SEARCH & EXPLORATION: EFFICIENT PLANAR SEARCH FOR AUTOMATED ROBOTIC DISCOVERY

Scott Burlington

Department of Computer Science
McGill University, Montréal

15 December 1999

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Science

0-612-64328-X

ABSTRACT

An agent is placed in an unknown environment and charged with the task of locating a lost object. What can the agent use as an efficient technique to find the object?

We propose a new algorithm for planar search. The algorithm stems from theoretical work on search games, in particular provably optimal search techniques on restricted domains. This thesis addresses the problem of efficiency in robotic search: having a mobile robot find a target object in an unknown environment with obstacles in an efficient manner. As a side-effect, the robot explores the environment.

Based on previous results, a formal description of the problem is presented along with an algorithm to solve it. This algorithm has good worst-case performance, in terms of its competitive ratio. We show experimental data validating the feasibility of our approach and typical results. Quantitative results are demonstrated showing the advantage of modified spiral search versus traditional approaches.

RÉSUMÉ

Un agent est placé dans un environnement inconnu et sa tâche est de localiser un objet spécifique. La question est de savoir quelle est la technique la plus efficace pour la recherche de cet objet.

Nous proposons un nouvel algorithme de recherche en spirale dans le plan cartésien. Inspiré par la théorie des jeux, notre algorithme est basé sur les techniques de recherche optimale dans des domaines restraints. Dans cette thèse, nous addressons le problème de faire parvenir un robot mobile à trouver efficacement un objet spécifique dans un environnement complexe et inconnu. Comme effet secondaire, le robot explore cet environnement.

Basée sur des études antérieures, une description formelle du problème, ainsi qu'un algorithme pour le résoudre seront présentés. Cet algorithme est compétitif, dans le sens où le rapport entre sa performance et celle d'un algorithme idéal est constant. Des résultats experimentaux démontrent la validité de notre approche. De plus, nous presenterons les avantages de l'algorithme de recherche en spirale que nous proposons comparé àux autres méthodes de recherche traditionelles.

ACKNOWLEDGEMENTS

My many thanks for the assistance offered to me from my supervisior, Professor Gregory Dudek, who is responsible for both the environment and great motivation to explore the many twists and turns this research has taken and for the patient guidance to keep me on some sort of track.

My parents are the ones who are responsible for my being here in many ways. I thank them immensely and send my gratitude. I hope that they will find this thesis is worthy of their efforts.

I would like to thank Robert Sim for his lightning fast technical support with *RoboDæmon* and for just helping, always. Eric Bourque also for his careful reading and edits, the time that he spent is time that I saved.

I would also like to thank all my mates in the Mobile Robotics Lab for helping create such a great environment. For all the time I spent at this thesis, not a second was wasted in the lab.

Je voudrais dire merci à Sylvain, François et Maya pour leur aide dans la traduction du résumé. Je ne pourrais pas l'avoir fait sans vous.

DEDICATION

This dissertation is dedicated to Zachary. Who doesn't know where Montréal is, but who knows that it isn't that far away.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Search is one of the fundamental tasks of living beings. Animals spend large parts of their life foraging for food, hunting, roaming for more hospitable environments, trying to find members of their species. Whatever the specific task, these are all types of searching. All animals need to be successful searchers to ensure that survival tasks are accomplished efficiently. From insects to humans, animals everywhere spend their waking hours seeking food, shelter, habitat and mates. Vast fields of human knowledge have been acquired as a result of this task of searching our environment, and no longer is it even our own environment that we expend effort to search. Undersea and space research are frontiers that human-kind has embarked upon in order to help satisfy our innate need to explore [**17, 28**]. Search leads us into new environments, but how can search be made efficient when exploring an unknown environment? This is the question to be addressed in this work.

Herein an analysis of some exploration and search techniques useful in the field of mobile robotics is presented. A new technique for mobile robot search and exploration will be introduced. Motivated from mathematical results in restricted theoretical settings, our approach for mobile robot search in unknown environments will be shown to be more efficient than traditional methods. The search environment will be a planar network of halls and rooms. It is the robot's task to search efficiently throughout the network and to thereby explore the *a priori* unknown environment. Traditional methods are inefficient in terms of the time taken to find an object or particular location that can only be determined by a local sensor in such a setting. Our particular brand of search is a form of iteratively deepening depth-first search based on logarithmic spirals to determine the points of iteration. This method has better average-case performance with regards to the task of searching within an environment.

Let us first establish that exploration, in a general sense, is the process of expanding one's knowledge about the environment. It is important to realize that the term environment is very flexible in its interpretation, ranging from underground ant passageways to far more abstract environments, such as the recently popularized *Noosphere*[1], the space of all human thought and ideas. By the process of exploration, living beings sample their environment for indications of danger, availability of food, materials for tools as well as any large variety of other potential targets. If exploration is seen as expanding the knowledge the environment then the next logical question is how to proceed with this exploration to be most successful.

Search is at the core of exploration. Search can be seen as the method employed to increase knowledge of one's surroundings. When exploring a new environment, what strategy should be employed to ensure that the exploration is being accomplished in an efficient manner? Thus the method of search then becomes a critical element of exploration. How do we describe what a "method of search" is? How do we quantize such a method for analysis? Is it even possible to define a simple strategy for search that will be effective in radically different scenarios?

Ideally, we might be able to construct sets of scenarios for which a particular search strategy is most appropriate or useful. How then can we attempt to define different types of environments when these environments may be be such different things as "the space of all thoughts" versus a road map for a city, for instance. Certainly it is clear that any particular strategy of search will be inefficient within all the different possible environments,

[1]Eric S. Raymond recently wrote an article *Homesteading the Noosphere* [58] in which he discusses the space in which ideas are developed by human thought, both academically and industrially. Here he speaks of the "Noosphere" to be the space of all possible thoughts and ideas, within which human knowledge is constantly expanding as we add to our kernel of ideas.

but perhaps there is some set of strategies that can be applied and tested in a new environment to see how it fares against other known approaches, and hence using hybrid search techniques composed of fundamental search techniques that may be used in a wide range of applications.

If successful search is so important a factor to an individual how is it possible to do better? Working in groups can be a way to speed the search process, so that the team work pays dividends greater than might be had otherwise. The parallelization of tasks, the distribution of workload among several individuals is a relatively difficult task requiring communication and agreement between co-operating agents. But can we define a scheme to improve on what a single agent can achieve? Working in teams, cooperation for mutual benefit, would be a indicator of intelligence in any sort of social system.

The solution presented for efficient search is termed "Spiral Planar Search" for a mobile agent. The method implemented is based on extensive theoretical grounding and addresses the issues of scalability of theory to a functional robotics level. Both of which are active research problems. Spiral planar search will be shown to be more efficient than the competitive methods of depth-first or breadth-first search in terms of searching an unknown environment, on average.

1. Background

There are two meta-problems addressed in this thesis. The first is a search problem developed from theory in a theoretical, constrained setting. This first problem may have been motivated by a need in robotics, but is properly restricted to mathematical theory. The second meta-problem has to do with the extension of a theoretical idea to a functional mobile robotic implementation. There has been a long standing call in mobile robotics research for the integration [45] of fundamental problems, or rather the solutions as they stand to those problems, in order to produce more highly functional robotic systems.

Shown will be the development of a theoretically grounded problem to a fully functional simulation designed to act as test bed for the solutions to the initial problem. This brings together theory and application studies in an integrated robust system.

There has been much work completed in the field on the union of these meta-problem areas. Much good theory for the underlying structure that is necessary to develop structured solutions comes to us from a variety of sources. Classic sources for motivating planar search and rendezvous games come from works like Schelling **[65]** and Gal **[36]**. Generalized graph searching algorithms that are related to the problem at hand but which don't have any robotic motivation necessarily are covered in **[42]** on wandering RAMS with bounded memory searching binary trees and **[15]** where the search is for integers in unbounded sets with no distance traverse cost.

Work on developing the underlying theory for search games in general **[39, 49]** and the means with which to analyze the performance of search algorithms **[1, 2, 4, 5]** has been foundational work.

The problem of online search has been a popular theme is robotics research. Typically the problem is set in restricted environments in order to aid analysis or develop very particular algorithms. Solutions to problems in searching restricted polygons often are very specific to the constraints on the polygon. Interesting results in searching simple polygons **[44]** and searching the kernel of a star-shaped polygon **[52]** both have excellent approaches to the solution of online search algorithms. Another closely related polygonal search problem is with the set of infamous polygons known as *generalized streets* initially set forth by Icking **[22]** with follow-up work from the research field **[43, 38, 44, 51]**. Such restricted application domains need extension to be useful in more general settings, however the problems do apply to an interesting set of polygons.

Brumitt **[20]** has addressed the area of optimal paths for multi-robotic systems, but global infinite communication has been assumed to achieve this goal, clearly a specialty requirement not generally available in desired robotic settings. A huge variety of work on the area of exploration, whether in an *a priori* unexplored environment **[55, 27]** explicitly or more generally with exploration and optimal map building, has been done **[56, 52, 53, 67, 59, 60, 55, 27, 50, 18, 19, 53]** in the research field. With such a large variety of work, it becomes difficult to categorize all the contributions brought to us. Much of this work can benefit from an improved method to discover unknown space. Whether actively searching for a known target in unknown surroundings or exploring large environments in a reliable manner efficient methods to move within the environment are needed to accommodate the relatively slow physical motion exhibited by most mobile robots.

The studies by Yamauchi *et al.* [**68, 70, 69**] have been very motivating with their approach to functional robotic systems. Zelinsky's work [**71**] too has been a well considered approach to the study of robotic exploration. Aside from the motivation that any of these previous works may have provided, they also showcase an entire research field of potential application for a more efficient means of planar search. Any of the above studies that do not explicitly deal with exploration algorithms *per se* could benefit from an improved search and exploration algorithm.

Another approach that can be taken to speed the discovery of unknown surroundings or to shorten the time taken to find a target consists of utilizing multiple agents concurrently. Multi agent map merging [**41**] has always been an interesting topic. The coordination [**54, 34**] of a multitude of agents for any sort of task continues to be an ongoing research area. Inter agent communication studies [**12, 13, 62**] also help us to understand the capabilities and restrictions that may apply to multi-agent systems.

The topic of navigational methods has been a very important foundation for this thesis as well. Navigational approaches for exploration tasked mobile robots by Blum [**16**], Foux [**35**], Rao [**57**] and especially by Dudek *et al.* [**29, 30, 31, 33**] have been core to the development of the robotic simulation and to the experimental test-bed. The decision to use a topological representation for this work has was made based on these references and on research pertaining to spatial models and representation by Kuipers [**46, 47, 48**].

Of Particular note, due to the intersection of academic principles, we note work by Alpern [**1, 2, 3, 4**] and Anderson [**8, 7**] for their work in game theory and rendezvous or similar search techniques. Along with the very central work of Nicholas Roy [**62**] as a more functional robotics based piece of rendezvous work.

In the area of planar search we have followed closely work by Baeza-Yates [**10**] with Culberson and Rawlins [**11**] for a basis of search algorithms in more precise environments. The collection of these results served in part as motivation to the beginnings of this work.

The importance of the work and direction provided by Dudek *et al.* [**29, 30, 31, 33, 59, 61, 60, 66, 62**] would be hard to overstate. These contributions have had a strong influence on this thesis in many disperse subjects such as path planning, navigation and localization in mobile robotics research.

2. Integrating Interdisciplinary Research

This thesis will reiterate the nature of two separate aspects of modern robotics and propose a means of integration between them. The primary aspect is a novel robotics application, specifically a search and exploration method that will be extended from theoretic results in planar geometry. Qualitative experiments will be performed to show the efficacy of application. The second aspect is consideration of techniques to use multiple agents to efficiently streamline a complex search problem.

Mobile robotics has traditionally been approached using "bottom up" reasoning. The mobile robot is the sum of its components and behaviours. However, the problems that need to be dealt with for successful robotics are themselves so complex that they deserve study in their own right. There have been alternative attempts at attacking the robotics problem in a more holistic manner such as Brooks' [**18, 19**] subsumption architecture where different layers in the abstraction of the mobile robot are more insulated from one another. The desire is to let the world be its own representation, and that intelligent behaviour at lower levels does not rely on an accurate complete internal representation of the world.

