles;
- all its critical points correspond to vectors of weights yielding an equitable power diagram.

This is the case of the study in this work also.

Figure 4.2 shows an example of the Equitable Power diagram and the Voronoi diagram calculated with the same team of vehicles. Figure 4.1 depicts the function of measure, $\hat{p}(\pi)$, defined over

the workspace Q, used to compute the Equitable Power diagram in Figure 4.2.

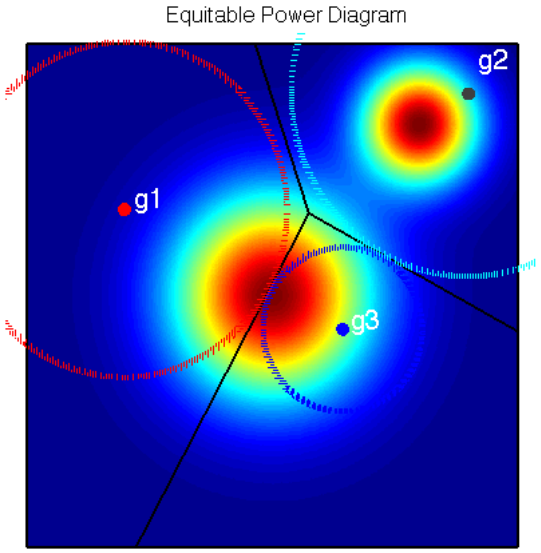


Figure 4.1: *Measure function is defined on the background and the corresponding Equitable Power Diagram by black line is drawn over.*

4.3 Information-driven Waypoints Selection

Given the prior map derived from the historical information, the purpose of the search strategy is to direct the search along the routes that maximize the expected percentage of detected objects, or, in an equivalent fashion, maximize a certain *information gain*. The

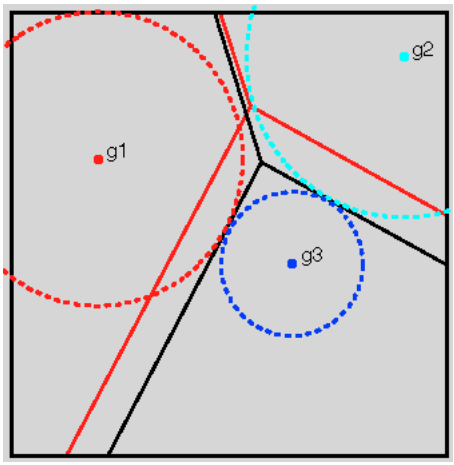


Figure 4.2: *The Voronoi diagram and the Equitable Power diagram are in comparison. Red lines representative the Voronoi diagram, blue lines the Equitable Power diagram, dotted circle the agents' weights and dots the positions of the vehicles over the area.*

idea is to choose the local routes (within the assigned sub-region) along which the percentage of detections is higher. In practice the path that maximizes the expected density of objects on the seabed among the possible straight-line paths of pre-assigned length (or time duration) departing from the vehicle current position is found. Furthermore, when the vehicles reach the selected waypoints, a new set of locations is generated, by using the portion of the map which has not been already visited.

Among the available techniques in the literature, see the review in [29] as an example, the systematic search strategy is interested in computationally simple estimators of the Information measures, that have the property to be continuous and differentiable. Again,

a non-parametric estimator is chosen, which can take advantage of the non-parametric characteristic of the prior density map, based on the Parzen window technique. It will be shown that the optimized measure of information used in this work can be recast as the minimization of a proper Renyi's cross-entropy between the possible routes and the map.

Let $S_Q \subset Q_i$ be a sector of the i-th sub-region, encoding a possible route. For example, this sector could be an area, a line or a geographic position. Recalling Equation 3.5, the percentage of detections η over the route S_Q can be evaluated by integrating the density map over S_Q, that is:

$$\eta\left(\pi \in S_Q\right) = \int_{S_Q} \widehat{p}\left(\pi\right) d\pi \tag{4.5}$$

Using the non-parametric estimation of the function $\widehat{p}\left(\pi\right)$ derived in Equation 3.4, yields to

$$\begin{aligned} \eta\left(\pi \in S_Q\right) &= \int_{S_Q} \widehat{p}\left(\pi\right) d\pi \\ &= \int_{S_Q} \frac{1}{N_\pi} \sum_{\pi_{ij} \in \Upsilon_\pi} \Phi\left(\pi - \pi_{ij}, \Sigma_\pi\right) d\pi \end{aligned} \tag{4.6}$$

The optimal search can thus be formulated as finding the route which maximizes $\eta\left(\pi \in S_Q\right)$ over the given sub-region, that is

$$\max_{S_Q} \int_{\{S_Q\}} \frac{1}{N_\pi} \sum_{\pi_{ij} \in \Upsilon_\pi} \Phi\left(\pi - \pi_{ij}, \Sigma_\pi\right) d\pi \tag{4.7}$$

where $\{S_Q\}$ denotes the set of all possible routes contained in the sub-region Q_i and S_Q is the one which maximizes the cost 4.7.

Clearly, the evaluation of the integral in Equation 4.7 does not generally admit a closed form, since the set $\{S_Q\}$ has an infinite cardinality, as well as every component $S_Q \in \{S_Q\}$. For this reason, a Monte Carlo sampling approximation is applied, by employing a finite discretization $\{S_Q\}_k$ of the foregoing set and by sampling a certain number of points within the selected route $S_Q \in \{S_Q\}_k$. Without loss of generality and in absence of prior assumptions for the distribution of the points in S_Q, a density function $U(S_Q)$ which depends on the nature of S_Q is sampled. For instance: if S_Q is a line, then $U(S_Q)$ is chosen uniform, while if it is an area (or a *point*), $U(S_Q)$ is chosen normal, with proper covariance matrix. Under the above assumptions, the integral can be evaluated numerically as

$$\eta(\pi \in S_Q) \approx \frac{1}{N_{S_Q}} \frac{1}{N_\pi} \sum_{\pi_n \in S_Q} \sum_{\pi_{ij} \in \Upsilon_\pi} \Phi(\pi_n - \pi_{ij}, \Sigma_\pi) \tag{4.8}$$

being $\pi_n \in S_Q,\ n = 1, ..., N_{S_Q}$ the finite discretization of the foregoing sector, sampled from the distribution $U(S_Q)$. Actually, Equation 4.8 can be already used for the optimization of the search path. However, for the purposes of the presented work, a slight modification of the above measure is proposed by embedding a further degree of freedom in the computation of the variance of the Gaussian kernel in Equation 4.8, that is

$$\eta(\pi \in S_Q) \approx I(S_Q, Q) \tag{4.9}$$

where $I(S_Q, Q)$ can be interpreted as a suitable *information potential* [28], defined as

$$I(S_Q, Q) \doteq \frac{1}{N_{S_Q}} \frac{1}{N_\pi} \sum_{\pi_n \in S_Q} \sum_{\pi_{ij} \in \Upsilon_\pi} \Phi\left(\pi_n - \pi_{ij}, \Sigma_\pi + \Sigma_{S_Q}\right) \tag{4.10}$$

where, the covariance is augmented with the term Σ_{S_Q}. The reason behind such choice is twofold. First, that is convenient to include an uncertainty (encoded by Σ_{S_Q}) in the search of the route which maximizes the cost function, over the assigned sub-region Q_i. This allows to keep the sampling of the possible routes made by the search algorithm (i.e. the selection of a finite number of routes among the infinite set of possible ones) less dense, if an exhaustive optimization is performed. On the other hand, following [28], Equation 4.10 has an interesting theoretical interpretation. In fact, it is straightforward to show that Equation 4.10 is the argument of the natural logarithm of the non-parametric estimation of the Renyi cross-entropy between $\widehat{p}(\pi)$ and $U(S_Q)$, the latter being estimated employing a Gaussian kernel built from π_n, with covariance Σ_{S_Q}. That is

$$I(S_Q, Q) \doteq \int \hat{U}(S_Q)\, \hat{p}(\pi)\, d\pi \tag{4.11}$$

with $\hat{p}(\pi)$ given by Equation 3.4 and:

$$\hat{U}(S_Q) = \frac{1}{N_{S_Q}} \sum_{\pi_n \in S_Q} \Phi(\pi - \pi_n, \Sigma_{S_Q}) \tag{4.12}$$

Equation 4.7 together with 4.9 and 4.11 aim at showing that solving the maximization problem, Equation 4.7, can be considered numerically equivalent to minimizing the following Renyi cross-entropy, with respect to the possible $S_Q \in \{S_Q\}_k$.

$$H_R(S_Q, Q) = -\ln I(S_Q, Q) \tag{4.13}$$

In the following subsections two possible methods are present for the optimization of the information measure 4.13. The first one is

actually a batch search over a finite set of possible routes, made via gridding, radiating, etc. each sub-region and by selecting the ones that minimize the information measure. The other one is a gradient descent iterative search scheme, which minimizes the given information measure over a certain parametrization of the sub-regions. In this thesis the Gradient based optimization will be employed in the cooperative algorithm, while the Exhaustive search is reported here only for completeness.

4.3.1 Exaustive optimization

In the batch search variant of the algorithm, the optimization problem can be formalized, per each vehicle, as

$$\begin{cases} \hat{S}_Q = \min_{S_Q} \ H_R\left(S_Q, Q\right) \\ \forall S_Q \in \{S_Q\}_i \subset Q_i \end{cases} \tag{4.14}$$

Every possible route S_Q inside the finite set $\{S_Q\}_i$ is checked with respect to the measure 4.13. For example, it is possible to span in a finite set of directions the region assigned to the i-th vehicle via straight lines of fixed lengths. Thus, a possible route S_Q is a straight line along one of these directions. At each evaluation, N_{S_Q} particles are sampled from the density $U\left(S_Q\right)$ (which is chosen uniform in the example case of straight lines) and the cost function in Equation 4.13 is evaluated. The optimal waypoint is the end-point of the route encoded by $\widehat{S}_Q$, for which the value of the information measure is minimum.

The optimal selection of a new waypoint is applied every time the *current* waypoint is reached, that is, when the relative difference

between the current position of the vehicle and the waypoint is below a certain given threshold. When the result of the optimization is a new position which has never been visited before, the optimization step is considered valid and the vehicle is driven toward the selected waypoint, by using the policies discussed in the next section. However, the optimal search of a new waypoint for one vehicle can fail in two cases:

- no points exist, inside the region assigned to the vehicle, with lesser entropy than the entropy value of the current vehicle position;
- the new waypoint is chosen among the portions of the map already visited.

Usually, the first condition happens in those regions where the density map is constant, that is sufficiently far from those regions with high expected density of objects. On the other hand, the second condition usually happens when the vehicle reaches a portion of the map where the expected density of objects reaches a local maximum. In this case, the optimal criterion would force the vehicle to remain inside this area, if no specific countermeasures are taken. In both these cases we use a heuristic method to push the vehicle out of this *local optimum* by fostering the portions of the map that are still unexplored. Thus an occupation map of the environment, composed by regular cells controlled via a binary variable, is employed (see Figure 4.3). Each cell assumes the value one if it is coincident with a position with an obstacle or which was previously explored, otherwise it assumes the value zero. The heuristic method simply selects a new

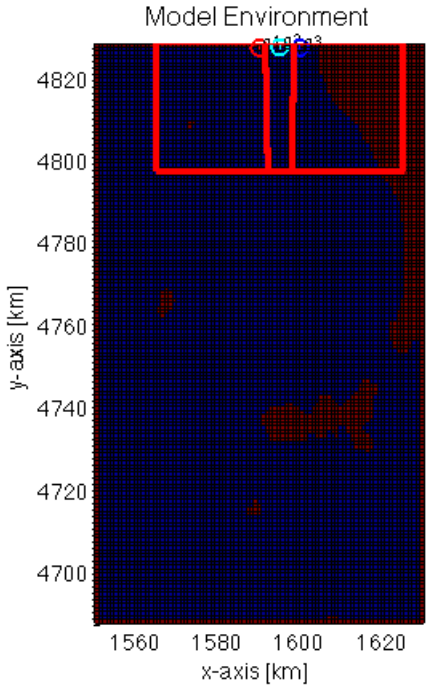


Figure 4.3: *An example of the initialization procedure: the map is divided into square cells. The red colour corresponds to value one, while the blue colour to value zero.*

waypoint coincident with one of those locations that are marked as *unexplored* and are *sufficiently far* from the current vehicle position. In this way, the algorithm marks the already explored areas as *not interesting* for the future selections of the waypoint.