Ideally topics such as localization, exploration, navigation, path-planning, map-building, computer vision, feature recognition and natural language comprehension/synthesis would be blended together seamlessly to form a functional and very impressive robotic companion. Roboticists have delved into these problems and developed sciences appropriate to the study of each discipline. Mobile robotics remains a puzzle of unsolved problems. This call for integration has been formalized by Kortenkamp and Shultz [**45**] in the general robotics community.

3. Search Properties

There are some desirable properties that a reasonable search algorithm should display. Certainly these desirable properties are not minimal requirements, for instance a *Random Walk* could be admissible as a search algorithm. We will use the following properties in for the purposes of qualitative evaluation.

(i) Termination — Will the entire region be searched? Is exhaustive search guaranteed.
(ii) Principle of Locality — Targets located near the origin of search should be discovered with precedence over distant targets.
(iii) Efficiency — the search algorithm should exhibit reasonable worst-case or average-case behaviour. Canonical failure modes should not arise in common cases.
(iv) Exploration — the search should result in complete exploration of the environment. Moreover, this exploration should be done in an efficient manner.

These properties will be referenced with relation to the various search techniques throughout the text.

4. The Approach

The underlying theory of optimal planar search in theoretically restricted settings is described initially to give basis to decisions made later in the more complex environments required by the mobile robots. This theory is developed to establish some formal justification for the novel approach that we develop for efficient robotic search and exploration as extension to and experimental verification of the core work by Baeza-Yates, Culberson and Rawlins [**11**] and Baeza-Yates [**10**].

In addition rendezvous search theory and techniques will be investigated. We see how some of these techniques have been used in modern work to motivate multi-agent coordination for mobile robots. A branch of statistics known as game theory, more precisely the theory revolving around cooperative, two-player, zero-sum games will provide a basis for the investigations of multi-agent cooperation. The foundational work here from Alpern [**1, 2, 4**] provides methodology to the approach of the problem and a means by which to analyze it. Roy [**62**] provides a wonderful example of a similar extension of theoretical results into a robust technical verification that corresponds to and gives appropriate testing to those ideas in a functional setting.

Lastly we implement an experimental framework exhibiting the functionality of a robotic system performing several different types of search in similar environments. Our framework will serve to assess the performance issues involved with the differing algorithms, and to show results from a complete and operational system.

5. Contribution

Spiral search techniques for planar search and discovery provide a more efficient method for robotics search and discovery. While the current implementation is restricted to the two dimensional case for purposes of analysis and evaluation, there is nothing in the theory that prevents application in three dimensions. Moreover, spiral search techniques have a wide variety of application domains and do not suffer from many of the severe restrictions that similar work on the theory of searching requires. This general approach to online search makes very few assumptions about the environment and is thereby largely unrestricted in terms of the potential application domains.

6. Organization of the Thesis

This thesis is organized as follows. In chapter 2 we will present a reasonably comprehensive overview of the work which is of central importance to this work. In particular the most appropriate results and definitions from the work of [**1, 2, 11, 62**] will be brought out as well as the theoretic foundation for this thesis. The core ideas to the development and design of our experimental system will be introduced here. All of these ideas will be expounded upon and further detail will be deferred to the appendix or the original works themselves.

Section 2.1 covers the issues involved with rendezvous search. It looks at treatments of the problem as both a two player statistical game problem [**1, 2, 3, 4, 7**] and also as a mobile robotic experimental system [**62**]. These queries will be most useful later in chapter 3 which focuses most directly on rendezvous search strategies in the sense that we wish to make use of them.

Section 2.2 is an enumeration of the difficulty of search in the plane given different sets of knowledge regarding the target. This section is a partially complete summary of Baeza-Yates' work on search **[11, 10]**. This groundwork will be extended in chapter 4 where an extension to our mobile robotic setting will be outlined.

The description and analysis of a simple rendezvous search problem is detailed in chapter 3. Motivated by the optimality results that have been found for spiral search techniques and by the analysis of the rendezvous search problem we investigate a hybrid problem. A competitive ratio analysis for search in the half-plane is presented as a *symmetric rendezvous value* for the problem.

Chapter 4 explains how the theory of search has been adapted to the current problem and provides context for the entire thesis. This adaptation of theoretical ideas into implementable strategies has been carefully considered so as not to make any assumptions that might invalidate the interpretation of the theory into a real world system.

Chapter 5 will document the application environments and the experimental framework that is being developed in this thesis. The framework documented in section 5.3 will be properly analyzed and exhibited in the following chapter 6. Details in chapter 5 will reveal the experimental procedure used in this thesis. Brief discussion will be given to some of the assumptions that had to be made and some defense of the experimental dependencies will be given.

The experiments and the resultant data are discussed in chapter 6. Any interesting or particular details related to the individual experiments are given and the results from the experiments are collected and analyzed. We give good experimental verification to the initial claims that spiral search is an efficient search technique worth adopting.

Chapter 7 concludes this thesis. Summary is given to the stated problems, the means by which they were addressed, the techniques involved in testing those hypothesis and a look forward is taken. The potential for the application of this technique to mobile robotics problems is reiterated and future work topics stemming from this current will be suggested. Directions for related and continuing studies in this area will be considered.

CHAPTER 2

Previous Research

It is worthwhile here to present a brief summary of other work that is closely enough related to this thesis topic to be of value. Many of the ideas that will be called upon later as "foundation" ideas are set out in the following group of papers. So a reasonable presentation of the papers and the appropriate ideas therein shall be given here.

Firstly we shall consider the area of multi-agent coordination. We shall use the terminology of "Rendezvous Methods" to refer to the concept of coordinating multiple agent behaviour according to any sort of strategy. Any attempt to describe a multi-agent system requires consideration of both how the entire group is managed as well as what each agents behavior is in the group. So the mechanism of how control is distributed throughout the group, the amount of communication possessed by the agents, the tightness of their awareness of each other and the way that work is distributed among the agents will all be classified as the rendezvous technique. Alpern's [2] work in the theory of this area is pivotal to the adoption of this topic within the thesis. Work by Anderson and Fakete [6] as well as Roy [62] also will be examined for some of the ideas that they present.

Concern will be geared more toward the group management scheme in section 1, rather than on the particular behaviour of each agent. That topic we will regarded in the next section on Exploration Strategies. This is reasonable because what the particular behaviour of an agent should be fairly transparent within the entire group scheme. Certainly the specific method of search used is necessary to identify a complete "strategy" but we will defer the details of the exploration strategy to section 2 on Exploration.

1. Rendezvous Methods

The topic of rendezvous methods is to be regarded as a thread of multi-agent control theory, theory which has been worked on rather extensively in a variety of fields (see Balch [**12, 13**] and Arkin [**9**] for a subset related to multi-agent robotics). Explicitly, however, the term "rendezvous methods" is intended to refer more specifically to work relating to the subset of multi-agent coordination together with some sort of fail-safe behaviour in the event of communication failure, sensor failure, failure to encounter other agents or any other sort of unpredictable short comings that may occur in the real world.

Good results come to us from Alpern [**2, 4**], Anderson [**6**], and Roy [**62**] in this new field. For the remainder of this section we will look at what kind of results are supplied to us from some of the literature. Specifically we will consider the "rendezvous search problem" proposed by Alpern and we will consider the problem of "multi-agent exploration and rendezvous" as posed by Roy and the solutions that they offer to these problems.

1.1. The Rendezvous Search Problem. Alpern's work [**2, 4**] serves as a good reference to the ideas of what it is that the rendezvous search problem incorporates as well as allowing us to introduce the concepts that are involved with coordinated search in a formal way. Also, the paper holds very promising insights to this field which will help exhibit the merits of further study.

Alpern defines the rendezvous search problem as follows. Two agents are placed randomly in a known environment with the objective of finding one another. Each agent moves at unit speed and knows nothing about the location of the other agent until it is actually discovered. The problem is to minimize the expected time to discovery. The solution is presented as a general solution cast in a compact metric space together with a group of isometries [see B.3 for a definition] which reflect the amount of information that the agents share about the environment. It is an important realization that the rendezvous time (or *rendezvous value*, $\mathcal{R}$) that is finally assigned as the solution is a function of the topological characteristics of the search region $\mathbb{X}$ for the given search strategies.

Multi-agent search strategies come in two types. Symmetric search strategies are those such that each agent follows the same set of rules while searching. Asymmetric search strategies are those such that each agent has a potentially different search algorithm that it follows. An example might illustrate best. Imagine two agents on the boundary of the same circle looking for each other. They have no vision and must collide to detect each other. A symmetric search strategy might be "walk clockwise". Both agents following such a strategy will never discover each other if they walk the same speed unless they start at the same spot. An asymmetric example might be for agent 1 to "walk clockwise" and for agent 2 to "walk anti-clockwise". Here each agent employs a different strategy than the other. Discovery is guaranteed within a walk of at most half the circumference if the agents are the same speed.

Alpern uses the symmetric and asymmetric alternatives to motivate two different measures for search strategy quantization. The symmetric rendezvous value, $\mathcal{R}^s$, and the asymmetric rendezvous value, $\mathcal{R}^a$. These values are the *least expected meeting times* for the search strategy over the defined environment. The asymmetric case yields lower values than the symmetric case, i.e. $\mathcal{R}^a \leq \mathcal{R}^s$ typically. Alpern concentrates primarily on symmetric strategies, as will we in the bulk of our work, since we will not assume that we can differentiate between agents in the environment. The kind of symmetric strategies that we are going to see are known as *mixed strategies*, whereby the actual strategy being used is a combination of strategies and a method for how to blend them. Mixed strategies allow for symmetric strategies to break the kind of deadlock that we have already observed in our example of the searchers on the circle when they each simply walk clockwise. So the algorithms are of the flavour "perform x for some time t_1, and then switch to y for time t_2 ... " so that there is a mixed nature to the behavior of the agents. By no means does "symmetric search" imply that the agents behave identically, simply that they follow the same algorithm, deterministic or not.

Alpern acknowledges that the use of "focal points" is of practical utility, although his intention is to show that focal points in the environment are not a prerequisite to successful[1] rendezvous. Indeed, in practice these focal or "distinctive" points will be of great utility and in Roy's **[62]** work we see how a practical application can take great advantage of the non-homogeneity of the environment while still employing the multistage strategies that differentiates between Alpern's work and Schelling's **[65]**.

To define the problem formally Alpern first tackles the rendezvous problem in its most general setting "rendezvous on a compact metric space" and then goes into studying the problem in more restricted regions. We will present his argument for the general setting and then show the results most related to the current work. For more complete discussion the reader is of course referred to the actual paper **[2]**.

The general problem consists of a compact metric space $(\mathbb{X}, \rho)$ with a given detection radius of δ and a group, G, of isometries of $\mathbb{X}$ [see B.3 for a definition]. The distance function [B.6], ρ, is what turns the compact space into a metric space. It is assumed that the players know the environment, and can localize themselves within it, but cannot detect the other player until they occupy positions closer than δ to each other. The group G represents the player's uncertainty about the region that their companion occupies.

Consider the set of paths, P, that may be taken within the region as $P = \{p : \mathbb{R}^+ \to \mathbb{X}, \rho(p(t_1), p(t_2)) \leq |t_1 - t_2|\}$. The subset of paths originating at a point, $x \in \mathbb{R}^+$, is denoted P_x. The *meeting time* $T : P \times P \to \mathbb{R}^+$ of two paths p, q is defined by the time $T(p, q)$,

$$T(p, q) = \min\{t : \rho(p(t), q(t)) \leq \delta\}, \tag{2.1}$$

[1]success here can be taken to imply that we are able to successfully rendezvous with greater likelihood if we are able to choose our search algorithm rather than simply wandering about

where δ is the detection radius and ρ is our distance function. The group G induces an equivalence relation on $\mathbb{X}$ and P according to:

$$x \sim y \iff g(x) = y \quad , \exists g \in G$$
$$p \sim q \iff g(p(t)) \equiv q(t) \quad , \exists g \in G$$

The associated sets of equivalence classes are referred to as $\tilde{\mathbb{X}}$ and $\tilde{P}$, with the cosets of a point represented by the $[\,]$ operator.

Then a *search strategy* is a map $s : \tilde{\mathbb{X}} \to \tilde{P}$ such that there is some path $p \in s(x)$ with $p(0) = x$. The set of all such search strategies will be referred to as S.