4.3.2 Gradient-based optimization

An alternative method to the batch search is obtained via parametrization of the possible exploration route, starting from the vehicle current position. In this case the problem can be simplified by directly representing with S_Q the waypoint, i.e. a pure position inside the sub-region Q_i. Such position is thus modeled as a random variable $\xi(\theta)$, parametrized via a vector θ. Without loss of generality the random variable is assumed normally distributed around the given position $\overline{\xi}(\theta)$, with covariance Σ_ξ, that is $\xi(\theta) \sim N(\overline{\xi}(\theta), \Sigma_\xi)$. The parameter vector θ may encode either a polar couple, north/east position. Thus, the random variable can be written as

$$\begin{aligned} \theta &= \begin{bmatrix} \alpha \\ R \end{bmatrix} \rightarrow \xi(\theta) \sim N\left(\begin{bmatrix} R\cos\alpha \\ R\sin\alpha \end{bmatrix}, \Sigma_\xi \right) \\ \theta &= \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \xi(\theta) \sim N(\theta, \Sigma_\xi) \end{aligned} \tag{4.15}$$

In this case, the goal is to minimize the resulting information measure with respect to the selected parametrization of the position. It is worth to notice that now the discretization over the possible route (i.e. the particles π_n) can be represented by a single sample, that is the mean of the random variable $\xi(\theta)$ evaluated at a specific value of the parameter. Recalling Equation 4.11, this means that the cross-information is now evaluated between the prior map and the normal variable $\xi(\theta)$, that is

$$I(S_Q(\theta), Q) \doteq \int \Phi\left(\pi - \bar{\xi}(\theta), \Sigma_\xi\right) \widehat{p}(\pi)\, d\pi \tag{4.16}$$

Thus the Information potential, Equation 4.10, and the corresponding Renyi cross-entropy are reformulated accordingly

$$\begin{aligned} I\left(S_Q\left(\theta\right), Q\right) &\doteq \frac{1}{N_\pi} \sum_{\pi_{ij} \in \Upsilon_\pi} \Phi\left(\bar{\xi}\left(\theta\right) - \pi_{ij}, \Sigma_\pi + \Sigma_\xi\right) \\ H_R\left(S_Q\left(\theta\right), Q\right) &= -\ln I\left(S_Q\left(\theta\right), Q\right) \end{aligned} \tag{4.17}$$

The optimization problem can be thus formalized, per each vehicle, as

$$\begin{cases} \widehat{\theta} = \min\limits_{\theta} \; H_R\left(S_Q\left(\theta\right), Q\right) \\ S_Q\left(\theta\right) \subset Q_i \end{cases} \tag{4.18}$$

or equivalently

$$\begin{cases} \widehat{\theta} = \max\limits_{\theta} \; I\left(S_Q\left(\theta\right), Q\right) \\ S_Q\left(\theta\right) \subset Q_i \end{cases} \tag{4.19}$$

According to the Gradient descent approach, the parameter θ can be thus updated as

$$\theta_{k+1} = \theta_k - \gamma \nabla_{\theta_k} H_R\left(S_Q\left(\theta\right), Q\right) \tag{4.20}$$

or equivalently (gradient ascent over the information potential)

$$\theta_{k+1} = \theta_k + \eta \nabla_{\theta_k} I\left(S_Q\left(\theta\right), Q\right) \tag{4.21}$$

The optimization algorithm is applied until the estimation vector θ_k reaches a stationary point, within a tolerance ϵ, such that $\|\theta_{k+1} - \theta_k\| < \epsilon$.

4.4 Rules for movement of the vehicles: potential function

Motion planning is implemented in different approaches: heuristic, potential functions, roadmaps and cell decomposition. In this work, the second approach based on potential function is going to be used.

Motion planning based on potential functions theory defines the vehicles' paths following negative gradient of a particular function. This function – say potential function - is computed over the vehicles' workspace in based on the available knowledge of goals and obstacles' locations. In particular each goal and each obstacle represent an attractive and repulsive force, respectively. The sum of positive and negative forces, defined per each workspace's point, is the potential function. Consequently, this motion planning method is computationally simple because after the definition of the potential function the gradient is automatically computed.

In this thesis the workspace Q is a marine area to be surveyed, which is known and simplified in a set of 2-D spatial coordinates – say North and East –, since the vehicles move at constant depth on the seabed gathering data by looking-sensor (either side-scan-sonar SSS or multi-beam echo sounder MBES). For modelling purpose, each vehicle motion is described as a pure kinematic point by the following first-order differential equation:

$$\dot{x}(t) = u(t) + v(t) \tag{4.22}$$

where, $x \in \Re^2$ is the vehicle position, $\dot{x} \in \Re^2$ is the vehicle speed, $u \in \Re^2$ is the control input to be defined, such that, at each time instant, the vehicles avoid collisions and move towards the waypoints.

Finally, the term $v \in \Re^2$ allows for the inclusion of external motion disturbances acting on each vehicle as, for instance, marine currents. Although the proposed kinematic model is rather elementary, it is sufficient to define the motion planning in this work. This kinematic model decouple the performance of the search algorithm from the dynamical characteristics of the vehicles and thus it will allow to analyse the proposed cooperative distributed algorithm's performance in the last chapter.

As previous defined, the vehicles motion over the marine area depend on the control input u. The control input u is obtained through simple rules ([44], [60]). In particular, it is defined per each vehicle by refining the algorithm described in [47], [13] for oceanographic and security applications. Specifically, each vehicle moves in accordance to the following two rules of movement:

1. move towards the waypoint;

2. move away from neighbours and coasts.

Therefore the vehicles will be able to avoid collisions with coast or other vehicles while reaching the waypoints.

The above two rules are enforced by defininig two force, respectively an *attraction* and *obstacle avoidance* function, as described in the potential function theory. Specifically, they are defined as:

1. The *attraction function* $h_A(x(t), x_W)$ is a function of the vehicle's distance between its current position $x(t)$ and the waypoint x_W. It therefore defines the interest of each vehicle to move towards its goal. Formally, the function is defined as:

$$h_A(x(t), x_W) = \frac{1}{2}\,\mu\, d(x(t), x_W)^2 \tag{4.23}$$

where, μ is a scalar multiplier and $d(\cdot)$ is a measure of distance between the two locations.

2. The *obstacle avoidance function* $h_O(x(t), Q_{obs})$ is a function of the vehicle's distance from either coasts or vehicles neighbours. Q_{obs} is the union of the obstacle areas contained in the exploration region assigned to the vehicle. This function defines the interest to move away from the obstacles enforcing the second rule. Formally, it is written as:

$$h_O(x\,(t)\,, Q_{obs}) = \begin{cases} \frac{1}{2}\sum_{i=1}^{N_{obs}} \nu \left(\frac{1}{d\left(x(t), Q^i_{obs}\right)} - \frac{1}{q^i_*} \right)^2, & \text{if } d\,(x\,(t)\,, Q^i_{obs}) \leq q^i_* \\ 0, & \text{if } d\,(x\,(t)\,, Q^i_{obs}) > q^i_* \end{cases} \tag{4.24}$$

where, ν is a scalar multiplier, $d\,(x\,(t)\,, Q^i_{obs})$ is the minimum distance between the vehicle location $x\,(t)$ and the i-th obstacle, q^i_* is the domain of influence of the i-th obstacle and N_{obs} is the total number of obstacles over the exploration region.

Finally, the vehicle control input $u(t)$, at each time instant, is obtained as the vector sum of the gradient of each defined function:

$$u\,(t) = \nabla h_A\,(x\,(t)\,, x_W) + \nabla h_O\,(x\,(t)\,, Q_{obs}) \tag{4.25}$$

4.5 Updating and Tuning of the maps

The optimal search of the vehicles' waypoints requires the updating of the prior density map during the mission. Here, a mathematical method, which uses the sample data gathered by the vehicles' looking-sensor, is defined. Two different maps will be employed: the first to compute the waypoints during the search mission, while the second one to build a posteriori density map over the marine area of interest. Note that the posteriori map can be computed either during the exploration or when the search mission comes to the end.

The proposed method to update the a priori map requires the knowledge of the vehicles' paths or in equivalent mode the sharing of sample data gathered by the vehicles' looking-sensor, while the method to build the a posteriori map requires the knowledge of the findings along the vehicles' route. Therefore the vehicles communicate the sample data gathered among themselves at regular time instants. Although the underwater acoustic communication may not be continously allowed, in this work the communication is supposed possible everywhere. For instance, the problem of communication could be overcome if the vehicles come back up out of the water transmiting information by the radio modem. Hence, under the above assumptions the vehicles are able to share the sample data set and their own executed routes at regular time instants. A priori and a posteriori density maps are estimated by using the refined Parzen method described in chapter 3. In particular, in the updating of the a priori density map is used the Eq. 3.4 adding one negative Gaussian kernel per each point that the vehicles have explored. Formally, given the set $\mathbb{X}_{exp} = \{X_k\}$ of M points explored X_k, $k = 1, ..., M$

until the time instant t, the a priori map is obtained by the formula:

$$\hat{p}(\pi) = \max\left(0, \frac{1}{N_\pi}\sum_{i=1}^{N_{data}}\sum_{j=1}^{m_i}\phi_1(\pi - \pi_{ij}, \Sigma_\pi) - \frac{1}{M}\sum_{k=1}^{M}\phi_2(\pi - X_k, \Sigma_X)\right) \tag{4.26}$$

where $\phi_1(\cdot)$ is the bivariate Gaussian kernel with mean π_{ij} and covariance matrix Σ_π, $N_\pi = \sum_{i=1}^{N_{data}} m_i$ is the cardinality of the particles set $\{\pi_{ij}\}$ obtained from the historical data set as described in the first step of the algorithm summarized in Table 3.1 and $\phi_2(\cdot)$ is the bivariate Gaussian kernel with mean X_k and covariance matrix Σ_X. Moreover, the covariance matrix Σ_X (the *bandwidth* matrix) associated with the set $\mathbb{X}_{exp} = \{X_k\}$ of points is computed with the same method applied for the particle set $\{\pi_{ij}\}$, as described in section 3.2.

Starting from the sample data set used for the updating of the a priori density map also the a posteriori density map is built. In particular the data set $\mathbb{X}_{exp} = \{X_k\}$ of M points explored is divided into two subset, the set $\mathbb{X}_{find} = \{X_p\}$ of P findings and the set $\mathbb{X}_{not} = \{X_q\}$ of Q not findings, respectively. Then the posteriori map is built by executing the steps 2 and 3 of the algorithm in Table 3.1 using the data set $\mathbb{X}_{find}$ as particles.

4.6 Simulative results

The proposed cooperative distributed algorithm based on the minimum Entropy approach is now tested and evaluated in simulation. The cooperative algorithm is performed on the a priori density map corresponding to the Tuscan Archipelago, as built in section 3.3,

exploring this marine area by a team of three autonomous underwater vehicles AUVs. The simulative code is written in MATLAB and makes use of the C++ Computational Geometry Algorithms Library CGAL [1].

The scenario of the simulation study is represented by a region of $80\,km \times 140\,km$. Each vehicle moves at the maximum speed of $5\,knots$ and it is supposed equipped with a side-scan sonar for detection scanning, plus an acoustic and radio modems for cooperative communication and localization.