Now that we have described formally what it means to be a path, we can describe what it is we are looking for. What is the expected meeting time for pairs of equivalence classes of paths? We calculate the expected meeting time of two paths by integrating over all equivalent sets of paths in the environment. Let ν be denote the Haar measure on G [see B.7 for definition] and define[2] $\tilde{T}([p],[q])$ to be the expected meeting time of the two paths,

$$\tilde{T}([p],[q]) = \int_G T(gp, q)\, d\nu(g). \tag{2.2}$$

With this definition of distance between paths, we can progress to consider the expected time of rendezvous for search strategies. Given two search strategies, $s_1, s_2 \in S$, we want to generate the *normal form*, $\hat{T}$, for the search problem. In the case of the symmetric search strategy this simplifies by choosing $s_1 = s_2$. The normal form for the search game is thence given by $\hat{T} : S \times S \to \mathbb{R}$ working on pairs of search strategies and returning an expected time. We assume the players are placed independently according to the same measure, μ, to begin. We hereby define the normal form for two search strategies to be

$$\hat{T}(s_1, s_2) = \int_{x\in\mathbf{X}} \int_{y\in\mathbf{X}} \tilde{T}(s_1(x), s_2(y))\, d\mu(x) d\mu(y). \tag{2.3}$$

The rendezvous search value can now be defined in terms of the normal forms for sets of search strategies. We have seen how using symmetric strategies can lead to infinite expected meeting times. Our simple example of the "walk clockwise" strategy amounted

[2] G only acts on one path since for $g, h \in G, T(gp, hq) = T(h^{-1}gp, q)$

to an infinite expected rendezvous time akin to a dog chasing its tail. In order to avoid this pitfall we allow mixed strategies to be our symmetric strategies, which allows for independent randomization between agents. For more detail on the formal implications of this randomization see Alpern [2] but we will suffice it to say here that this does not impose any constraints that will be manifested in any environment that we will be concerning ourselves with in application.

The asymmetric rendezvous problem is to find the pair of strategies which minimize the expected meeting time of the agents in a given environment. Remember that the "environment" here is really the actual environment together with the set of isometries that tell us how much we know about the environment and our minimal encounter distance, δ. We take all the search strategies for the environment and assign a minimal one to be the value of the search problem

$$\mathcal{R}^a = \mathcal{R}^a(\mathbb{X}, G, \delta) = \min_{r,s \in S} \hat{T}(r, s). \tag{2.4}$$

The symmetric rendezvous problem is similar, we are also looking for the strategy that minimizes the time to rendezvous. However we must be a little more cautious in the definition to take care with the independent randomization that is occurring in the mixed symmetric strategy we are using. Because we are employing such mixed strategies, we necessarily have to extend our set of all strategies to the set of all mixed strategies. This set we will denote S^*. Having chosen our path space a little more cautiously, we are free to define the symmetric search value by

$$\mathcal{R}^s = \mathcal{R}^s(\mathbb{X}, G, \delta) = \min_{s^* \in S^*} \int_S \int_S \hat{T}(s_1, s_2)\, ds^*(s1)\, ds^*(s2). \tag{2.5}$$

The kinds of rendezvous strategies that are analyzed in [2] are summarized here to give some sense of setting. Rendezvous on a circle, C, is considered. We have two players on the circumference of a circle who wish to rendezvous. There are 4 different strategies analyzed as follows and shown in table 1.

Know	Init/Orient	Orient	Nothing
$\mathcal{R}^s$	$\frac{2}{3}$ GoToZ	$\frac{3}{2}$ CoHaTu	$\frac{3}{2}$ CoHaTu
$\mathcal{R}^a$	$\frac{1}{2}$ OpDir	$\frac{3}{2}$ OpDir	1 GoStay

TABLE 2.1. Summary of rendezvous values for 2 player rendezvous search games on the boundary of a circle.

- GOTOZ Take the shortest path to the point Z on the circle. Complete knowledge of the initial position and the whereabouts of the point Z are required. This is a symmetric algorithm, with an expected rendezvous time of $\mathcal{R}^s(GoToZ) = 2/3, \quad \forall Z$.
- COIN HALF TOUR[CoHaTu] Oscillate from the initial point to the antipodal point along either route (equiprobably each time). The expected rendezvous time here is, $\mathcal{R}^s(CoHaTu) = 3/2$. Incidentally, this is symmetric, but the result holds even if only one of the two players adopts this strategy.
- OPPOSITE DIRECTIONS[OpDir] One player proceeds clockwise, the other counter clockwise. This requires knowledge of "up", but not of initial position. Here we have $\mathcal{R}^a(OpDir) = 1/2$.
- GOSTAY One player sits still and the other randomly chooses a direction to begin a traversal. This requires no knowledge of initial position or orientation. The expected rendezvous value of this game is $\mathcal{R}^a(OpDir) = 1$.

These search values for the circle certainly help to illustrate how it is possible to speak of an expected rendezvous time for a given algorithm. Later in [**2**] another estimate is given for rendezvous search on a line. In this case we have the agents on a common line a distance, D, apart and they wish to find each other. The proposed search algorithm, $1F2B(x)$, is one step forward two steps back (where step size is x). In particular, when the distance is known then Alpern conjectures that $1F2B(\frac{D}{2})$ is optimal with a search value, $\mathcal{R}^s = 5/2$. This conjecture is more than reasonable and is precisely the kind of motivation that we need to start thinking about more generalized search. Consider that when we do not know the initial distance this search game becomes much more subtle to play and analyze. In section 2 we will see that the asymmetric case of one player doing $1F2B$ and the other sitting still will yield $\mathcal{R}^a = 9$.

1.2. Multi-Agent Exploration and Rendezvous. Roy's [62] work exhibits a novel implementation for the work in rendezvous search. From an understanding of the theory that we have investigated in the previous section Roy shows a real world multi-robotic system that uses these techniques effectively to improve upon the capabilities of a single agent.

Also of importance, explanation of this work will help bring to light some of the issues commonly involved with mobile robotics. Limited sensors, difficulty with localization, path planning and navigation failures and unreliable recognition capabilities are all commonplace issues that have to be considered in the study of mobile robotics. A good robotic system will exhibit graceful degradation in failure modes and also show recovery techniques that allow continued operation of the system. We consider the contribution of Roy's [62] work for these reasons.

Roy considers the problem of rendezvous between two robots exploring an unknown environment. The question is how can two autonomous agents that cannot communicate with one another over long distances meet if they start exploring at different locations in an unknown environment. The intended application is collaborative map exploration. Inherent to multi-agent systems are problems of task division, synchronization and coordination. Communication issues between agents is also a limiting factor in realistic multi-agent systems. In this work a realistic, limited range "line-of-sight" communications model is assumed. The solution for these problems along with the issue of merging maps between the agents once communication and rendezvous have been established are termed by Roy as "Multi-agent Rendezvous".

1.2.1. *The Rendezvous Problem.* The problem discussed by Roy is how to determine the best strategy for a successful rendezvous between two agents in optimal time. Consider of how the rendezvous task can be efficiently accomplished under assumptions regarding the environment and the perceptual abilities of the agents. Multi-robot exploration using rendezvous is considered in the context of unknown environments. Initially an "abstract" sensor that allows the agents to recognize one another when they are sufficiently close together is assumed. This sensor is also used to evaluate discovered areas in the evolving exploration as to their suitability as a rendezvous points.

The rendezvous task itself is divided into two sub-problems:

(i) Identification of appropriate rendezvous points. The agents must have some way of independently choosing some set of locations as appropriate points for attempted rendezvous.

(ii) The ability for agents to agree on the same location for rendezvous. Given a set of appropriate potential locations, how should rendezvous be attempted within these locations? An appropriate rendezvous strategy will take into account the possible asymmetry between the agents' exploration as well as asynchrony between the agents, and allow for missed rendezvous attempts.

Rendezvous will involve the agents searching through the environment for good meeting points, and then traveling to the best meeting point at a pre-arranged time. The conceived rendezvous problem consists of four steps to be performed:

(i) Explore the environment
(ii) Categorize areas according to goodness as rendezvous points
(iii) At a pre-arranged time, choose the best rendezvous location
(iv) Travel to the best location for rendezvous

The problem of rendezvous is also concerned with how to proceed if a particular rendezvous attempt fails. Important are Strategies that permit a robot to interleave exploration and attempted rendezvous so that even in the case of failed rendezvous the robot can continue its work. Such an approach is indicative of good robotic system design.

1.2.2. *Formal Parameters of the Rendezvous Problem.* Fundamental to rendezvous is the ability to identify rendezvous points in the environment. One method to identify rendezvous points is to use a potential function at all points in the known environment and taking the points which maximize that potential function as the candidates for rendezvous points. These points are often called distinctive points in the environment. The potential at any point is then determined by a *distinctiveness function*. The **distinctiveness** is a value defined at every point in the environment as a function of the location in the environment and the sensor(s) used to gather information about the environment at that point.

To try and coordinate rendezvous as best as possible it is clear that each of the agents should use identical distinctiveness functions with which to categorize the environment. Still differences in the agents' sensors will result in different perceptions of the environment and hence different distinctiveness functions will be assigned, even with the environment common to each agent. Compensation for these differences is also a necessary aspect to the rendezvous problem. In order to manage this compensation for independent error, in hopes to arrive at a common perception of the environment, a parameterization of the distinctiveness measure, $D(x, y)$, is performed. The parameters chosen by Roy for this job are systematic differences between agents, and random noise. $D(x, y)$ is also a function of the sensors the agents use:

$$\begin{aligned} \vec{S}_i(x,y) &= \vec{S}(x,y) + \vec{\eta}_i(x,y) + \vec{\lambda}_i && (2.6) \\ D_i(x,y) &= D(\vec{S}_i(x,y)) && (2.7) \end{aligned}$$

where $\vec{S}(x, y)$ is the ideal perception of the environment by the given sensor, in the absence of noise. $\vec{S}_i(x, y)$ is an agents perception of the environment at position (x, y), encapsulating the agent's systematic error $\vec{\eta}(x, y)$ over the measurement at that position and $\vec{\lambda}_i$ is an agent's random sensor noise.

Then in order to quantify the inter-agent differences, λ and η are modeled as scalars, the random and system errors are collapsed into one term. These inter-agent differences are expressed for any agent relative to a reference agent, represented D_1

$$\begin{aligned} D_i(x,y) - D_1(x,y) &= \hat{\delta}_i\hat{\eta}_i(x,y) + \hat{\delta}_i D_1(x,y) && (2.8) \\ D_i(x,y) &= (1-\hat{\delta}_i)D_1(x,y) + \hat{\delta}_i\hat{\eta}_i(x,y) && (2.9) \end{aligned}$$

where $\hat{\eta}_i(x, y)$ is all stochastic and systematic noise processes of each robot, and $\hat{\delta}_i$ specifies the extent to which the two robots (D_i and D_1) sense (or perceive) the same thing. Modeling differences in sensor measurement across agents, the $\hat{\eta}$ and $\hat{\delta}$ parameters can be treated as a single parameter, δ.

Roy identifies critical attributes that affect the performance of rendezvous strategies. These attributes envelop issues regarding the actual robots themselves as well as issues regarding the environments within which the robots must manage rendezvous. The parameters of the rendezvous search problem are listed as follows:

- Similarities — The coincidence of the agents' perceptions. This is greatly dependent on the kind of sensor used for the measurement. This is expressed as $\hat{\delta}$ of Equation 9.
- Sensor noise — The $\hat{\eta}(x, y)$ term of Equation 9, is meant to encapsulate the independent differences from ideal perception that an agent makes at a given location. This effect leads to the need to consider a greater number of less-than-ideal rendezvous points.
- Landmark Commonality — The extent of overlap between the explored areas of the different agents. Landmarks that fall outside of this overlap are useless as rendezvous points. This disparity forces that multiple rendezvous points be considered.
- Synchronization — Appropriate action must be taken by the agents to overcome failed meetings. This effect leads to a need for strategies that may re-visit the same landmarks repeatedly to compensate for missed meetings.
- Landmark Cardinality — The manner in which different landmarks are ranked and the number of ranked landmarks that are considered for rendezvous points.

These parameters of the rendezvous search problem lead to an adoption of a necessarily robust scheme for performing real world rendezvous. Failure and variation from expected behaviour are aspects of functional mobile robotics that must be addressed. This aspect of Roy's work is precisely the merit of it's discussion here.

1.2.3. *Landmark Selection Algorithms.* There are two main classes of algorithm considered for the selection of landmarks to visit: deterministic and probabilistic. The deterministic class of algorithm creates a list of all possible combination of landmarks and specifies the order in the list. The probabilistic class of algorithm does not generate an ordering of landmarks, but simply generates probabilities for landmarks being visited. Brief descriptions of the types of algorithms are given:

- Deterministic Algorithms — these algorithms always create the same ordering of landmarks.
  - **Sequential** – One robot waits at a landmark while the other robot searches all of its landmarks. If rendezvous fails the first robot moves to a different landmark and waits.
  - **Smart-sequential** – Each pairwise combination of landmarks known to a robot is assigned a value of the product of the distinctiveness of the pair. The list of landmark pairs is sorted and the robot then visits the landmarks in this order.
- Probabilistic Algorithms — The landmarks are sorted with respect to their distinctiveness and then assigned a likelihood of visitation p_i for landmark i. The algorithm probabilistically selects a landmark to visit, using p_i for each landmark.
  - **Exponential** – The likelihood of visiting the $i - th$ best landmark is $\propto e^i$. This accentuates relatively highly distinctive points.
  - **Random** – On each attempted visit, each robot selects a landmark at random and goes there.

The success of rendezvous attempted using each of these algorithmic approaches was compared and assessed. These algorithms were the variables under study in the experiments.

1.2.4. *Multi-Agent Exploration.* Of particular interest in Roy's experiment is the ability for the rendezvous algorithm to overcome the communication restriction and yet maintain the increase in speed that multiple-agent robotics promises. A speed increase is demonstrated for exploration, even accounting for the time to rendezvous.

The increase in speed, ΔS, of the mapping process is then given by Equation 12,

$$\Delta S = \frac{S_{combined} - S_{single}}{S_{single}} \quad (2.10)$$

$$= \frac{\frac{A_c}{T_c} - \frac{A_s}{T_s}}{\frac{A_s}{T_c}} \quad (2.11)$$

$$= \frac{A_c}{A_s}\frac{T_s}{T_c} - 1 \quad (2.12)$$

Taking the area of a single agent, A_s as the area explored by the active agent, and the time of the single agent T_s as the time allowed for the exploration process alone. The combined area, A_c was the explored area of the merged maps, and the combined time, T_c was the time to explore, T_s added to the time to rendezvous, T_r so that $T_c = T_s + T_r$.

Index	Active	Passive	Combined	% Increase in Area
0	48.0	42.2	69.1	44.2
1	48.0	58.4	72.0	50.2
2	48.0	66.5	74.6	55.6
3	48.0	59.5	68.5	42.8
4	48.0	67.7	73.9	54.1
Average				49.4

TABLE 2.2. The increase in explored area, as a percentage of the environment, for each of the 5 initial configurations.

Table 2 (from [62]) shows the average speed increases observed by use of each of the algorithms from the previous section. As Table 2 shows, the increase in explored areas was a minimum of 42.8%, and on average 49.4%. The data expressed in this table assumes total communication between the robots and hence expresses ideal speedup based on the rendezvous algorithms used.

1.2.5. *Section Summary.* This work exhibits a practical multi-agent rendezvous scheme to overcome practical communication limits by periodically having the agents converge and share information. It shows the efficacy of a multiple robot system compared to the single robot system, while eliminating the traditional assumption of global communication between agents.

The rendezvous problem was divided into two separate subproblems. The first is determining what points in the environment constitute good rendezvous locations. This problem is addressed by modeling the environment as a function of the sensors giving rise to a distinctiveness surface, defined over the domain of the environment.

Which points the robot visited was dictated by the trajectory prescribed by the underlying task, and so demonstrates a mechanism to overcome trajectory dependencies.

Three key parameters that characterize the problem are identified. By choosing a number of different points in the environment for meetings, an intelligent scheme for visiting these points allows rendezvous to be achieved reliably. The three parameters are identified as *sensor noise*, *map commonality* and *asynchrony*.

The problem of which appropriate behaviour to use in choosing the landmarks to visit is the second of the two subproblems of rendezvous. Two main classes of algorithms are proposed *deterministic* and *probabilistic* guidance algorithms.

The rendezvous algorithm was demonstrated both in simulation and on physical robots. The simulation tests were used as a confirmation of the numerical results. An interesting conclusion from these results is that, depending on a combination of these confounding factors, no strategy is canonically a good or bad choice. The experiments using physical robots gave a compelling demonstration that the rendezvous algorithms are an essential part of the rendezvous process.

2. Exploration Strategies

We have seen (equations 5, 4) that the choice of an appropriate search algorithm is directly dependent upon the environment that we are considering. It is well worth identifying some environments and optimal search strategies to get a feel for what kind of results we will expect to find as our environments become less constrained and harder to analyze. Bearing in mind Moore's Law[3], the processing power available in current mobile robots

[3]A rule of thumb proposed by Thomas Moore in the late 1960's that microprocessor speeds will continue to double roughly every 18 months. This shocking exponential claim has been consistently accurate for the last 30 years, and according to current estimates will likely continue for at least another 10–15 years. See [37] for a more complete discussion of the growth trends in microcomputer architecture.

far outweighs their mobile capabilities, this is a trend that we can assume will persist. Any sort of reduction in physical motion, especially when the cost is processing time, is a worthwhile investment.

What sort of approach should be used for determining this strategy? We will agree that the actual environment we are interested in is the plane. Searching in more general environments is in itself an interesting problem, but here we will restrict our interest to domains topologically equivalent to the plane since that is where our mobile agents will be working.

By investigating search paths in the plane it is possible to gain a better understanding of the efficacy of search strategies that we may wish to employ. Baeza-Yates, Culberson and Rawlins **[11]** show some interesting findings regarding exactly this problem of finding optimal search paths in the plane. We will look at their results in section 2.2.

It is well known that a traversal through a graph-like environment according to a "depth-first" approach (section 2.1) can be performed optimally in $2 \times \|V\|$ edge movements, where $\|V\|$ is the number of vertices in the graph. Depth-first search, or *DFS*, is a sure way to traverse the graph, but it is inappropriate for our task. In unknown and potentially [practically] unbounded environments DFS may not perform well in terms of average case behaviour. Unreliable cycle detection poses a real problem while performing DFS. One of the failures of DFS is that a failed cycle detection will lead to the robot falsely believing that it has continued to explore new ground while walking in circles. Even bounded failure will result in a serious topological morphing of the search environment that would not be the case in breadth-first searching that does not allow itself to descend deeply into an error. This difficulty associated with recognizing cycles[4] in the graph, the failure of adhering to the *Principle of Locality*[5] and the fact that the environment is unknown *a priori* makes DFS an undesirable approach.

[4]the trouble with cycle detection in a robotics setting comes from a difficulty in accurate position detection in the presence of noisy unreliable sensors.

[5]Rule of thumb regarding cache hierarchies (see **[37]**). The principle posits that references that are located proximally tend to be accessed proximally. Here the principle is extended beyond its purpose to indicate that average case behaviour is very important while searching. An object that was lost 10 minutes ago is *misplaced*, where as an object misplaced 10 days ago is more likely *lost*.

2.1. Depth-First Search. Depth-first search is a simple search strategy that expands a graph-like structure by searching deeper into the graph whenever possible. A more detailed explanation than will be presented here can be found in [**21**].

According to the depth-first strategy any unexplored edges leading from the most recently discovered vertex are explored before any further unexplored edges are taken leading out of the parent of the current vertex. This process is applied recursively at every vertex and continues until all of the nodes in the graph have been explored. All of the nodes neighbouring the current node μ, aside from the predecessor, are said to be *children to* μ. The pseudo-code shown as algorithm 1 shows the details of a recursive depth-first search algorithm which keeps track of the discovery time of each node in the *SearchMe()* routine.

Algorithm 1 DEPTH-FIRST_SEARCH(μ)

```
for $\nu \in Child[\mu]$ do
  if $\nu$.hasNotBeenSearched() then
    Depth-First_Search($\nu$)
  end if
end for{Children of $\mu$ will be searched before $\mu$}
$\mu$.SearchMe()
```

Depth-first search has several redeeming features that make it a good strategy to start with. Firstly, it is exhaustive. We are guaranteed that every node will be visited eventually. Secondly, it performs minimal re-traversal of the graph. With DFS we are assured that no edge will be visited more than twice[6], which is a great property bearing in mind the burden of physical motion for mobile robots. Thirdly, this algorithm has minimal memory space requirements. The memory requirements for the algorithm consist of only needing to remember the graph itself and the path to the current vertex. The memory size requirements are thence $\mathcal{O}(||V|| \log ||V||)$. This allows the robot not to have to try to match its current location to a vertex in the graph since this is implicitly accomplished through the topology of the graph. Since reliable localization is a difficult problem in its own right, this is a very attractive property of depth-first search. Despite these qualities DFS is not exactly what we

[6]Whereas one of the major drawbacks with breadth-first search is that vertices with children are visited $2 * ||subtree||$ times. This does not factor in asymptotically to the performance, but it is a consideration in reality for our slow moving agents.

need. There are major shortcomings of DFS that we need to address for use in real, *a priori* unknown, environment that we will be dealing with.

In unbounded environments DFS stands to perform perhaps very poorly, since it may never[7] successfully descend the depth of the entire search tree. The principle of locality is also sacrificed by DFS, something that we do not wish to give up on so easily.

2.2. Searching the Plane. Consider the series of results collected by Baeza-Yates, Culberson and Rawlins in [**11**]. We can get a flavour of how we may do better than DFS in a planar environment by adopting the trend suggested in the hierarchy of table 3. The trend to discover is that the family of *logarithmic spiral curves*, properly constrained to the appropriate environment, provide optimal solutions to search problems. We see this trend developed here in environments that are simple enough to study rigorously. It is our intention that with this motivation that later, when looking for efficient search strategies in less confined environments, that we may extrapolate these results to provide a mechanism for efficient search in dynamic real environments.

Under the assumption that we have to pay a significant cost to "probe" in our environment proportional to the distance of the probe from our current location there may be better ways to search. Table 3 shows a collection of known best results for such search problems under the restrictions given.

The problem of searching in the plane for a line with different kinds of *a priori* knowledge regarding the line is an illustrative example. Starting at the origin, $(0, 0)$, we begin looking for the line. Supposing that the line is a distance n steps away from the origin and that travel is performed taking unit "steps". We assume that we detect the line as soon as we encounter any point in the line. In the real world this is a non-trivial condition if one considers an example such as a target that can be identified as soon as there exists a direct line of sight from the agent to the target.

To demonstrate the results of table 3 we suppose that we have a variety of different types of information concerning the line before starting.

[7]perhaps "never" should be qualified with "for all intents and purposes". Bearing in mind of course that the relatively slow physical motion of real robots may require an infeasible amount of time to search the extant of an environment.

Info Given	Search Distance
Normal of a line	n
Distance, Slope	$3\,n$
Distance, Axis Aligned	$3\sqrt{2}\,n \approx 4.24\,n$
Distance to line	$(1+\sqrt{3}+7\pi/6)n \approx 6.39\,n$
Slope only	$9\,n$
Axis Aligned	$13.02\,n$
Nothing	$13.81\,n$

TABLE 2.3. An enumeration of distances searched to discover a line that is a distance n from the origin. The hierarchy reflects varying kinds of knowledge regarding the line. Competitive ratios for are found by dividing by n. These worst-case values are reported in [**11**]

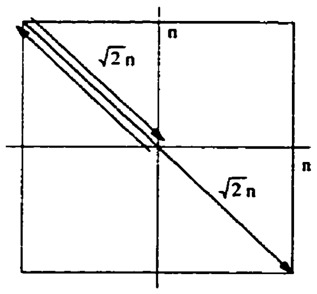
2.2.1. *Searching for a Line given its Normal.* When the [directed] normal to the line is known, the problem is trivial. We know that the distance to the line is n and the direction that the line lies in relative to the origin. This condition is equivalent to knowing the solution. Recall that the shortest distance from the origin to a line is in the negative direction of the normal.

Our direction of motion is constrained to $\mathbb{R}^2/\mathbb{R} = \mathbb{R}$, a line and we know which direction the line lies in, so walking along the normal in that direction will yield our result in exactly the minimal n steps.