The purpose of the simulation is to check the capability of the cooperative distributed algorithm, not that of the sensors, nor that of the AUVs employed. In this respect, one simplifying assumption is made on the operability of the side scan sonar. In particular, the sonars are assumed without mistakes during their detection scanning but they give measurements with *uncertainty*. Therefore, every time a vehicle moves over a marine area with objects resting on the seabed the sonar always detects the whole objects but nevertheless the detected positions of the findings are uncertain. Thus a bi-dimensional Gaussian random variable directed as the sonar is associated on the measuring. In fact the random variable is able to represent the confidence in the knowledge of the position of a findings (or not) and how such position is uncertain. Hence, the model depicted in Figure 4.4 for the sonar measurements is given mathematically as

$$p_{S_i} \sim N(\pi_i, \Sigma_{\pi_i}), \quad i = 1,\, 2,\, 3.$$
$$\Sigma_{\pi_i} = A \cdot A^T, \quad A = \begin{bmatrix} \cos\alpha_i & -\sin\alpha_i \\ \sin\alpha_i & \cos\alpha_i \end{bmatrix} \cdot \begin{bmatrix} sss_x & 0 \\ 0 & sss_y \end{bmatrix} \tag{4.27}$$

where, π_i is the geographical position of the $i-$th vehicle, α_i is the

4.6 Simulative results

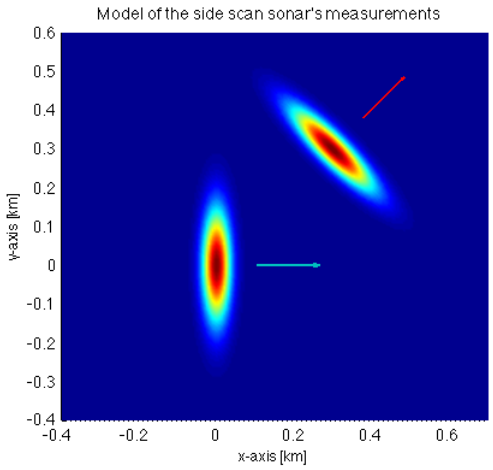
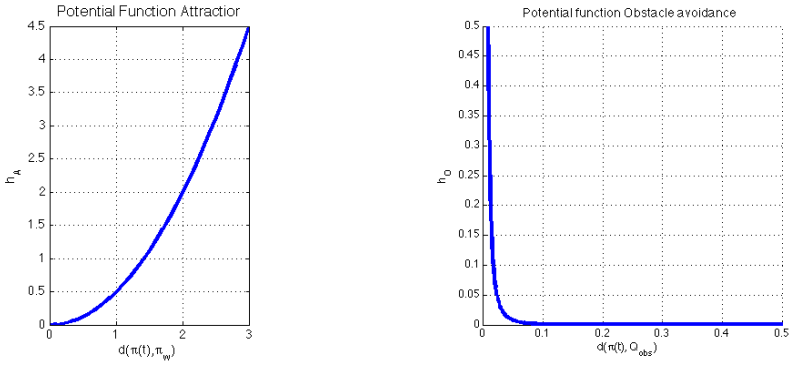


Figure 4.4: *The model of the side scan sonar's measurement. The red and blue arrows refer to the vehicle heading.*

heading angle of the $i-$th vehicle and sss_x, sss_y are the dimensions of the beam pattern, which is assumed rectangular with an aperture of $200\,m \,\mathrm{x}\, 50\,m$ on the ground, along sway and surge axes respectively. In order to this assumption the sonars measurements gathered throughout the exploration task will be obtained by sampling a certain number of particles $\pi_{S_{ij}}$, $j = 1, ..., m_i$ from p_{S_i} at each time frame. The number m_i will be set up with value 1 or 10 in accordance to the objects resting on the seabed and the resolution of the region to be surveyed. Specifically m_i assumes value 1 when the sonar finds an object, and value 10 in other cases. Furthermore all the particles extracted from each vehicles will be arranged in the data set $\mathbb{X}_{exp}$, which also will be able to divide into two data subset $\mathbb{X}_{find}$ and $\mathbb{X}_{not}$ of findings and not findings respectively.

Before starting the mission, initial conditions are configured as

following (see Figure 4.6): vehicles position is chosen within the bound of the environment Q in the North-East portion of the map, the Equitable diagram weights are set to zero and the a priori density map $\hat{p}(\pi)$ is built one time with a maximum of five samples selected for K_{max}, according to the algorithm described in chapter 3. Moreover, potential functions to enforce the two rules of movement for each vehicle - avoid collisions while some vehicle reaching the waypoint - as detailed in section 4.4, are defined as Figure 4.5 shows, plotted against the distance term $d(\pi(t); \pi_w)$ and $d(\pi(t); Q_{obs})$ respectively.



(a) (b)

Figure 4.5: *The obstacle avoidance (left) and attraction (right) potential functions, plotted against the distance terms.*

During the exploration task, the vehicles move toward the local optimal waypoints, in accordance with the minimum entropy criterion, scanning the followed routes with the side scan sonar. At the same time, each vehicle communicates its own routes and the found

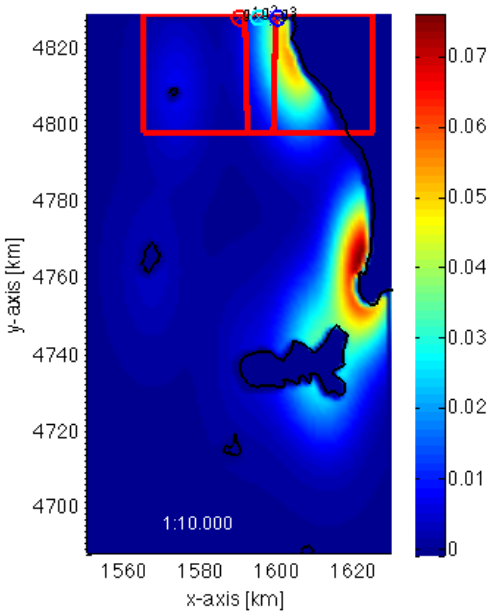


Figure 4.6: *The starting a priori density map in the Tuscan Archipelago via kernel density estimation and the initial positions of the vehicles. The straight red lines represent the Power Equitable diagram, the crosses are the waypoints and the circles are the vehicles position.*

relicts to the other vehicles. Hence, the equitable power diagram is updated dynamically as the mission is in progress, together with the a priori density map, using the side scan sonar measurements. Figure 4.8 shows the realization of the exploration task performed starting from the North-East portion of the map, and Figure 4.7 depicts an update of the equitable power diagram together with the a prior density map at the time when the first waypoint was reached by every vehicle. In particular, Figure 4.8(a) shows the status of

the task at an intermediate point, while Figure 4.8(b) the complete paths of the vehicles after the exploration came to an end. It is worth to note that the routes of the vehicles follow the directions in order to maximize the a priori expected percentage of detections that corresponds, in equivalent fashion, to minimize the Renyi's entropy.

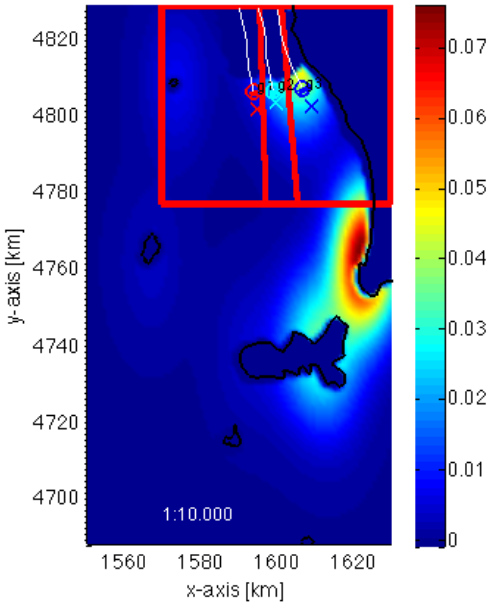
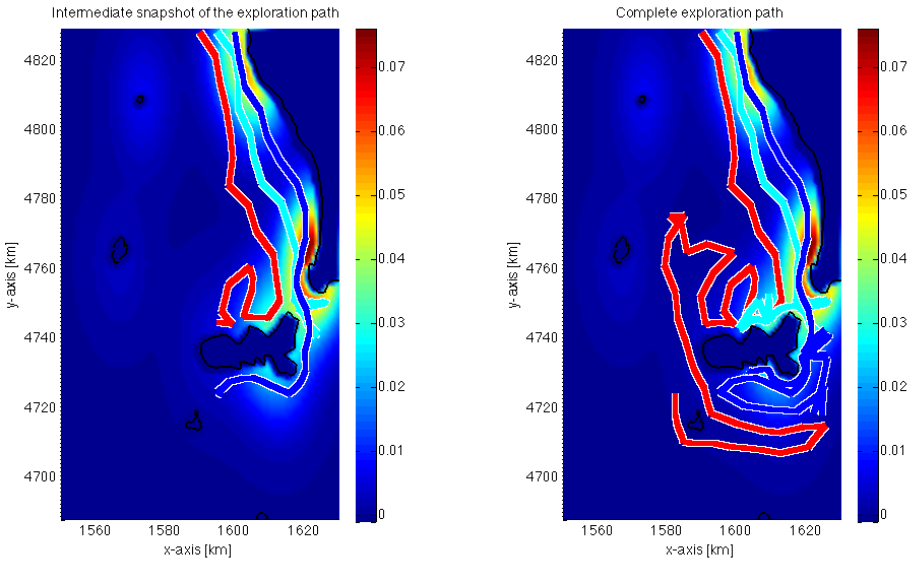


Figure 4.7: *The updated a priori density map, at the time when the first waypoint was reached by every vehicle. The straight red lines represent the Power Equitable diagram, the crosses are the waypoints and the circles are the vehicles position.*

The exploration task finish discovering about 90 percent of the total relicts and covering around only 40 percent of the total marine area. Additionally, after the end of the exploration the a posteriori



(a) (b)

Figure 4.8: *Paths of the vehicles during the exploration task. (a) Status of the mission at an intermediate point; (b) complete paths. The red, cyan and blue lines are the paths of the three vehicles*

map is tuned (see Figure 4.9) in accordance with the mathematical model defined in section 4.5.

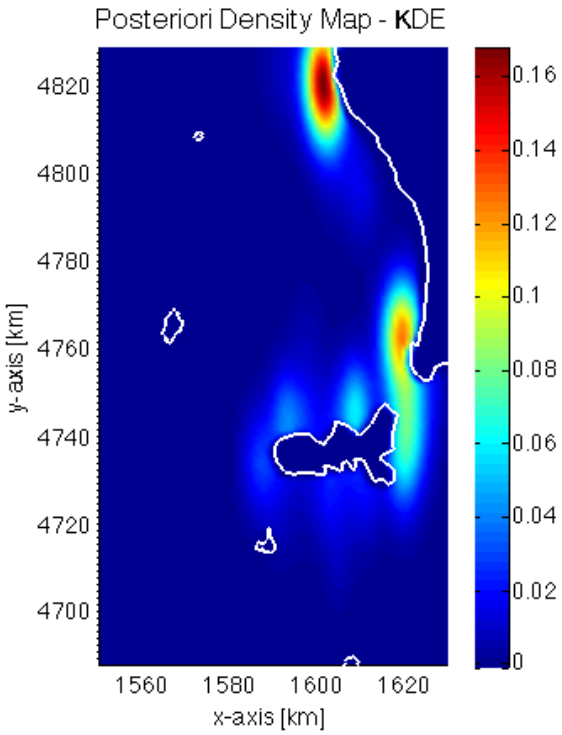


Figure 4.9: *The a posteriori density map built using the side scan sonars measurements.*

Chapter 5

Multi-Agents RRT* Algorithm

Motion coordination is a problem of finding a set of non-conflicting trajectories for several mobile agents, such as autonomous aircrafts, cars, or underwater vehicles. The systematic search of object resting on, or buried in the seabed by a team of AUVs requires a cooperative path-finding approach to perform the complex tasks efficiently. Therefore the AUVs seabed surveying can be modelled as an instance of motion coordination.

Motion coordination in terms of planning has been a highly active area of research since the late 1970s. Early algorithms relied on some heuristic forward-search path-finding technique, such as A* [39], but during the last decade, incremental sampling-based motion planning algorithms, such as the Rapidly-exploring Random Trees (RRTs), have been shown to work better in practice and to possess theoretical guarantees [39]. Recently, Karaman and Frazzoli [32]

have also introduced a novel any-time sampling-based motion planning algorithm for single agent, called the RRT* algorithm, which offers good scalability to high-dimensional environments and at the same time it guarantees convergence to an optimal solution. In this chapter, a novel Multi-RRT* algorithm is proposed: an algorithm for multi-agent path planning that builds upon the RRT*. An extension of RRT* to multi-agent path planning was recently proposed by Cáp [17]; with respect to the latter, the algorithm proposed here introduces distance constraints on the tree structure. The algorithm is developed for AUV seabed surveys, exploring the same regions of the cooperative distributed algorithm (defined in chapter 4), and taking in a centralized way the decision on optimal AUV paths via minimization of the Renyi's entropy computed over the assigned marine area. The proposed Multi-RRT* algorithm is expected to overcome the issues of the standard planning methods (zig-zag or lawn-mower paths) exploring the marine area more efficiently. In simulative results it will be shown that the Multi-RRT* algorithm outperforms the standard search approach, as expected, and that has similar performances but not better with respect to the cooperative distributed algorithm defined in chapter 4.