2.2.2. *Searching for a Line Given Slope and Distance.* When the distance and slope of the target line are known at the outset, but not the side of the origin to which the line lies then there is some ambiguity to consider. The direction in which the line lies is unknown. This problem is equivalent to finding a point in a line knowing the distance.

Again our search is constrained to a line perpendicular to the slope, but unlike the previous case we do not know in which direction the line lies. So we walk along the direction perpendicular to the line for a distance n, and if the line is not on that side then we know that we chose wrong initially and that the line is on the other side of the origin along the perpendicular. The total worst-case distance is thus $3n$.

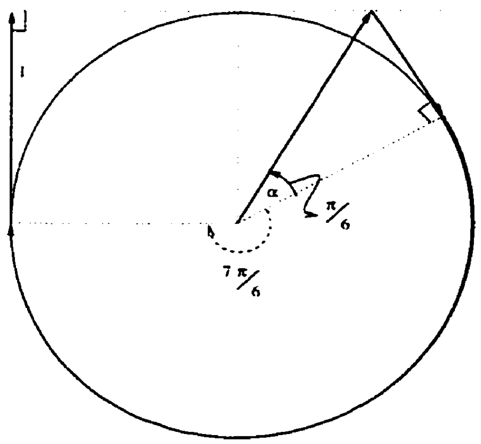
2.2.3. *Searching for an Axis Aligned Line Given Distance.* In this version, it is known that the line for which we search is parallel to one of the axes, but which is not known. With the information that we are given we know the line that we search for is one of the sides of the square shown in the following diagram. It is easily seen that the problem can be solved by investigating opposite corner points in the square formed by the possible configurations of the problem.



Again this results in finding a particular point on a line without knowing the correct direction (see diagram). In this problem the distance to either of the points is $\sqrt{2}\,n$ yielding a worst-case performance of $3\sqrt{2}\,n$.

2.2.4. *Searching for a Line Given only Distance.* This case is one of the least intuitive of the cases presented. It is worth spending a few moments to consider some possible approaches before reading on to the solution. The question and first [non-optimal] solution date back to 1956 from Bellman [**14**] and shortly afterward, in 1957 Isbell [**40**] presented the solution in a brief [and difficult to find] paper which is optimal.

Given the distance to the target, but no more information, it is clear that the target lies somewhere on the circle of radius n centered at the origin. A quick calculation shows that a tour of $n + 2\pi n \approx 7.28n$ is an upper bound, by walking to any point on the circle and then traversing the circumference. Isbell has shown that we can do better than this according to the following method which asserts the optimal solution is achieved in $(1 + \sqrt{3} + \frac{7\pi}{6})n \approx 6.397n$ metric units.



Assume a circle radius $n = 1$ and centered a p. We have the following eloquent explanation by the original author:

> Being at p and unoriented, imagine a clock face. Walk toward one o'clock for $\sqrt{\frac{4}{3}}$ units. (This takes you to a vertex of a circumscribed regular hexagon.) Then turn on the tangent which strikes the unit circle at two o'clock. Follow the circle to nine o'clock and continue on a tangent. Upon striking the line which is tangent to the unit circle at twelve o'clock you have swept all the tangents to the unit circle, and that in a path of minimum length.
>
> *Isbell* **[40]**

The proof amounts to finding the minimum for

$$f(\alpha) = \sec\alpha + \tan\alpha + \pi - 2\alpha, \quad \alpha \in [0 .. \frac{\pi}{4}]$$

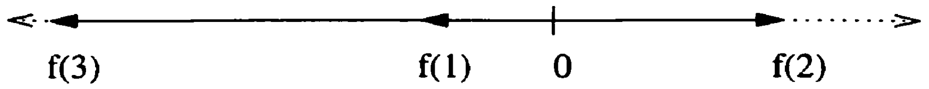
which is indeed achieved at $\alpha = \frac{\pi}{6}$.

2.2.5. *Searching for a Line Given Slope.* Knowing only the slope of the potential line but not the distance is equivalent to the problem of finding a point in a line, since it only makes sense to walk along the perpendicular to the slope. Here we do not know the distance to the line. This case is equivalent to searching for a *point in a line* an unknown distance away. An analogy might properly be made to searching a darkened hallway for a light switch, when the searcher has no expectation for either how far the light switch might be nor for the length of the hallway.

We can describe the solution as a function $f(i) : \mathbb{Z}^+ \to \mathbb{Z}^+$ where $f(i)$ reports the number of steps to take to the right or left of the origin before the i-th turn. This function, provided with the *progress* condition of,

$$f(i) \geq f(i-2) + 1$$

is enough to specify a solution to the problem. The *progress* condition simply ensures that our solution will indeed explore new territory each time there is a change in direction. This is assured by taking at least one more step into unexplored territory than has been taken so that the search will terminate. Since this is an alternating search, $f(i)$ and $f(i+2)$ are consecutive distances from the origin on the same end of the line.



Baeza-Yates *et al.* [11] show that the following function,

$$f(i) = 2^i, \quad \forall i \geq 1 \tag{2.13}$$

results in an optimal solution. With this function the total distance that is walked is $2\sum_{i=1}^{\lfloor\log n\rfloor+1} 2^i + n$, since we walk out and back on every iteration of 2^i steps except for the last probe, during which we walk n. This is easily seen to be bounded by $9n$ steps,

$$\begin{aligned} 2\sum_{i=1}^{\lfloor\log n\rfloor+1} 2^i + n &< 2\left(2^{\lfloor\log n\rfloor+2}\right) + n \\ &\leq 2\left(2^2 \cdot 2^{\lfloor\log n\rfloor}\right) + n \\ &= 2^3 n + n \\ &= 9n \end{aligned} \tag{2.14}$$

with a little bit of slack in the first inequality. In fact there exist algorithms which perform at $9n - \Theta(\log n)^i$, $\forall i$. It can be shown that the class of *logarithmic spiral curves* will achieve this bound in general. These performance properties of the logarithmic curves will be exploited in general framework for mobile robotic exploration. The proof that linear spiral search is optimal up to lower order terms is deferred to appendix A.

	Known	Search Distance
Section 2.2.1	Direction and Distance	n
Section 2.2.2	Distance	$3n$
Section 2.2.5	Nothing	$9n$

TABLE 2.4. This sub-table of 3 illustrates the 1 dimensional subproblem of finding a point in a line with various amounts of *a priori* knowledge

It is interesting to note the apparent regression that this result provides. This concludes the 1 dimensional analog to searching the plane. While searching for a point in a line, there are 3 kinds of information that we may have (summarized in table 4). If we know the direction and distance to the point then we walk directly there, taking the required n steps. If we know only the distance, then the ambiguity of which side to search on reveals a worst-case performance of $3n$ making our search 3 times longer possibly. Knowing nothing about the location of the point results in a further complication by a factor of 3. This last case requires a potential $9n$ steps and is the case we have just analyzed.

2.2.6. *Searching for a line Given Little or No Information.* In this instance of the problem, the distance to the line is not known nor is the slope. We have no knowledge of the line beyond that there is a line in the plane. It seems reasonable that an optimal search path that discovers this line will be one that exhibits *spiral self similarity*. That is to say that the curve will expand outwards during the search and that it does not rely on orientation or scale, since neither of these properties are known. This optimal curve will be self similar with respect to both rotations and dilatations. This is a reasonable presumption for a solution curve by converse reasoning. Any curve that does not exhibit these properties will admit some line in the plane that is not found in the minimal number of steps.

More precisely our search space is the plane, so from equation 5 we have that $\mathbb{X} = \mathbb{R}^2$ and our set of isometries are $G =$ (rotation, dilatation). The only known family of curves that exhibit *spiral similarity* in the plane are the curves known as the logarithmic spirals.

The class of logarithmic spirals are defined by the polar equations $r = k^{\theta}$. It is claimed in [11] that the "best" numerical approximation for the logarithmic spiral is such that $k \approx 1.250\cdots$. Whether the authors have some theoretical method of generating this number or whether it is just the number that yields the lowest worst case performance is unclear and unspecified. Clearly we are free to choose k as we wish as it indexes the entire family family of curves. Choosing $k \approx 1.250$ yields the claimed lowest worst case performance of $13.81n + \mathcal{O}(\log n)$ from table lowest 3.

Although this is not a rendezvous method, we can cast it as one, with the rendezvous being between a mobile point agent and a line. The line has no search strategy and the point robot follows the strategy laid out by $r = 1.250^{\theta}$. In this sense we have an asymmetric rendezvous value of

$$\mathcal{R}^a = \mathcal{R}^a(\mathbb{X}, G, 0) = 13.81$$

With the restriction that the line be axis aligned, slightly more information about the pose of the target line is known. This has the effect of lowering the upper bound from $13.81n$ to $13.02n$. These results are arrived at in a similar manner as before, but the limits are approximated using numerical methods.

This concludes the explanation of the entries in Table 3. It also raises some interesting points. The method of search that is employed seems to be rather sensitive to the amount of *a priori* knowledge that we have of the environment and especially of the properties of our target therein. Selecting the search method that we should use is not always clear and the choice is dependent upon several factors, like the principle of locality, that are not necessarily related to efficient search in other common domains. Searching in noisy, dynamically changing and unknown environments is a difficult problem.

The important observation at this point is that there are a set of properties which underly this entire series of solutions. There is a trend to a general solution method upon which the rest of this work relies. The identification of the class of logarithmic spirals as the family of curves which yield the optimal search strategies in their respectively constrained environments. This general method employs the family of curves that we have already introduced and whose properties we will consider next.

2.3. Logarithmic Spirals. In retrospect here we observe an interesting aspect of this hierarchy of inquiries. Aside from the case of finding the shortest distance to a line given the distance, n, to the line[8], a general trend of *logarithmic spiral search* restricted to the appropriate domain serves as an approach which uniformly delivers asymptotically optimal performance.

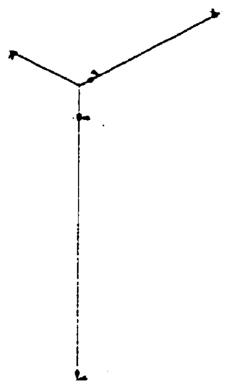


FIGURE 2.1. The path traced by $f(i) = \left(\frac{3}{2}\right)^i$ intersecting with $m = 3$ concurrent rays.

2.4. M Concurrent Rays. The case for searching m-concurrent rays is very similar to that of searching for a point in a line, but offers further insight into that same problem. The assumptions here are that the robot is standing at the [common] intersection of m rays [see definition 2.1] and endeavours to find a target located on one of the rays. If the distance to the target is known, but not which ray the target lies within (comparable to

[8]This is a special case in that the domain of the search is bounded. We know that the solution will be found on the circle of radius n. This condition separates this case from the others, whose domain extent is infinite. As we will see this is the mark of a problem whose solution may be best approached using a logarithmic spiral search method.

knowing distance but not direction) then the obvious $(2m - 1)n$ algorithm is optimal. If the distance is unknown, then this becomes the counterpart of the point-in-a-line problem from section 2.2.5 and yields some interesting results.

DEFINITION 2.1 (Concurrent Rays). *Rays or search paths that emanate from a common intersection and extend without further intersections are said to be* concurrent. *If from that common intersection point there are M such co-terminal rays, we say there are* M-Concurrent Rays.

This problem is worth addressing as it will turn out to be the correct setting for the experimental work presented in chapter 4. In fact a deviation of this problem will be what is specifically needed, when we drop the assumption that we have concurrent rays. Being able to speak quantitatively of searching in non-concurrent rays will be important later on and this nomenclature will serve to be useful throughout this paper. In environments with a "star shaped" nature, this is the most useful approach. Although the theory will be extended to handle cases homeomorphically [see B.5 for a definition] similar to star polygons, the concepts will remain essentially the same.

Given that we have m rays, the order is not important since the names of any of the rays are simple labels and can be permuted, so cyclical addressing is fine from $1, 2, \ldots, m$. Later in the case of non-concurrent rays will this issue come up again, since then, indeed, the order of visitation will make a difference in the overall performance. Again we see that in order to study the problem, we have the *progress requirement* that the function, $f(i)$, which relates how far to walk from the origin before the $i - th$ turn is,

$$f(i) = f(i - m) + 1 \qquad \forall i \geq 1 \text{ where } f(-j) = 0 \, \forall 0 \leq j \leq m - 1 \tag{2.15}$$

This has been coined by Baeza-Yates *et al.* **[11]** as *generalized linear spiral search* as presumably opposed to just *spiral search*, since the points we are interested in are the intersection points of the two sets in figure 1 and the actual spiral set which is the set of dotted points forming the logarithmic spiral depicted. The generalized linear spiral search algorithm is defined in a similar manner to the linear spiral search from equation 13 as the number of steps to walk before the ith turn starting from the origin. It can be shown (see [**11**] and appendix A.2) that,

$$f(i) = \left(\frac{m}{m-1}\right)^i \qquad \forall i \geq 1. \tag{2.16}$$

will yield optimal performance. Thus we are instructed about what base to choose for the denominator in the tell-tale function. In the point-in-a-line case ($m = 2$) we have $f(i) = 2^i$, for the case of $m = 5$, for instance, we see that the function we should use is $f(i) = (5/4)^i$. This comes naturally from the proof (see A.2), with a worst performance ratio of,

$$1 + 2\frac{m^m}{(m-1)^{m-1}} \qquad \text{(for large } m\text{)}.$$

So equation 16 provides for us an online way to determine the optimal distance that should be walked during this kind of search. Upon realizing that we are faced with some number, M, of co-terminal rays to search and being at the common root, we can use M and equation 16 to infer the search distance that will make our efforts most effective. This fact will be called upon for our further generalization in chapter 4.

CHAPTER 3

Rendezvous Strategies

Rendezvous strategies provide a scheme whereby two or more agents in a common environment, and wishing to meet, can proceed in a manner that will result in an encounter. Rendezvous can be cast as a classic two player, zero-sum game, and the resultant analysis from game theory [36] can be useful for many restricted settings [1, 2, 4]. Rendezvous between mobile robots in a real environment is more complex and requires graceful failure routines for the continued operation of the robotic system. Such a system has been designed and tested by Roy [62] and we have seen some of the details of that work in chapter 2.

In this chapter we consider the purely algorithmic solutions to the symmetric rendezvous search problem in the plane. This will be an ideal formulation and will not accommodate obstacles in the plane to interfere with the search. Handling the inclusion of obstacles in the plane is certainly not a difficult modification but we do not need to accommodate obstacles to arrive the long term properties of the rendezvous strategies, which is the goal.

1. Rendezvous Search in the Plane

We consider the case of 2 unit speed agents searching the plane for a single target. The agents can sense a region in the plane in a disk about their position of radius, r. We will assume that both agents adopt the same search strategy, so that without loss of generality, we can perform analysis on the half-plane and a single agent.[1]

Following the advice of [**11**] we will adopt the appropriate model of spiral search as the exploration strategy. In this circumstance, spiral search simplifies to walking concentric circles of odd radius, so that the boundary of the sensed regions just touch. This leaves a measure zero set unsearched so we are guaranteed to find the target if it lies within the searched region. See figure 1 for an illustration of the search pattern.

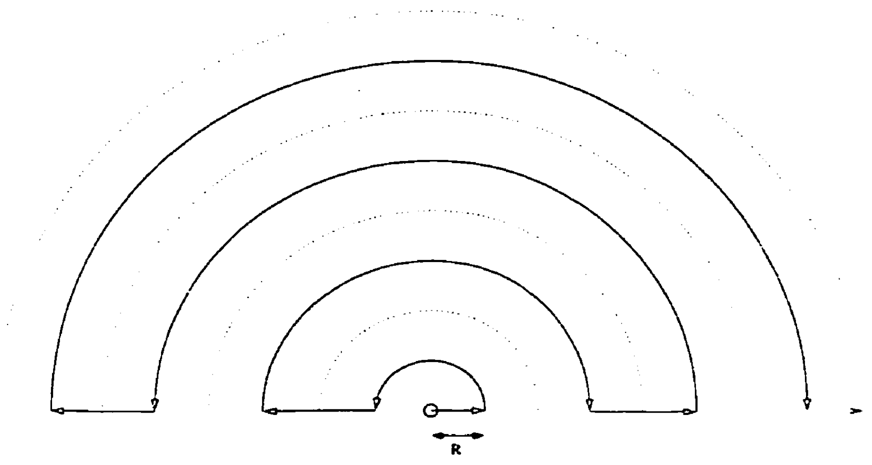


FIGURE 3.1. The path traced by σ_∞ of concentric hemi-circles. The solid lines indicate the path the actual robot will sweep and the dotted lines indicate the boundaries between points covered in consecutive sensing passes. The robot can sample points a distance r from its physical location.

The family of rendezvous strategies that we will consider can be described as follows. Every time our agent arrives back at the border of the half plane it has a choice to move back to the origin to meet with the other agent, or continue to search the next level of its region. We identify these alternative behaviors with an index σ_i. We will evaluate and analyze the cost of the alternate behaviors and attempt to classify the family of strategies according to the distances that they travel relative to one another. The distances that we will compare will be the total distance traversed. *Distance traversed* will be the total distance traveled by the agents, whether resulting in further search or in attempted rendezvous. Assuming unit

[1]The *asymmetric* case where each agent may adopt a different strategy is a future expansion of this analysis, that would have to include the probability of failed rendezvous attempts in the cost function.

speed agents, there is a directly proportional relationship between the distance traversed and the time spent searching. These σ_i's will be the strategies employed, which can now be defined.

Let $\sigma_i \in S$ be the strategy that rendezvous is attempted upon every i-th arrival at the search boundary. Let $\mathbb{Z}$ be the positive integers then rendezvous is attempted at border encounters from the set, $\{\mathbb{Z} \setminus \mathbb{Z}_i\}$, for strategy σ_i. Define σ_∞ to be the strategy ensuing from spiral search without rendezvous attempts (since $\{\mathbb{Z} \setminus \mathbb{Z}_\infty\} = \emptyset$).

THEOREM 3.1 (Hemi-Planar Search Distance). *Given sensor radius, r, the distance traversed by a hemi-planar search strategy that does not attempt rendezvous is*

$$D(\sigma_\infty) = \sum_{i=1}^{\infty} [2r + (2i-1)\pi r]$$

PROOF. The proof is a simple matter of adding the terms in the progression of the different legs of the search.

$$\begin{aligned} D(\sigma_\infty) &= (r + \pi r + r) + (r + 3\pi r + r) + (r + 5\pi r + r) + (r + 7\pi r + r) + \cdots \\ &= 2r + \pi r + 2r + 3\pi r + 2r + 5\pi r + 2r + 7\pi r + \cdots \\ &= \sum_{i=1}^{\infty} [2r + (2i-1)\pi r] \end{aligned} \tag{3.1}$$

□

Clearly this distance is infinite, since σ_∞ never attempts to rendezvous. The family of σ_i, $i \in \mathbb{Z}$ are defined by the frequency that they attempt rendezvous at. For the strategy σ_3 for instance, rendezvous would be attempted at the origin after searching for 3 turns since the last rendezvous attempt. Rendezvous attempts require a walk back to the origin, to check to see if another agent is there waiting for rendezvous, and a walk back to where the search was abandoned. Only distance is considered here as a metric.

THEOREM 3.2 (Distance traversed during Rendezvous Search). *Given sensor radius, r, and a family of rendezvous search algorithms defined for searching the half-plane, σ_i, that attempt rendezvous after every i-th turning point, the distance traveled after the n-th* *rendezvous attempt is*

$$D(\sigma_j) = \sum_{i=1}^{n} (4ji - 2)r + (2j^2 i - j^2)\pi r$$

PROOF. With the general pattern provided by equation 17 we can evaluate how each of the search strategies performs in terms of the distance that it travels. We can easily go ahead and enumerate the distance traversed by each strategy in the family, by applying similar analysis. Note that the sum is chosen to reflect the distance traveled between rendezvous attempts, breaking the infinite sum in this way will be useful when analyzing the strategies according to rendezvous attempts and we can safely take the sum term-wise due to the compactness [2] of the plane. Yielding the sequence of distances walked at the n-th rendezvous attempt,

$$\begin{aligned}
D(\sigma_1) &= (r + \pi r + r) + (3r + 3\pi r + 3r) + (5r + 5\pi r + 5r) \\
&\qquad + (7r + 7\pi r + 7r) + \cdots \\
&= 2r + \pi r + 6r + 3\pi r + 10r + 5\pi r + 14r + 7\pi r + \cdots \\
&= \sum_{i=1}^{n} \left[(4i - 2)r + (2i - 1)\pi r\right] \\
D(\sigma_2) &= (r + \pi r + 2r + 3\pi r + 3r) + (5r + 5\pi r + 2r + 7\pi r + 7r) + (9r + \cdots \\
&= 6r + 4\pi r + 14r + 12\pi r + \cdots \\
&= \sum_{i=1}^{n} \left[(8i - 2)r + (8i - 4)\pi r\right]
\end{aligned}$$

[2]because we are working over a compact domain, we are assured that convergent infinite sums will converge to the same point regardless of how we group the terms of the sum. Addition is associative on convergent series on compact domains.

$$\begin{aligned}
D(\sigma_3) &= (r + \pi r + 2r + 3\pi r + 2r + 5\pi r + 5r) \\
&\qquad + (7r + 7\pi r + 2r + 9\pi r + 2r + 11\pi r + 11r) + \cdots \\
&= 10r + 9\pi r + 22r + 27\pi r + \cdots \\
&= \sum_{i=1}^{n} \left[(12i - 2)r + (18i - 9)\pi r\right] \\
&\ \ \vdots \\
D(\sigma_j) &= \sum_{i=1}^{n} (4ji - 2)r + (2j^2 i - j^2)\pi r
\end{aligned} \tag{3.2}$$

$\square$

Splitting the sums up in this fashion allows us to see the pattern that evolves by the members on each of the respective rendezvous attempts. The pattern that evolves is intuitive enough since for each member the distance, traveled between rendezvous iterations consecutive odd traversals of hemi-circles, followed by the trip to the origin. This accounts for the terms in equation 18, the πr term for the exploration trips, and the r term accounting for the rendezvous trips.

Now we can characterize the distance that is wasted, D_w, performing rendezvous by taking the difference of equation 18 and equation 17 resulting in the extra distance traveled at each iteration. This *wasted distance* is distance that would not be traversed by a lone searcher, or by independent searchers in independent environments with no concern for rendezvous. D_w is simply a measure of the extra work that is being done to coordinate the efforts of the agents. This is the largest part of the inefficiency introduced by the parallelization of work. This slowdown due to parallelization is quantified according to *Amdahls Law* [37] of which a variation will be used to study the necessary inefficiency introduced in the parallelization of work.

$$\begin{aligned} D_w(\sigma_j - \sigma_\infty) &= \sum_{i=1}^{n}(4ji-2)r + (2j^2i - j^2)\pi r - \sum_{i=1}^{n} 2r + (2i-1)\pi r \\ &= \sum_{i=1}^{n} 4jir \qquad ,\forall j \end{aligned} \tag{3.3}$$

If this is the accumulated distance that we waste at each iteration then our opportunity cost is searching this far into the next level of the exploration. So what is our expected gain from further search? It is the area, A, that we will see on the next iteration of the search strategy. This is easy enough to figure out: it is the area given by the πr term in equation 18.