5.1 Problem Formulation

This section is devoted to introduce notation and problem formulation for the Multi-RRT* algorithm, which will be defined in detail later on. In particular, first the path planning is formalized taking into account the background of section 2.3.3, and then the optimality

problem is defined.

A team of n vehicles explores an assigned marine area. According to planning theory the marine area to be surveyed is modelled formally by a bounded connected open subset X of $\mathbb{R}^d$, where $d \in \mathbb{N}$ and $d = 2$. Thus the *obstacle region* X_{obs} and the *goal region* X_{goal} are open subsets of X. The *obstacle-free space* is denoted, i.e., $X \setminus X_{obs}$, as X_{free} and the *initial state* for i-th agent, x^i_{init}, is an element of X_{free}. Moreover, a path in X with length $s \geq 0$ is defined as a continous function $\sigma : [0,\ s] \to X$, where s is considered in the usual way as $\mathsf{sup}_{\{n\in\mathbb{N},\ 0=\tau_0<\tau_1<\ldots<\tau_n=s\}} \sum_{i=1}^{n} \|\sigma(\tau_i) - \sigma(\tau_{i-1})\|$. In the sequel, a path in X_{free} is said to be a *collision-free* path and n-collision-free path will have to be separated avoiding collision between the vehicles, that in mathematical form is

$$\forall j; k; t : j \neq k => \mathrm{dist}(\sigma_i[t]; \sigma_j[t]) > d_{sep}$$

where, d_{sep} is the required minimum separation distance among the vehicles. Thus, the i-th vehicle's collision-free path that starts at x^i_{init} and ends in the goal region with respect to the minimum separation distance is said to be a *feasible path*, i.e., a collision-free path $\sigma : [0, s] \mapsto X_{free}$ is feasible if and only if

i) $\sigma(0) = x^i_{init}$

ii) $\sigma(s) \in X_{goal}$

iii) $\forall j, k, t : j \neq k => \mathrm{dist}(\sigma_i[t]; \sigma_j[t]) > d_{sep}$

Finally, a cost function is also defined, $\mathsf{c} : \sum_{X_{free}} \mapsto \mathbb{R}_{>0}$. It assigns a non-negative cost to all nontrivial collision free paths computing

the cost regarding the maximization of the information gain, i.e. the Renyi's cross-entropy between the a priori map built over the marine area to be surveyed and the paths. The *optimality problem* of path-finding is to find a feasible path with minimum cost, if one exists, and report failure otherwise.

Problem 1 (Optimal multi-agents planning) *Given a bounded connected open set X, an obstacle space X_{obs}, an initial state x^i_{init}, a goal region X_{goal} and a minimum separation distance d_{sep}, find a path for i-th agent $\sigma^*_i : [0, s] \mapsto \mathsf{cl}(X_{free})$, where $\mathsf{cl}(X)$ is the closure of X, such that (i) $\sigma(0) = x^i_{init}$ and $\sigma(s) \in X_{goal}$,* (ii) $\forall j, k, t : j \neq k => \mathrm{dist}(\sigma_i[t]; \sigma_j[t]) > d_{sep}$ *and (iii)* $\mathsf{c}(\sigma^*) = \min_{\sigma \in \sum_{\mathrm{cl}(X_{free})}} \mathsf{c}(\sigma)$. *If no such path exists, then report failure.*

5.2 Multi-RRT* Algorithm

A novel optimal sampling-based algorithm for multi agent system, called the Multi-RRT* algorithm, is now introduced. Some common primitive procedures are defined before to provide the Multi-RRT*. It is worth to note that the proposed algorithm is representative of paradigms for optimal sampling-based motion planning algorithms since it is an extension of the single agent RRT*.

5.2.1 Primitive Procedures

Before detailing the Multi-RRT* algorithm the primitive procedures that it relies on, are defined.

Sampling: The function $\mathsf{Sample} : \mathbb{N} \mapsto X_{free}$ returns independent identically distributed (i.i.d.) samples from X_{free}.

Sharing: Given a set of data Γ, the function $\mathsf{Sharing} : \Gamma \to \{g_i\}$ communicates the entire set of information between the vehicles $\mathbb{G} = \{g_i\},\ i = 1, \ldots, N_v$.

Nearest Neighbor: Given a graph $G = (V, E)$ and a point $x \in X_{free}$, the function $\mathsf{Nearest} : (G; x) \mapsto v$ returns a vertex $v \in V$ that is ''closest'' to x in terms of a given distance function. The Euclidean distance will be used, i.e.,

$$\mathsf{Nearest}(G = (V, E), x) = \text{argmin}_{v \in V}\{\|x - v\|\}$$

Near Vertices: Given a graph $G = (V, E)$, where $V \subset X$, a point $x \in X$, and a positive real number $r \in R_{>0}$, the function $\mathsf{Near} : (G, x, r) \mapsto V' \subseteq V$ returns the vertices in V that are contained in a ball of radius r centered at x, i.e.,

$$\mathsf{Near}(G = (V, E), x, r) := \{v \in V \ : \ v \in \mathcal{B}_{x,r}\}$$

Steering: Given two points $x, y \in X$, the function $\mathsf{Steer} : (x, y) \mapsto z$ returns a point $z \in X$ such that z is "closer" to y than x is. Throughout the section, the point z returned by the function Steer will be such that z minimizes $\|z–y\|$ while at the same time maintaining $\|z - x\| \leq \eta$, for a prespecified $\eta > 0$, i.e.,

$$\mathsf{Steer}(x; y) := \text{argmin}_{z \in \mathcal{B}_{x,\eta}}\|z–y\|$$

Collision Test: Given two points $x, x' \in X$, the Boolean function $\mathsf{CollisionFree}(x, x')$ returns True if the line segment between x and x' lies in X_{free}, i.e., $[x, x'] \subset X_{free}$, and False otherwise.

Line: Given two points $x_1, x_2 \in \mathbb{R}^d$, the function $\mathsf{Line} : (x_1,\ x_2) \mapsto X$ denotes the straight continuos path from x_1 to x_2.

Parent: Given a tree $G = (V, E)$, $\mathsf{Parent} : V \mapsto V$ is a function that maps a vertex $v \in V$ to the unique vertex $u \in V$ such that $(u, v) \in E$. By convention, if $v_0 \in V$ is the root vertex of G, $\mathsf{Parent}(v_0) = v_0$.

Cost: Given a tree $G = (V, E)$, $\mathsf{Cost} : V \mapsto R_0$ is a function that maps a vertex $v \in V$ to the cost of the unique path from the tree's root to v. For simplicity in stating the algorithm, an additive cost function will be assumed, so that $\mathsf{Cost}(v) = \mathsf{Cost}(\mathsf{Parent}(v)) + \mathsf{c}(\mathsf{Line}(\mathsf{Parent}(v); v))$. By convention, if $v_0 \in V$ is the root vertex of G, then $\mathsf{Cost}(v_0) = 0$.

Minimum Cost Walk: Given a set of trees $\mathbb{G} = \{G_i\}$, the function $\mathsf{MinimumPath}(G)$ returns a set of paths computed one per each trees of length between the medium and the maximum length among the all possible paths and with minimum cost.

Edges Minimum Distance: Given a set of trees $\mathbb{G} = \{G_i\}$, one new edge per each tree $\mathbb{E} = \{E_i\}$, $n \in \mathbb{N}$ and a minimum separation distance d_{sep}, the Boolean function $\mathsf{EdgesMinimumDistance}(G,\ E,\ n,\ d_{\text{sep}})$ returns True if the entire new edges $\mathbb{E}$ lie at the distance d_{sep} or more from the last n edges of the trees $\mathbb{G}$, and False otherwise.

5.2.2 Proposed Algorithm

Multi-RRT* algorithm is similar to other incremental sampling-based planning algorithms such as PRM, RRG, RRT, RRT*. It take as input a path planning problem $(X_{free};\ \{x^1_{init},\ \ldots,\ x^n_{init}\};\ X_{goal})$, an integer $N_v \in \mathbb{N}$ (number of agents), a cost function $c : \sum_{X_{free}} \mapsto \mathbb{R}_{>0}$ and initially, it start with N_v graph, one for each agent. At the beginning each graph has a single vertex, which corresponds to the

initial state x^i_{init}, and no edges. Then, the Multi-RRT* incrementally grow the N_v graphs on X_{free} by sampling states $X_{rand} = \{x^1_{\text{rand}}, \ldots, x^{N_v}_{\text{rand}}\}$ from X_{free} at random and extending the graphs towards X_{rand}. At the end of $N \in \mathbb{N}$ *iterations* the Multi-RRT* algorithm return a set of trees, $G = \{G_1, \ldots, G_{N_v}\}$, where $\forall i$, $i = \{1, ..., N_v\}$, $V_i \subset X_{free}$ and $E_i \in V_i$ x V_i. At this point the path-planning for a single $i-$th agent can be described with the corresponding tree, $G_i = (V_i, E_i)$, where the vertices represent the positions in X_{free} and the edges define the optimal feasible path. Therefore Multi-RRT* resolves the optimal multi-agents planning problem.

The detailed procedure of the Multi-RRT* algorithm is presented in Algorithm 2. First, Multi-RRT* extends the nearest neighbour towards the sample per each graph (Lines 7-8). Then, it connects the new vertex per each i-th graph, x^i_{new}, to the vertex that incurs the minimum accumulated cost of the unique path up until x^i_{new}, and lies within the set X^i_{near} of vertices returned by the Near procedure (Lines 10-18). Notice that the cost of the unique path from x^i_{init} to a new vertex $v^i \in V_i$ is denoted by the function Cost(v). Finally, if the all extensions of the entire N_v graphs result in collision-free paths respecting the minimum separation distance (Line 22) are added to the graphs as edges, and their terminal points as new vertices (Line 24). Multi-RRT* also extends per each i-th graph the new vertex to the vertices in X^i_{near} in order to "rewire"the vertices that can be accessed through x^i_{new} with smaller cost (Lines 25-29). It is worth to note that the function EdgesMinimumDistance in Line 22 checks the minimum separation, d_{sep}, between the entire computed new edges and the last three edges of each optimal feasible

path. Only the last three edges and no the whole optimal feasible path are checked because the collision between the agents could happen when they explore close regions in X_{free}. Obviously, there will be no collision between agents if one optimal path is crossed by a new edge before of the its own last three edges, because the agents will be move on the point of intersection at different time frame.