THEOREM 3.3 (Hemi-Planar Search Area). *Given sensor radius, r, the area, A, swept in the half plane by the σ_j-th rendezvous search algorithm after the n-th rendezvous attempt is*

$$A(\sigma_j) = \sum_{i=1}^{n}(2j^2i - j^2)\pi r$$

PROOF. The proof proceeds in a similar fashion as theorem 3.2, first enumerating the steps for each strategy, we collect the terms into a closed form and extrapolate.

$$\begin{aligned} A(\sigma_1) &= r\left((\pi) + (3\pi) + (5\pi) + (7\pi) + \cdots\right) \\ &= \sum_{i=1}^{n}(2i-1)\pi r \\ A(\sigma_2) &= r\left((\pi + 3\pi) + (5\pi + 7\pi) + \cdots\right) \\ &= \sum_{i=1}^{n}(8i-4)\pi r \\ A(\sigma_3) &= r\left((\pi + 3\pi + 5\pi) + (7\pi + 9\pi + 11\pi) + \cdots\right) \\ &= \sum_{i=1}^{n}(18i-9)\pi r \\ &\vdots \end{aligned}$$

$$A(\sigma_j) = \sum_{i=1}^{n} (2j^2 i - j^2)\pi r \tag{3.4}$$

□

Then from equation 21 we see that what we expect to gain is area of the next iteration of search for our strategy, σ_j. This is in inverse proportion to the accumulated distance that we have walked for rendezvous purposes. That proportion is what we need to understand.

$$\begin{aligned} \frac{Ex(\texttt{gain from search})}{Cost(\texttt{rendezvous})} &= \frac{\texttt{Area next Level}}{D_w(\sigma_j - \sigma_\infty)} \\ &= \frac{(2j^2 n - j^2)\pi}{\sum_{i=1}^{n-1} 4ji} \qquad \text{from (20),(19)} \\ &= \frac{(2j^2 n - j^2)\pi}{\frac{4j(n^2-n)}{2}} \\ &= \frac{(2j^2 n - j^2)\pi}{2jn^2 - 2jn} \end{aligned} \tag{3.5}$$

The *gain from search* refers to the area that can potentially be gained by choosing to search further at the next turning point. The *cost of rendezvous* is the distance that we have to cover, of already visited territory, to attempt a rendezvous with another agent. Both of these factors are explicit and simply refer to the distance traversed by the agent.

The function illustrated in figure 2 shows the shape of the behaviour of choosing increasingly large search strategies. The height of the function gives an indication of the time *wasted* by our chosen search strategy relative to σ_∞. It makes sense to have strategies of $j < 1$, which would correspond to not exploring the entire semi-circle before returning to the rendezvous origin, perhaps in circumstance where the sensing region was very large relative to the speed moved, $v \ll R$.

2. A Competitive Ratio Examination

Assuming that we know the distance to the target for multiple agent search it is easy enough to derive the formula for the distance traveled for each search strategy employed.

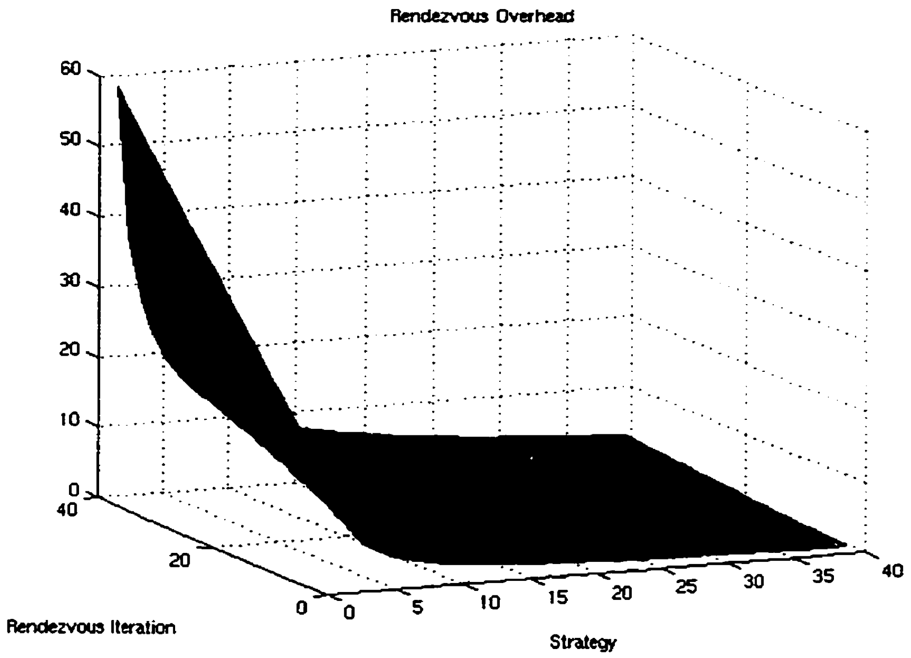


FIGURE 3.2. The Quantitative Performance prescribed by equation 21 of exploration versus rendezvous overhead.

If the distance is known to be d, the sensor radius r, then number of rendezvous attempts made will have to be $\lceil \frac{d}{rj} \rceil$ for strategy σ_j, where $\lceil \cdot \rceil$ is the integer *ceiling* operator. Then the distance traveled by each strategy, $D_d(\sigma_j)$ is given by

$$\begin{aligned} D_d(\sigma_j) &= \sum_{i=1}^{\lceil \frac{d}{rj} \rceil} (4ij-2)r + (2i-1)\pi j^2 r \qquad \text{(from 18)} \\ &= \left\lceil \frac{d}{rj} \right\rceil \left((\pi j^2 + 2j) \left\lceil \frac{d}{rj} \right\rceil + 2jr - 2r \right) \end{aligned}$$

and the resulting curve is plotted, for example $d = 10$, in figure 3.

Of course the competitive ratio that we are examining here is in comparison to the distance traveled by an omniscient agent that simply walks to the target and back to the rendezvous location, for distance of $2d$.

$$CR = \frac{D_d(\sigma_j)}{2d}$$

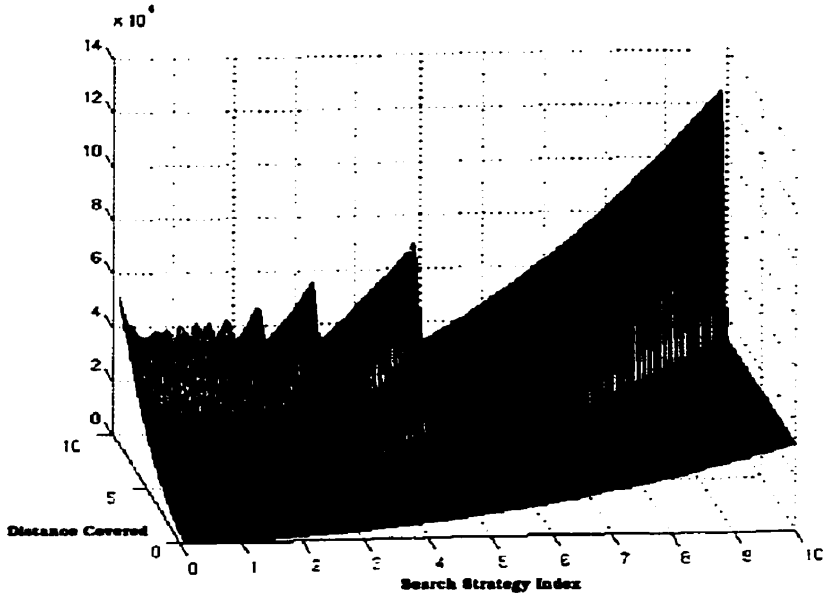


FIGURE 3.3. A plot of distances walked by the different strategies for a target at distance, $d = 10$. Here $r = 1$. Note: the curve is monotonically increasing to the right after σ_{10}. The graph is layered by search strategies. $d = 10$ was chosen to compare the growth of the various strategies without occlusion.

3. Bounded Distances

If d can occur equally likely in the discrete range $[1..D]$ then we have a uniform distribution on the probability of $D_{Tar} = d$ with an upper bound of D. So

$$\Pr(D_{Tar} = d) = \frac{1}{D+1} \qquad \text{on } [0..D]$$

and figure 3 shows that the minimum distance traveled occurs for strategy $j = d$. What strategy is now our best? That follows from

$$\begin{aligned} j &= \frac{1}{D}\int_0^D x\,dx \\ &= \frac{1}{D}\left|\frac{x^2}{2}\right|_{x=0}^{D} \\ &= \frac{D}{2} \end{aligned}$$

easily enough.

The general case is found by integrating over the probability distribution on the region[3] if we assume a uniform distribution over the surface area. This case can be arrived at by

$$j = \int_{\delta D} \Pr(x) \cdot x \, dx$$

where δD is the boundary of the enclosing space.

4. Summary

From equation 21 we have calculated the payoff schedule for the different rendezvous strategies. The quantitative results are shown in figure 2. The qualitative payoff schedule allow the development of an intuition for what the different strategies really provide during planar search.

The result of quantifying the progression of the search allows for the development of the competitive ratio of the performance of each strategy to that of an omniscient searcher with ideal search behaviour. A plot of the efficiency of each of the rendezvous strategies is shown in figure 3.

Section 3 details how the analysis can be modified to relax the restriction that the distance to the target is known. With this relaxation we assume a uniform probability over the *bounded* region for the search and show the necessary resulting modifications. It makes no sense to allow $D \to \infty$ and to speak of average performance with uniform distribution over an infinite range. Simply enough $\Pr(D_{\mathrm{T}} = d) = \lim_{D \to \infty}(\frac{1}{D}) \to 0$ and there is no chance of finding the target by any scheme.

Much of the previous analysis has been motivated by Alpern's work as the result of both private communication in the winter and spring of 1998 and through various publications such as [**1, 2, 4**]. Also the important consideration of Anderson [**5, 7, 6**] and the original works by Gal [**36**] and Schelling [**65**] should be noted.

Having developed our intuition now for planar search, we turn our attention to extending the theory that we have seen for searching the plane into a more general scheme that we

[3]The distribution of distances in some case, say a room, will be the distribution of distances from the agent to the points on the walls in the region. This is opposed to the simple concentric enclosure that this model assumes.

can actually use on a functional robotic system. This adaptation from theory to application is the topic of the next chapter.

CHAPTER 4

Efficient Search in Real Environments

We have seen in section 2.2 several examples of optimal search results for planar search with various sorts of *a priori* knowledge about the environment. The setting that will be of particular interest is for a mobile robot in a somewhat less contrived environment. Given the concern that physical motion is the most expensive operation for a mobile agent in terms of time we would like to design a system for efficient exploration.

The model for searching in an environment of concurrent rays will serve as a starting point for building a functional system to deal with the task of searching in practical environments. To map the interior of a building or perhaps set of roadways for a mobile agent is desirable functionality. From experience we see that rarely will such structured environments have more than 4 concurrent rays at an intersection. Despite that even having the ability to search 4 concurrent rays is insufficient. More commonly we will find that environments will not only have "concurrent ray" subproblems but that they will in addition have hierarchies of these subproblems. This issue of hierarchical concurrent paths will be dealt with in section 1.

The problem of search in an unknown planar environment has been addressed in the literature to some extent. The work of Anderson and Fakete [5, 7] refers to this class of problems as *lawn-mower search* problems. The idea being that to scour an environment with a sensor and overlap as little as possible the path that has already been traced. It is of no use to cover the same ground twice with a lawn-mower and hence is inefficient. Chapter 3 addresses a similar issue to the lawn-mower problem in an unconstrained environment with multiple agents assisting one another as a study of pure efficiency.

In constrained environments finding an optimal solution to the lawn-mower problem is a problem worthy of study. However a map is required of the environment before any calculation can begin. If the environment is unknown, then the issue is simply to explore the environment in an efficient manner. In order to extend the capabilities afforded by concurrent search, let us consider searching non-concurrent rays.

1. Searching Non-Concurrently Branched Environments

Demanding that we can examine concurrent rays is still too simplistic in practice. In the structured world that we would like to be able to cope with the techniques studied in chapter 2 are insufficient. Useful are the techniques that the theory provides for us, however adaptation and extension are going to be required for functional use in desired environments.

Concurrency of the search rays is a specification to simplify the theory, however we would like to relax this condition to be able to perform well in more robust environments. Figure 1 gives some indication of what sort of relaxation we are asking for. We will introduce the notion of *non-concurrent* rays.

DEFINITION 4.1 (Non-Concurrent Branches). *Given a tree $T(V, E)$ of vertices ($v \in V$) and edges ($e \in E$) between vertices and the tree being rooted at a vertex $r \; \exists r \in N$ called the root vertex, we say that T is a* branched tree *emanating from r .*

Our desire is to evolve a system based on what we know about the optimality conditions present during concurrent path searching and build them into an heuristic that can cope with this more robust setting.