5.2 Multi-RRT* Algorithm

Algorithm 2: Summary of the Multi-RRT* algorithm

$V_i \leftarrow \{x^i_{\text{init}}\}$, $E_i \leftarrow \emptyset$, $G_i \leftarrow (V_i, E_i)$, $i = 1, ..., N_v$;
for $k \leftarrow 1$ **to** N **do**
  $G \leftarrow \{G_1, ..., G_{N_v}\}$;
  for $i \leftarrow 1$ **to** N_v **do**
    $G_i \leftarrow (V_i, E_i)$;
    $x^i_{\text{rand}} \leftarrow \mathsf{Sample}(k)$;
    $x^i_{\text{nearest}} \leftarrow \mathsf{Nearest}(G_i, x^i_{\text{rand}})$;
    $x^i_{\text{new}} \leftarrow \mathsf{Steer}(x^i_{\text{nearest}}, x^i_{\text{rand}})$;
    if $\mathsf{CollisionFree}(x^i_{\mathsf{nearest}}, x^i_{\mathsf{new}})$ **then**
      $x^i_{\text{min}} \leftarrow x^i_{\text{nearest}}$;
      $X^i_{\text{near}} \leftarrow \mathsf{Near}(G_i, x^i_{\text{new}}, r)$;
      forall the $x^i_{\mathsf{near}} \in X_{\mathsf{near}}$ **do**
        if $\mathsf{CollisionFree}(x^i_{\mathsf{near}}, x^i_{\mathsf{new}})$ **then**
          $c_i \leftarrow \mathsf{Cost}(x^i_{\text{near}}) + \mathsf{c}(\mathsf{Line}(x^i_{\text{near}}, x^i_{\text{new}}))$;
          if $c^i < \mathsf{Cost}(x^i_{\mathsf{new}})$ **then**
            $x^i_{\text{min}} \leftarrow x^i_{\text{near}}$;
    else
      $x^i_{\text{new}} \leftarrow \emptyset$;
  if $x^i_{\mathsf{new}} \neq \emptyset$, $\forall\, i = 1, ..., N_v$ **then**
    $\mathsf{Sharing}(G, X_{\text{min}}, X_{\text{new}})$;
    $\hat{G} \leftarrow \mathsf{MinimumPath}(G)$;
    if $\mathsf{EdgesMinimumDistance}(\hat{G}, \{X_{min}, X_{new}\}, 3, d_{sep})$ **then**
      for $i = 1, ..., N_v$ **do**
        $V_i \leftarrow V_i \cup \{x^i_{\text{new}}\}$; $E_i \leftarrow E_i \cup \{(x^i_{\text{min}}, x^i_{\text{new}})\}$;
        forall the $x^i_{\mathsf{near}} \in X^i_{\mathsf{near}} \setminus \{x^i_{\mathsf{min}}\}$ **do**
          if $\mathsf{CollisionFree}(x^i_{\mathsf{near}}, x^i_{\mathsf{new}})$ *and* $\mathsf{Cost}(x^i_{near}) >$ $\mathsf{Cost}(x^i_{new}) + \mathsf{c}(\mathsf{Line}(x^i_{new}, x^i_{near}))$ **then**
            $x^i_{\text{parent}} \leftarrow \mathsf{Parent}(x^i_{\text{near}})$;
            $E_i \leftarrow E_i \setminus \{(x^i_{\text{parent}}, x^i_{\text{near}}\}$;
            $E_i \leftarrow E_i \cup \{(x^i_{\text{new}}, x^i_{\text{near}}\}$;

5.3 Simulation and Results

The Multi-RRT* algorithm described in the preceding sections is now tested and evaluated in simulation. Multi-RRT* algorithm is performed on the a priori density map corresponding to the Tuscan Archipelago, as built in section 3.3, solving the optimality problem of path-finding for a team of three autonomous underwater vehicles (AUVs). The simulative code is written in MATLAB.

The scenario of the simulation study is represented by a region of $80\,km\,x\,140\,km$ width, which corresponds to the Tuscan archipelago. Before of starting the simulation, the initial conditions are configured. In particular, the vehicles position is chosen within the bound of the environment Q and the a priori map is built one time with a maximum of five samples selected for K_{max}, according to the algorithm described in chapter 3.

During the exploration task, the vehicles move toward the waypoints, scanning the followed routes by the side scan sonar. All the vehicles' waypoints are optimized offline in accordance with the Multi-RRT* algorithm, before that the exploration mission begins. Figures 5.1 and 5.2 show the solved path-finding by the Multi-RRT*, starting from the East portion of the map. The first Figure shows the application of the Multi-RRT* at an intermediate point, while the second one shows the complete status of the algorithm. Defined the waypoints and the paths offline, Figure 5.3 depicts the complete real paths of the vehicles, after the exploration came to an end. As Figure 5.3 clearly shows, the routes of the vehicles are directed along the maxima of the a expected percentage of detections, or, in an equivalent fashion, along the minima of the relative Renyi's

5.3 Simulation and Results

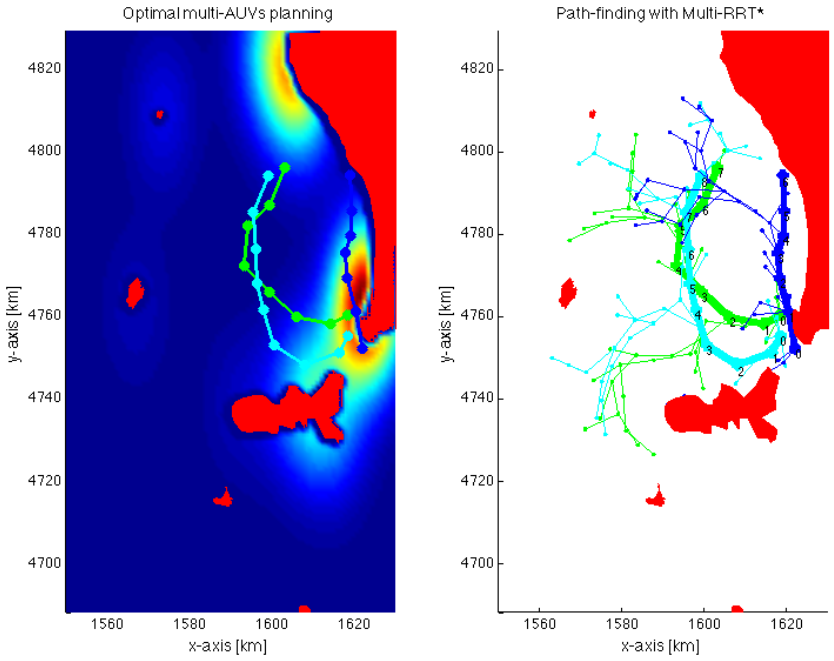


Figure 5.1: *Trees and optimal paths-finding at an intermediate point during the application of the Multi-RRT* algorithm. The green, cyan and blue lines are the trees of the three vehicles.*

entropy.

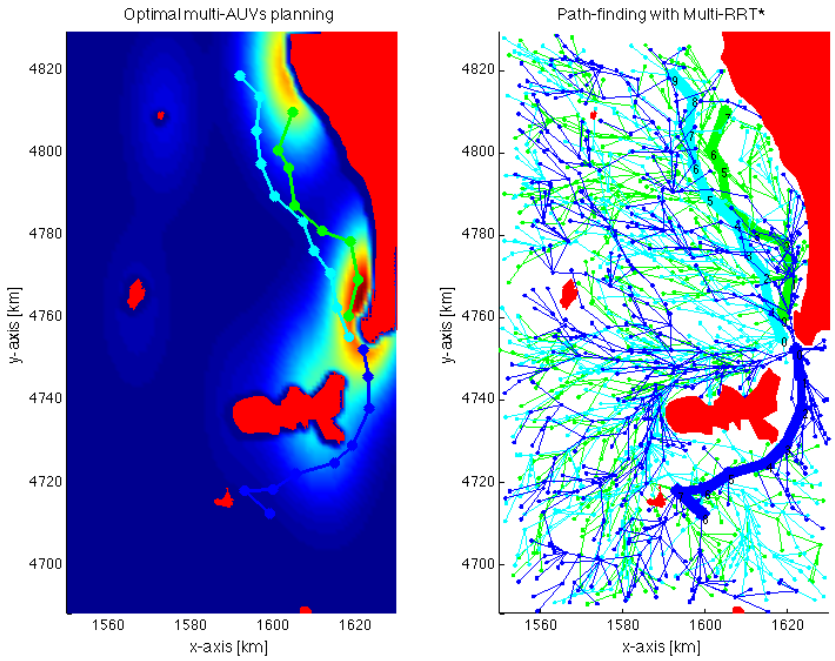


Figure 5.2: *Complete trees and optimal paths-finding when the Multi-RRT* algorithm comes to end. The green, cyan and blue lines refer to the trees of the three vehicles.*

5.3 Simulation and Results

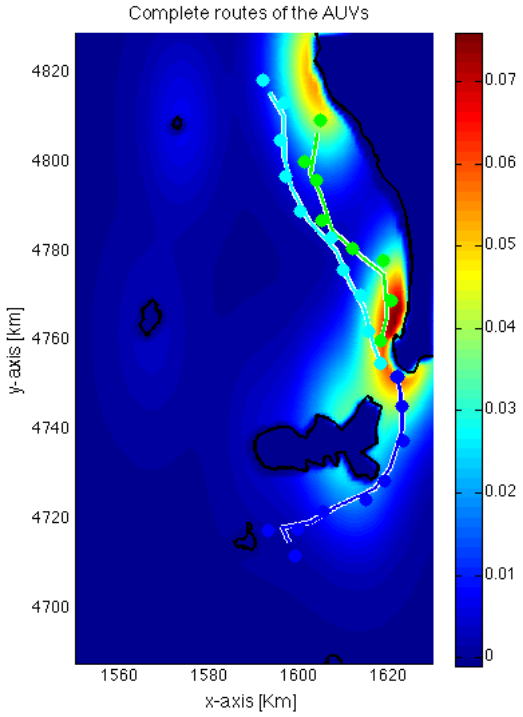


Figure 5.3: *Complete real paths of the vehicles during the exploration task. The green, cyan and blue lines refer to the paths of the three vehicles.*

Chapter 6

Performance for exploration algorithms

The design of an effective performance measurement system, which includes the selection of appropriate measures and approaches for analysing results, is central to evaluate either the effectiveness of search algorithms or the comparison among different feasible algorithms. Despite its importance, this is one area where the literature is poor. Thus, this chapter suggests some indicators of measure to evaluate and/or compare the previously proposed algorithms for AUV seabed surveying.

6.1 Introduction

The proposed cooperative algorithms in chapters 4 and 5 with the building of the a priori density map resolve the systematic search of objects resting on (or buried in) the seabed of marine areas by teams

of AUVs. At this point, some questions on performance arise: What is the best algorithm? How is the quality evaluated? What are the algorithm's vantages and/or advantages?

In literature there are few attempts [26] for evaluating the systematic search with teams of AUVs, thus this section focuses on the evaluation of proposed algorithms' vantages and advantages by a performance measurement system. Performance measurement system selects appropriate measures for the data of results to be analyzed. The selected measures are chosen in accordance with characteristics or proprieties of interest. For instance, in this work the main interesting metrics may be: the total mission time, the explored area over the total available area, the number of findings, the time to discover the first n findings, the time to discover the total number of findings and the number of findings on the total mission time. It is remarked that, while the comparison is done in simulation, the data on the objects resting on the seabed are real. Thus, while vehicle and sensor performance can affect performance in real situation, our simulation allows to compare just the motion planning approaches over a real search scenario. It is remarked also that, to allow this comparison, we have taken roughly half of the entries in the historical data base to build the a priori map, and the remaining entries as validation data in the metric performance comparison.

6.2 Performance measurement system

The performance measurement system selects appropriate measures and defines approaches to analyse results of algorithms, novel sys-

tems, methods of optimization. In this thesis, two novel approaches are designed for AUV seabed surveying as explained in chapters 4 and 5. Consequently, the performance measurement system to analyse the algorithms' results of this work has based on three measures: findings, mission time, explored marine area. In detail the employed indicators of measures are:

- the fraction of explored area over the total available area;
- the number of findings over the total number of objects;
- the total mission time.

The performance measurement system even allows to depict graphs of performance to better explain the advantages and disadvantages of each algorithm combinating these three main indicators among (i.e. plotted against) each other. In this context, for an accurate analysis, are defined the optimal couples with more logical significance as

- the rate of found relicts with respect to the mission time;
- the fraction of explored area over the total available area with respect to the mission time;
- the fraction of explored area over the total available area with respect to the rate of found relicts.

Defined the performance measurement system the next section is thus devoted to illustrate the evaluation of the different algorithms.

6.3 Analysis of results

The cooperative distributed algorithm, the multi-agents RRT* algorithm and the standard search (regular lawn-mower transects) are now tested in simulative scenarios. The code is written in MATLAB and makes use of the C++ Computational Geometry Algorithms Library CGAL [1].

The objective of the all missions is to find the objects resting on, or buried in, the seabed of the Tuscan Archipelago, Northern Tyrrhenian Sea. The marine area is $80km$ x $140km$ width with the data shows by $100mt$ x $100mt$ cell resolution, as Figure 6.1 depicts.

A team of three vehicles performs the missions. The vehicles move at the maximum speed of $2.5m/s$ (five $knots$), and each vehicle is equipped with a side scan sonar SSS for detection scanning and with a network device that allows communication between any two vehicles in the team. The cooperative distributed algorithm and the multi-RRT* algorithm are tested via Monte Carlo simulations since the vehicles trajectory depend upon their initial conditions that, in turn, affect the selection of the waypoints during the search mission. Hence, as expected, the initial conditions have an impact on the performances of the cooperative algorithm and the Multi-RRT* algorithm. Figure 6.2 gives an example of the realization of the Monte Carlo evaluation of the algorithms, in which the vehicles starting positions are highlighted. Each group of three points of the same color corresponds to the initial position of the team of the three vehicles, at each simulation run. On the other hand, only one simulation, called the "lawn-mower-optimized", is performed for the standard search (regular lawn-mower transects). "Lawn-mower"because the vehicles

6.3 Analysis of results

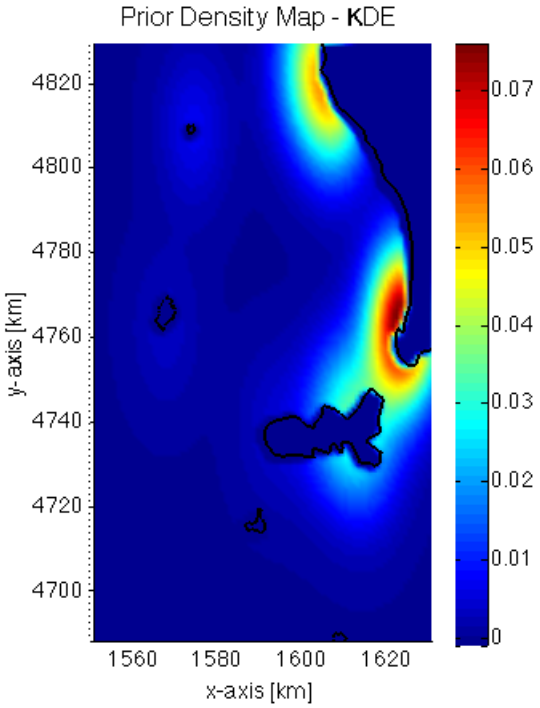


Figure 6.1: *The a priori density map over a sub-region of the Tuscan Archipelago to be surveyed.*

moves with zig-zag or lawn-mower transects and "optimized"since the waypoints are selected offline in according to the a priori density map. Figure 6.4 shows the "lawn-mower-optimized" pattern. However, all the testing use the same a priori density map depicted in Figure 6.1. Finally, the simulative missions by either cooperative distributed algorithm or multi-RRT* algorithm come to end whenever there is not a new waypoint or, in other case, when the particles set Υ_π - i.e. the set used to compute the prior map, Equation 3.4

- becomes empty, due to every time the vehicles explore a region in which some of the sampled particles π_{ij} are localized, the findings are removed from the set Υ_{π}. Nevertheless the simulation of the "lawn-mower-optimized"search is completed in about 120 hours because it must explore the whole marine area, being a method exhaustive in nature.

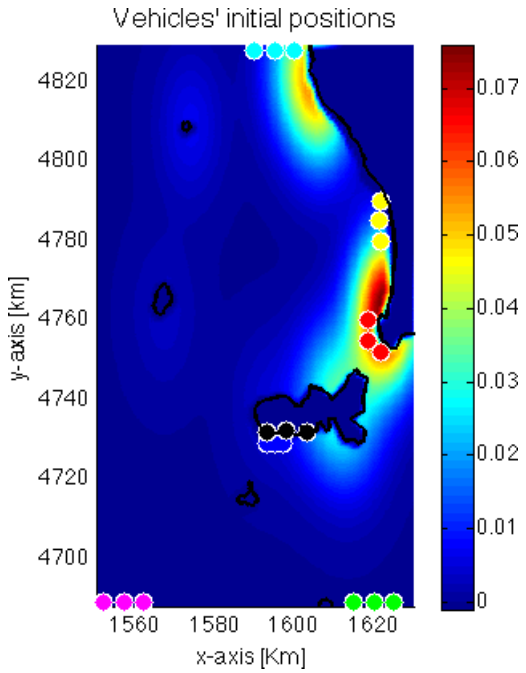


Figure 6.2: *Vehicles' positions used as the initial conditions in the simulations of the systematic search strategies. The dots refer vehicles' positions, the colours to the different executed simulation.*

The purpose of the simulations is to evaluate the algorithms performance and the vantages of the proposed novel cooperative approach for AUV seabed surveying, not that of the sensors, nor that

of the AUVs employed. In this respect, two simplifying assumptions are made, and applied to all the three tested methods. In particular, all the vehicles in the team are assumed that move at constant depth and their own side scan sonars do not make mistakes. Specifically, every time a vehicle moves over a marine area with objects resting on the seabed the sonar always detects the whole objects but nevertheless the detected positions of the findings are uncertain. Thus a bi-dimensional Gaussian random variable directed as the sonar is reasonable to associate on the measuring. In fact the random

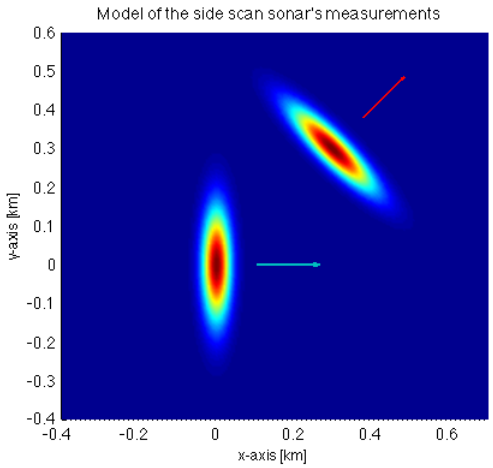


Figure 6.3: *The model of the side scan sonar's measurement. The red and blue arrows refer to the vehicle heading.*

variable is able to represent the confidence in the knowledge of the position of a findings (or not) and how such position is uncertain. Hence, the model for the sonar measurements depicted in Figure 6.3

is mathematically the following:

$$p_{S_i} \sim N(\pi_i, \Sigma_{\pi_i}), \quad i = 1,\, 2,\, 3.$$
$$\Sigma_{\pi_i} = A \cdot A^T, \quad A = \begin{bmatrix} \cos\alpha_i & -\sin\alpha_i \\ \sin\alpha_i & \cos\alpha_i \end{bmatrix} \cdot \begin{bmatrix} sss_x & 0 \\ 0 & sss_y \end{bmatrix} \quad (6.1)$$

where, π_i is the geographical position of the $i-$th vehicle, sss_x, sss_y are the dimensions of the beam pattern along the x, y axes and α_i is the heading angle of the $i-$th vehicle. As described in section 2.4 that explains background detail, the beam pattern has a wide and narrow shape along lateral axis and longitudinal axis, respectively. Thus, in simulation it is set up with a range about of $200mt$ x $50mt$. At this stage, the sample data gathered throughout the simulative missions are obtained by sampling a certain number of particles $\pi_{S_{ij}}$, $j = 1, ..., m_i$ from p_{S_i} at each time frame. The number m_i of particles extracted from each Gaussian variable depends on the resolution of the region to be surveyed and on the objects resting on the seabed. Recalling that the resolution of the marine area is $100mt$ x $100mt$, the selected value to m_i is 1 or 10, depending on the detection of the side scan sonar. More specifically, m_i is set up with value 1 when the side scan sonar finds an object resting on the seabed, and with value 10 in the other cases. Finally, as defined in section 4.5, all the particles extracted from each vehicles are the data set $\mathbb{X}_{exp}$, which can be divided into two data subset $\mathbb{X}_{find}$ and $\mathbb{X}_{not}$ of findings and not findings, respectively.

Following the above assumptions, the remaining part of Section is devoted to show comparisons of search strategies by the performance measurement system previously defined. Thus two graphs are depicted. The first shows comparisons of search strategies according

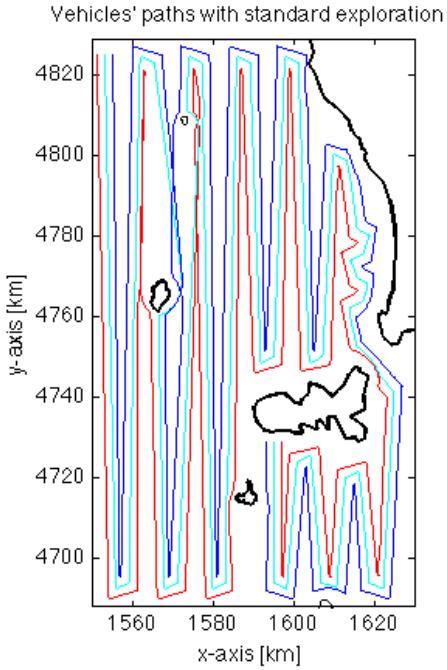


Figure 6.4: *Optimal lawn-mower pattern. The red, cyan and blue lines are the paths of the three vehicles.*

to the rate of findings evaluated against the mission time, and the second shows comparisons of search strategies according to the rate of explored area evaluated against the rate of findings. For the sake of clarity, each graph is also split into two sub graphs showing comparisons of "lawn-mower-optimized" search with cooperative distributed algorithm and multi RRT* algorithm, respectively. As a result Figures 6.5 and 6.6 show the rate of findings evaluated against the mission time for all the simulative search strategies, "lawn-mower-optimized", cooperative and multi-RRT* algorithms.

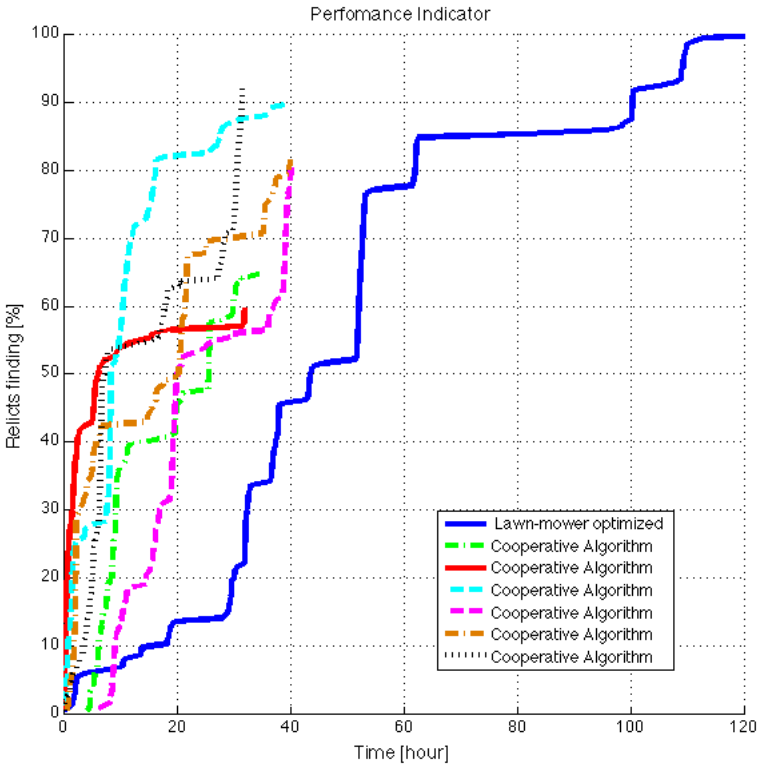


Figure 6.5: *Comparisons of the rate of found relicts plotted against the mission time by two different approaches for the systematic search strategy. The blue line refers the standard search "lawn-mower-optimized", the other line to the cooperative distributed algorithm.*

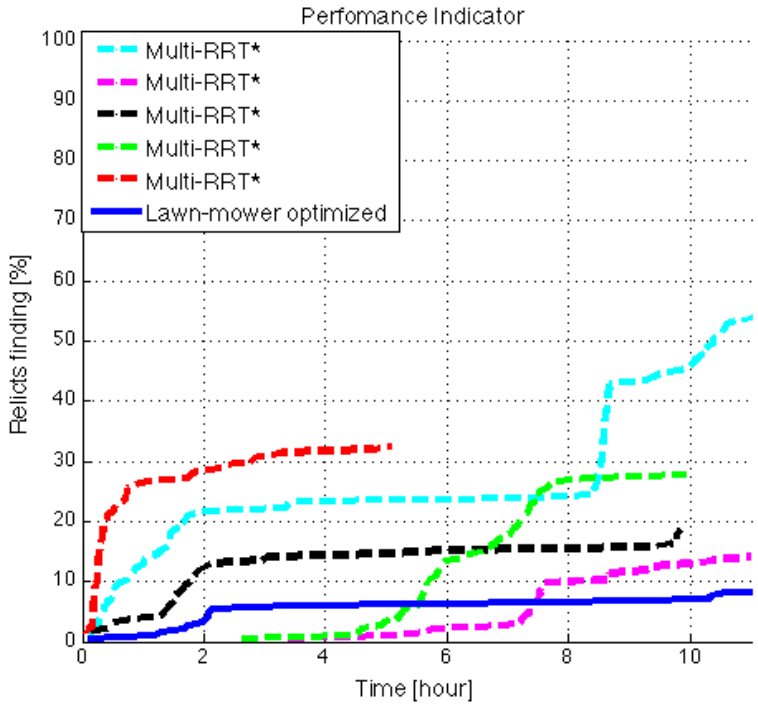


Figure 6.6: *Comparisons of the rate of found relicts plotted against the mission time by two different approaches for the systematic search strategy. The blue line refers the standard search "lawn-mower-optimized", the other lines to the Multi-RRT* algorithm.*

And, Figures 6.7 and 6.8 show the number of findings over the total number of objects to be found plotted against the fraction of explored area over the total available area for the same simulative search strategies. Note that the colours of the lines are the same employed in Figure 6.2 to show the vehicles' initial geographic positions.

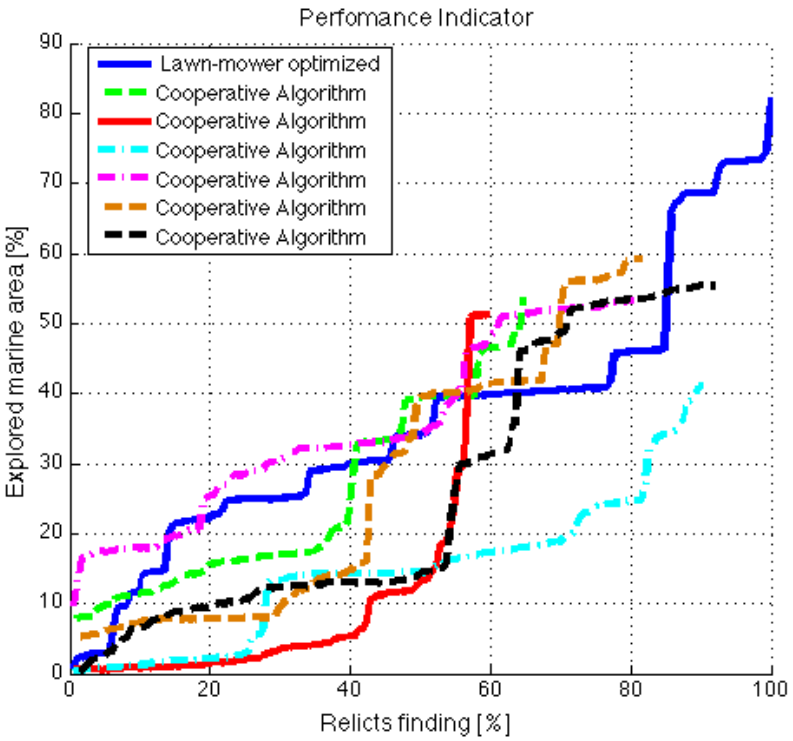


Figure 6.7: *Comparisons of the fraction of explored area over the total available area plotted against the rate of found relicts by two different approaches for the systematic search strategy. The blue line refers to the standard search "lawn-mower-optimized", the other lines to the cooperative distributed algorithm.*

Comparisons of survey cooperative simulative missions (Figures 6.5, 6.6, 6.7 and 6.8) showed that cooperative distributed algorithm and multi RRT* algorithm consistently outperformed "lawn-mower-optimized" search, while reaching a comparable number of found relicts. It is worth to note that the cooperative distributed algorithm

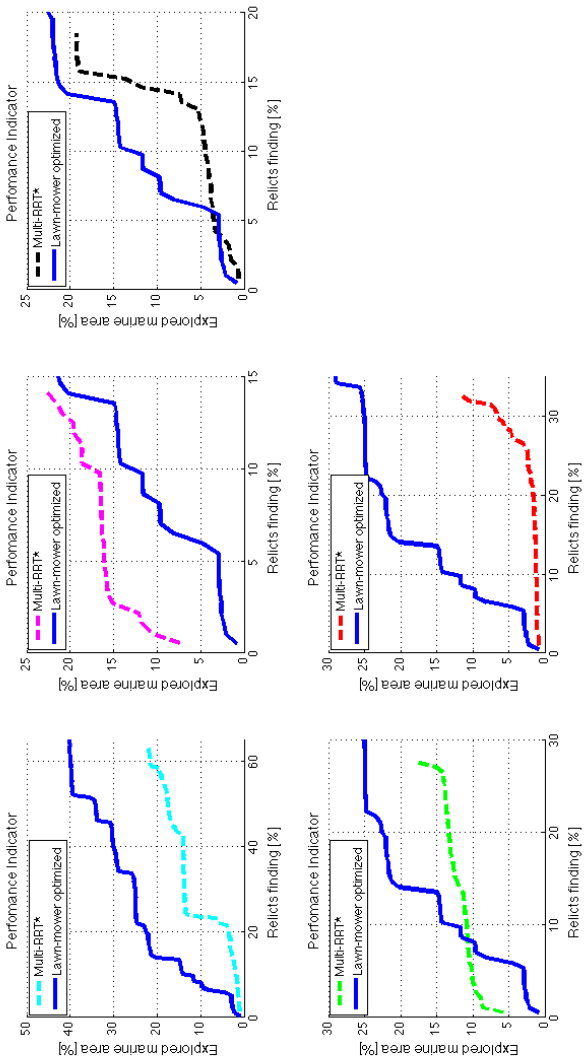


Figure 6.8: *Comparisons of the fraction of explored area over the total available area plotted against the rate of found relicts by two different approaches for the systematic search strategy. The blue solid lines refer to the standard search "lawn-mower-optimized", the dashed lines to the Multi-RRT* algorithm.*

covers around the 60 percent of the total marine area to be surveyed, against the 99 percent explored from the "lawn-mower-optimized" search, in a time about four times less. At the same time, the cooperative distributed algorithm has similar if not even better performances with respect to the multi-RRT* algorithm. Figures 6.9 and 6.10 shows the comparisons of the cooperative distributed algorithm and the multi-RRT* algorithm in detail. Notice the Multi-RRT* exploration strategy covers more marine area than the cooperative algorithm with the same ratio of found relicts and it requests more time than the cooperative algorithm to discover the same amount of relicts.

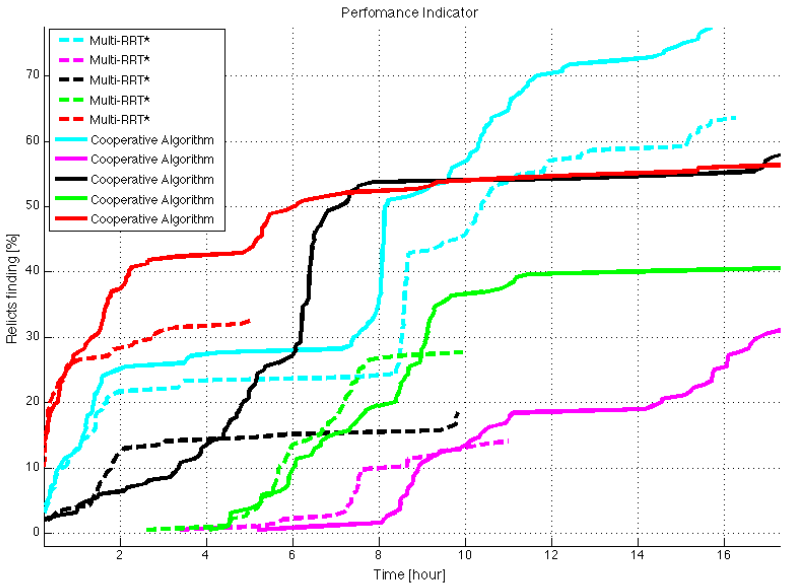


Figure 6.9: *Rate of found relicts plotted against the mission time. The solid lines refer to the cooperative distributed algorithm, the dashed lines to the multi-RRT* algorithm, per each performed simulation.*

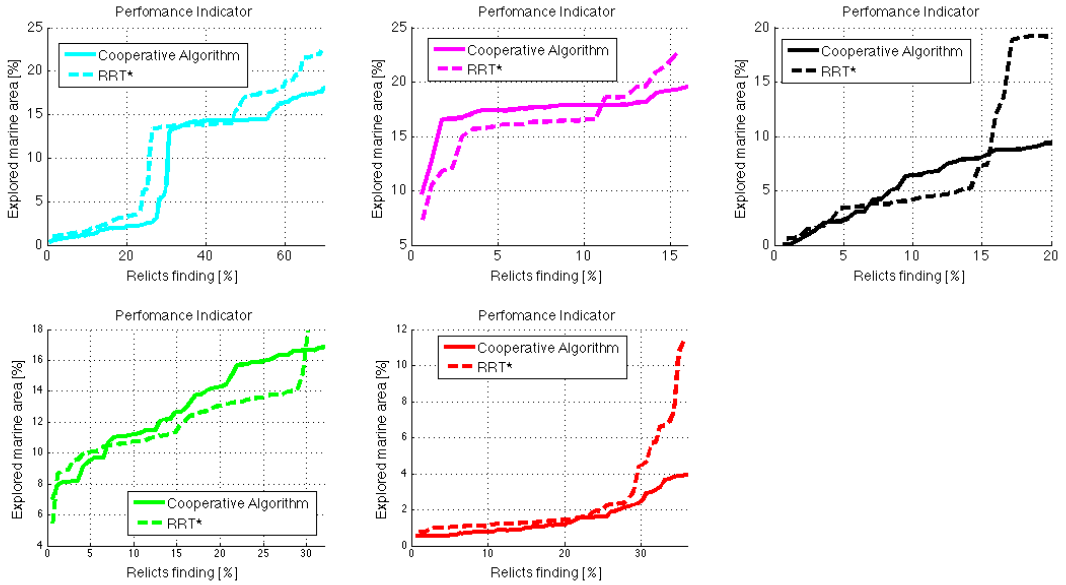


Figure 6.10: *Rate of explored area plotted against the rate of found relicts by the cooperative distributed algorithm and the multi-RRT* algorithm for each performed simulation.*

Conclusion

In this thesis we have presented two novel cooperation algorithms for a team of Autonomous Underwater Vehicles, applied for surveying areas with archaeological interest, and an innovative mathematical method to consistently represent the prior knowledge by the a priori density map over a marine area to be surveyed. In details, the first algorithm employs the theory of Equitable Power Diagrams and of potential functions to define rules that characterize the motion policies of the vehicles. The second algorithm instead is built upon the RRT* algorithm. Both algorithms have been tested with numerical simulations to compare the Information-based approaches with historical algorithms usually used for seabed surveying application. The historical approaches plans the surveys by lawn-mower or zig-zag patterns that are made prior to surveying and are not adapted while the vehicles is underway.

The comparisons showed that the proposed cooperative algorithms outperform the lawn-mower method, having minor costs in terms of exploration time and of exploration area, versus the number of discovered objects.

These cooperative algorithms have been focused on the marine

archaeology case but they are indeed general. It is actually possible to transfer the presented approach to other search applications that may include different sensor payloads, with minor modifications. On the other hand, the availability of information from the Archaeological data base of the Tuscan Superintendence has allowed to test the proposed algorithm over a real distribution of findings, and to consider uncertainties and reliability of the a priori data as defined and found in the field.

Publications

- **Journal papers**

  1. F. Di Corato, D. Meucci, A. Caiti, "Information-driven cooperative distributed motion planning for long range search over marine areas", *Submitted to Autonomous Robots.*

- **Conference papers**

  1. A.Caiti, V.Calabrò, D.Meucci, A.Munafò "Underwater Robots: past, present and future", *Proc. COMPIT 2012, 11th Int. Conf. on Computational Methods and Information Technology in the Maritime Industries*, Liege, Belgium, April 2012.
  2. A.Caiti, V.Calabrò, F.Di Corato, D.Meucci, A.Munafò "Distributed Cooperative Algorithms for Autonomous Underwater Vehicles in Marine Search Missions", *Proc. COMPIT 2013, 12th Int. Conf. on Computational Methods and Information Technology in the Maritime Industries*, Cortona, Italy, April 2013.
  3. A.Caiti, V.Calabrò, F.Di Corato, D.Meucci, A.Munafò "Cooperative distributed algorithms for AUV teams: a

minimum entropy approach", *Proc. OCEANS'13 MTS / IEEE*, Bergen, Norwegian, June 2013.

4. A. Caiti, V.Calabrò, F. Di Corato, T. Fabbri, D. Fenucci, D. Meucci, A. Munafò, "Enhancing Underwater Localization Using Acoustic Underwater Networks", *Proc. UAC 2013, 1st International Underwater Acoustic Conference*, Corfu, Greece, June 2013.

5. F. Di Corato, D. Meucci, A. Caiti, "Cooperative Long Range Search over Marine Areas: an Information-driven Approach", *IFAC workshop on Multi-Vehicle Systems*, Genova, May 18 2015.

6. S. Nardi, D. Meucci, C. Della Santina, L. Pallotino, "Dynamic coverage against asymmetric threat: a game theoretic approach ", *Oceans'15 MTS/IEEE Genova*, Genova, May 19-21 2015.

Bibliography

[1] The computational geometry algorithms library, https://www.cgal.org/.

[2] I.F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. In *Ad Hoc Networks 3*, volume 3, pages 257–279, 2005.

[3] B. Allotta, S. Bargagliotti, L. Botarelli, A. Caiti, et al. Thesaurus project: Design of new autonomous underwater vehicles for documentation and protection of underwater archaeological sites. In M. Ioannides, D. Fritsch, J. Leissner, R. Davies, F. Remondino, and R. Caffo, editors, *Progress in Cultural Heritage Preservation*. Springer, 2012.

[4] B. Allotta, F. Bartolini, A. Caiti, R. Costanzi, et al. Typhoon at commsnet 2013: experimental experience on auv navigation and localization. In *Proc. 2014 IFAC World Congress*, Cape Town, S. Africa, August 2014.

[5] A. Alvarez, B. Garau, and A. Caiti. Combining networks of drifting proling foats and gliders for adaptive sampling of the

ocean. In *Proc. IEEE Conf. Robotics and Automation*, Rome, Italy, ICRA 2007.

[6] G. Antonelli, F. Arrichiello, A. P. Aguiar, and A. Pascoal. An observability metric for underwater vehicle localization using range measurements. *Sensors*, 13(12):16191–16215, November 2013.

[7] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[8] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23:345–405, 1991.

[9] O. Baron, O. Berman, D. Krass, and Q. Wang. The equitable location problem on the plane. *European Journal of Operational Research*, 183(2):578–590, 2007.

[10] D.J. Bertsimas and G.J. van Ryzin. Stochastic and dynamic vehicle routing in the euclidean plane with multiple capacitated vehicles. *Advances in Applied Probability*, 25(4):947–978, 1993.

[11] P. Blondel and Bramley J. Murton. *Handbook of seafloor sonar imagery*. John Wiley and Sons, Chichester/Praxis Publishing, 1997.

[12] J.D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, Cambride U.K., 1998.

BIBLIOGRAPHY

[13] A. Caiti, V. Calabrò, G. Dini, A. Lo Duca, and A. Munafò. Secure cooperation of autonomous mobile sensors using an underwater acoustic network. *Sensors*, 12(2):1967–1989, 2012.

[14] A. Caiti, V. Calabrò, T. Fabbri, D. Fenucci, and A. Munafò. Underwater communication and distributed localization of auv teams. In *Proc. MTS/IEEE OCEANS 13*, Bergen, Norway, June 2013.

[15] A. Caiti, V. Calabrò, A. Munafò, G. Dini, and A. Lo Duca. Mobile underwater sensor networks for protection and security: Field experience at the uan11 experiment. *Journal of Field Robotics*, 30(2):237–253, 2013.

[16] A. Caiti, E. Crisostomi, and A. Munafò. Physical characterization of acoustic communication channel properties in underwater mobile sensor networks. In Springer, editor, *Hailes, S., Sicari, S., Roussos, G. (eds.) S-CUBE 2009. LNICST*, volume 24, pages 111–126, Heidelberg 2010.

[17] M. Cáp, P. Novák, J. Vokrínek, and M. Péchoucék. Multi-agent rrt* : Sampling-based cooperative pathfinding. In Jonker, Gini, and Shehory, editors, *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, pages 6–10, Saint Paul, Minnesota, USA, May 2013.

[18] J. Carlsson, D. Ge, A. Subramaniam, A. Wu, and Y. Ye. Solving min-max multi-depot vehicle routing problem. In Providence,

editor, *Lectures on Global Optimization*, pages 31–46. RI: Amer. Math. Soc. and Field Inst., P. Pardalos and T. Coleman, 2007.

[19] B. Chazelle. Approximation and decomposition of shapes. In J.T. Schwartz and C.K. Yap, editors, *Algorithmic and Geometric Aspects of Robotics*, pages 145–185. Lawrence Erlbaum Associates, Hillsdale, New York 1987.

[20] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[21] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[22] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 2nd edition, 2000.

[23] G. Lejeune Dirichlet. Über die reduktion der positiven quadratischen formen mit drei unbestimmten ganzen zahlen. *Journal für die Reine und Angewandte Mathematik*, 40:209–227, 1850.

[24] H. Edelsbrunner. *Algorithms in Cobinatorial Geometry*. Springer-Verlag, Berlin, 1987.

[25] P.C. Etter. *Underwater Acoustic Modeling and Simulation, Fourth Edition*. EBL-Schweitzer. CRC Press, 4th edition, 2013.

[26] S. Frolov, B. Garau, and J. Bellingham. Can we do better than the grid survey: Optimal synoptic surveys in presence of

variable uncertainty and decorrelation scales. *Journal of Geophysical Research: Oceans*, 119, 2014.

[27] Voronoi Georgy. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, 133:97–178, 1908.

[28] E. Gokay and J. Principe. Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):158–171, February 2002.

[29] K. Hlaváčková-Schindler, M. Paluš, M. Vejmelka, and J. Bhattacharya. Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1):1–46, March 2007.

[30] J.E. Hughes-Clarke, L.A. Mayer, , and D.E. Wells. Shallow-water imaging multi beam sonars: A new tool for investigating sea floor processes in the coastal zone and on the continental shelf. *Marine Geophysical Researches*, 18:607–629, 1996.

[31] H. Imai, M. Iri, and Murota. Voronoi diagram in the laguerre geometry and its applications. *SIAM Journal on Computing*, 14(1):93–105, 1985.

[32] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, June 2011.

[33] E. Kavraki, M.N. Kolountzakis, and J.C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation*, 14(1):166–171, 1998.

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

[35] J.J. Kuffner and S.M. LaValle. Rrt-connect: An efficient approach to single-quert path planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.

[36] Cara LaPointe. *Virtual long baseline (VLBL) autonomous underwater vehicle navigation using a single transponder.* PhD thesis, Massachusetts Institute of Technology, 2006.

[37] J.C. Latombe. Robot motion planning. Technical report, Kluwer Academic Publishers, 1991.

[38] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, May 2001.

[39] Steven M. LaValle. *Planning algorithms.* Cambridge University Press, 2006.

[40] NE. Leonard, DA. Paley, RE. Davis, DM. Fratantoni, F. Lekien, and F. Zhang. Coordinated control of an underwater glider fleet

in an adaptive ocean sampling field experiment in monterey bay. *Journal of Field Robotics*, 27(6):718–740, 2010.

[41] B. Liu, Z. Liu, and D. Towsley. On the capacity of hybrid wireless networks. In *IEEE INFOCOM 2003*, pages 1543–1552, San Francisco CA, April 2003.

[42] Submetrix Ltd. Submetrix 2000 series training pack. Technical report, Bath U.K., 2000.

[43] X. Lurton. *An Introduction to Underwater Acoustics : Principles and Applications*. Geophysical Sciences. Springer, 2nd edition, 2010.

[44] S. Martinez, J. Cortez, and F. Bullo. Motion coordination with distributed information. *IEEE Control System Magazine*, pages 75–88, August 2007.

[45] A. Matos and N. Cruz. Auv navigation and guidance in a moving acoustic network. In *Oceans - Europe*, volume 1, June 2005.

[46] S.D. McPhail and M. Pebody. Range-only positioninf of a deep-diving autonomous underwater vehicle from a surface ship. *IEEE Journal of Oceanic Engineering*, 34(4):669–677, 2009.

[47] A. Munafò, E. Simetti, A. Turetta, A. Caiti, and G. Casalino. Autonomous underwater vehicle teams for adaptive ocean sampling: a data-driven approach. *Ocean Dynamics*, 61(11):1981–1994, 2011.

[48] A. Okabe, B. Boots, K. Sugihara, and S.N. Chiu. *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams*. John Wiley, 2nd edition, 2000.

[49] G. Parlangeli and G. Indiveri. Single range observability for cooperative underactuated underwater vehicles. In *IFAC World Congress*, volume 19, pages 5127–5138, Cape Town, S. Africa, 2014.

[50] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[51] L. Paull, S. Saeedi, and H. Li. Path planning for autonomous underwater vehicles. In Mae Seto, editor, *Autonomy for Marine Robots*, pages 177–224. Springer, 2012.

[52] Liam Paull, Sajad Saeedi, Mae Seto, and Howard Li. Auv navigation and localization - a review. *IEEE Journal of Oceanic Engineering*, 2013.

[53] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo. Equitable partitioning policies for robotic networks. In *IEEE International Conference on Robotics and Automation*, Kobe Japan, May 2009.

[54] M. Pavone, A. Arsie, E. Frazzoli, and F. Bullo. Distributed algorithms for environment partitioning in mobile robotic networks. *IEEE Transactions on Automatic Control*, 56(8), August 2011.

BIBLIOGRAPHY

[55] David R.Anderson. Some background on why people in the empirical sciences may want to better understand the information-theoretic methods, 2003.

[56] A. Renyi. On measures of entropy and information. *Proc. Fourth Berkeley Symp. Math, Statistics and Probability*, pages 547–561, 1960.

[57] R.M. Rogers. *Applied Mathematics in Integrated Navigation Systems*. American Institute of Aeronautics and Astronautics, Reston, VA, USA, 3rd edition, 2000.

[58] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[59] B.W. Silverman. Density estimation for statistics and data analysis. *Chapman and Hall*, 1986.

[60] HG. Tanner, A. Jabadaie, and GJ. Pappas. Stable flocking of mobile agents, part i: fixed topology. In *Proc. 42nd IEEE Conference on Decisions and Control*, 2003.

[61] L. Fejes Tóth. Illumination of convex discs. *Acta Mathematica Academiae Scientiarum Hungaricae*, 29(3-4):355–360, 1977.

[62] J. Vaganay, J.J. Leonard, J.A. Curcio, and J.S. Willcox. Experimental validation of the moving long base-line navigation concept. In *IEEE/OES Autonomous Underwater Vehicles*, pages 59–65. IEEE, 17-18 June 2004.

[63] Keith Vickery. A practical overview of current systems. *Acoustic Positioning Systems*, 1998.

[64] M. P. Wand and M. C. Jones. Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88:520–528, 1993.

[65] Sarah E. Webster, Ryan M. Eustice, Hanumant Singh, and Louis L. Whitcomb. Advances in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. *The International Journal of Robotics Research*, 31(8):935–950, 2012.

[66] S.E. Webster, R.M. Eustice, H. Singh, and L.L. Whitcomb. Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. In *Intelligent Robots and Systems*, pages 2053–2060. IEEE/RSJ International Conference on, October 2009.

BIBLIOGRAPHY