One of the features that makes searching non-concurrent paths different from concurrent paths in a fundamental way is that there is no way of knowing the branching factor before beginning. The breadth of the search tree is unknown to start with. In order to not

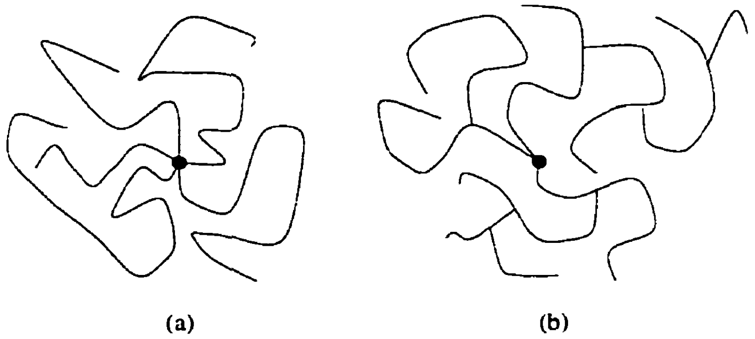


FIGURE 4.1. Examples of environments depicting concurrent [co-terminal] (a) and non-concurrent (b) branching. See definitions 2.1 and 4.1 respectively.

restrict the system to particular environments an online solution for this problem has been developed.

In section 2.4 of chapter 2 we find a description of how to search in concurrent rays in equation 16. Given that we know M, the number of separate rays, we know the distance to walk down the i-th ray is given by

$$f(i) = \left(\frac{M}{M-1}\right)^i \qquad \forall i \geq 1. \tag{4.1}$$

The case of non-concurrent rays poses the issue of discovering new rays during the search. The solution to cope with this is to allow M to change dynamically as the exploration progresses. M is therefore allowed to increase and decrease as information about the map is learned, as branches are detected and terminate, the search must continues on the rest of the graph (there is no longer a need to search again on the terminated ray). There is another way that M may decrease, in environments where intersection between the paths is allowed, in *cyclic* environments, M decreases as paths merge, as well as terminate. This case will be discussed later, for now we will assume that we are working on *acyclic* environments.

For searching in branching environments we keep track of the number of active [unterminated] branches, M, in the known environment and recalculate 22 as these updates M occur. This is the key modification that we need to extend the theory from searching concurrent rays to searching in branching environments. Using this method to deal with the expansion of the search, the expectation is that we are doing as well as possible with the knowledge we have.

Most importantly we are keeping with the underlying theme that we have seen developed in chapter 2. By recalculating the distances for the i−th turning point from equation 22 in a dynamic way, we maintain the spiral nature of the search. It's this underlying dependence on the spiral growth of the iterative deepening nature of the search algorithm that will provide the efficiency of this brand of search.

2. Trees of Non-Concurrent Search Paths

Distance will be measured as *path distance* from the origin to the current position. We will define *path distance* to be the distance traveled along the shortest known path yet traversed that joins the two points. In the case of acyclic environments there is no ambiguity in this value. In environments that contain cycles, it is important that we choose the *shortest* of the possible paths for uniqueness [see B.6].

In an tree structure it is sufficient to know only where the branches occur to have complete knowledge of the path structure. Branches are defined by the point at which they branch. The ability to reference the branch points together with knowledge about the preceding branch point is sufficient to allow deterministic navigation of the tree. Only very coarse sensoral information is necessary to be able to distinguish the possible courses available from a particular branch point.

The branch points in the tree will be referred to equally as reference points, since they serve to index the tree structure. Some notation often used for referring to tree structures are *parent branches* and *child branches*.

DEFINITION 4.2 (Tree Notation). *A* tree *is a graph in which every node has at most a single unique parent. This ensures that there is a well-defined route to each branch. Such a structure is known in graph theory as a* rooted DAG *(directed acyclic graph).*

A cyclic graph *is a graph $G(V, E)$ in which there is a path $\langle v_0, v_1, \ldots, v_k \rangle$ such that $v_0 = v_k$ and $v_0, v_1, \ldots, v_k$ are distinct* [21]. *In a directed graph this can mean that there are several paths possible to arrive at a given vertex. This has the general implication that there are multiple paths between two points in the graph.*

If q is a branch in a tree structure such that q is a branch off of a branch p (closer to the root) and r is a branch off of q (farther away from the root of the tree than q) then we say that the parent$(q) \subseteq p$, *and the* child$(q) \supseteq r$.

Navigation within cyclic environments is functionally more difficult with mobile robots than trees [rooted DAGs]. The trouble lies in determining whether or not, upon arriving at a new branching point, the same location has been previously discovered along a different route. This determination requires answering the problem "Have I been here before", the localization problem. Techniques in mobile robotics to perform this global position assertion depend on unreliable and limited sensors and are unstable. Localization techniques are contested in their own right, and although much progress has been made in this field [**63, 25, 26, 32, 53**] localization of sufficient resolution necessary to resolve path cycling issues can be difficult to attain.

For our purposes cycle detection is resolved by using odometric data together with "sonar signatures" of branch points. Sonar signatures are dense sonar samples taken at branch points and stored in order to try to identify that point uniquely at some later time. Instabilities are produced in performance on cyclic graphs by both failure to recognize arrival at previously visited location, and by false positive identification. False positive identification occurs when a new location is identified as being a previously visited one, wrongly. Either of these failures are very difficult to cope with, as they seriously effect the topology of the resulting map.

This perspective of our environment allows us to make the critical transition from graph theory to a robotic search algorithm. We induce a unique topology on a map with a given starting point according to where and when we detect branches in the environment. At any given branch point in the expanding search we have the beginning of two arms of the search. These are are topologically related to their parent arm with the root of the search tree being the origin of the search.

An isomorphism is thus defined between a robot searching in a map, and a topological tree. It makes sense for us to apply our algorithms on the space of this search tree, whether DFS, BFS or our spiral search approach. Due to the induced isomorphism, we have a means to translate the information back from the topological tree to the actual robot performing the search to optimize the progression of that search in a dynamic manner. We can exploit the algorithms that work on topological structures to guide our search through the real world by way of this isomorphism. The notions of breath- and depth-first search well known in graph theory take on practical real-world analogues.

3. Summary

This chapter details the necessary extensions from the optimal concepts revealed in chapter 2 to an efficient search algorithm defined over a relaxed domain. We need less restrictions on the type of domains over which we can apply the spiral search concepts. This will result in an algorithm that we can still quantitatively study, but yet is capable of performing in a real environment.

The means by which the extension from theory to application is done is by allowing for dynamic updates to the search parameters as information is gained about the environment. These dynamic updates are the key feature, but the search algorithm is a graph theoretic construct and so we need to interpret the real world as a graph. Hence the second important feature is the way in which we induce a topology on the search environment.

Given an branched environment, a starting point and a search algorithm there is a unique way to perceive the environment as a tree consisting of vertices and edges with the branches occurring in the order the search algorithm discovers them. This isomorphism between the search environment and it's underlying topology give us a means to apply the search algorithms by mapping the perceived environment to the dynamically updating tree, applying the search algorithm in order to determine the next step in the search and then mapping that decision back to the perceived environment as a decision in the progression of the search.

Now, having designed a method that should be applicable in real environments and we have predictions from the theory on the expected efficiency of that algorithm, we can go on to test the predictions. In the next chapter we will define the experiment that will be performed and some of the dependent and independent variables. Care has been taken to ensure that the experimentation will give correct results regarding the desired measurable quantities.

The perception of an isomorphism from the real environment of the robot to the topological structure of a rooted DAG allows a deterministic search through real environments. Chapters 5 and 6 will make thorough use of this relationship to match real-world behaviour to an algorithmically stable analogue.

CHAPTER 5

Experimental Method

Experiments to perform automatic exploration of unknown environments have been made to study the differences that various methods of search provide. Search is being performed using depth-first, breadth-first and our own spiral search algorithms. These searches are performed on a series of maps, some automatically generated, and some manually generated.

The robotic control technique is identical for the three different search algorithms, save for the steps explicitly related to choosing nodes within the underlying topological search tree. This fact ensures that the only differences in the performance of the search algorithms are explicitly due to the nature of the search algorithms and not some affect implicitly related to the systematic control of the robots.

Discussion of the robotic control system that may have any bearing on the dependent experimental variables follows in the subsequent sections. The issues related to the generation of the maps for the experimentation are addressed. The choice of the size of the maps, the relative size of the structures internal to the maps and details of the map generation are some of the critical issues. Identification of some of the dependencies of any robotic system will also be discussed. Mobile robotic systems in general have a certain complexity and some of the unique problems associated with mobile robots will be identified that may have an affect of biasing experimental method. This factor partly justifies the use of a simulator for the experimental trials in so much as it allows us to control the effect that real-world, noisy, error prone sensors may inflict on the experiment without treating these reasonably sophisticated problems outside the scope of this study. Also, details of how the exploration of the maps is performed are given. The methods for determining initial parameters for the various search algorithms are outlined to dispell any possible subjective bias in the experiments.

1. Map Generation

The search experiments to be done require a setting. There are several criteria that are desired from the maps that are to be used in the experimentation.

- Absolute Size — The maps must be large enough and contain sufficient structure to merit study.
- Relative Sizes — The overall size of the map with relation to the robot. The comparative sizes of structures in the map {hallway length, room size, corner angles, ...}.
- Structure — The design of the structures.
  - Manually versus Automatic — Manual maps induce bias on the design (purposefully or otherwise.) Automatic maps may induce some sort of implicit bias into the experiments.
  - Width of Hallways — Narrow hallways/doors may prevent entry of the robot to a large part of the map.
  - Branching Frequency — Determines in part the depth of the topological tree related to the map. Search algorithms are very sensitive to perceived topological changes.
  - Initial Pose — Serves as the root of the DAG. Small changes in starting position may result in unprecedented changes in the topological representation of a given map.

The maps used to test the search algorithms have to be large enough to provide sufficient variety and be of complex enough *topology*[1] to be able to tease apart the differences in the performance of the various search algorithms. Depth-first search and breadth-first search on a tree that is one level deep are identical. There is no point to even compare the algorithms on trivial trees.

The relative size of the map to the robot and of the map to its structures is important. Part of what is being studied is how the search algorithms perform according to time of execution of a certain task. With unit speed agents, time is linearly proportional to distance covered as long as the agents move continuously. So the relative distances in the map are important. The critical factors are: the scale of the map (relative to the size of the robot), the relative sizes of structures in the map (two maps with the same topology may induce different behaviours based on the relative sizes of the structures[2]) and of course, the actual topological structure of the map.

The manually generated maps have been designed with some bias in mind to try to illustrate pathological cases that may cause quantitative differences in the performance of the various search algorithms. Automatically generated maps are used to try to provide series of maps free from human bias to acquire more statistically valid results.

For the experimentation a large series of maps are used. Some of the smaller maps are hand drawn with purposeful intent to generate pathological behaviour in the algorithms. The larger maps have been created automatically in a semi-random fashion. The automatically generated maps have several key requisite properties. The topology of the connections in a map of the "hallways" from the perspective of the search algorithm is the important criteria. These "hallways" need not be much wider than the robot itself, an aspect of 3-5 times the diameter of the robot is sufficient for our purposes. The frequency that branches occur at are more critical and independent for this study of the way in which a "room" is actually searched. It is the large scale behaviour that we are interested in. The lengths of

[1]The *topology* of a map refers to the way that the structures in that map are connected to each other. Shape, size and distance are not our concern when we speak of the topology of a map, only the way in which the structures are interconnected. Tree diagrams are useful to convey topological information.
[2]There is a good example of this effect in section 6.4.

the hallways are also important as the underlying metric that is used in the spiral search algorithm is dependent on distance. Also the choice of the initial pose in a map affects the performance of the algorithms on that map. This effect is directly related to the topological layout that the robotic searcher perceives. With these concerns in mind the automatic generation of maps can be described.

The map generation is performed in the following manner. A regular square grid is created with the total size of the grid and the resolution of the grid pattern being parameters. These serve to address the issues of scale and the "radius" properties of the hallways. In order to create various maps from these similar initial conditions, the grid intersection points are then randomly[3] and independently displaced from their grid coordinate. A Delaunay triangulation is then performed on the displaced grid points creating a series of planar triangulations that differ from each other, maintaining still some sense of scale and radius. From one corner of the triangulated environment edges are recursively removed to create "runs" through the triangulation. From the interior of these "runs" edges are progressively removed in a pseudo-random manner to create progressions of branching "hallways" with the desired properties. Care is taken during this thinning not to join separate hallways. This has the effect of not creating cycles in the graph, although this is not a requirement for the search algorithms it is a requirement to ensure stable map generation. Boundary edges are not removed during this thinning since the condition of bounded maps is a requirement, especially for depth-first search, to ensure termination of the experiments.

The resulting maps from edge thinning of the Delaunay triangulation of irregular grids have roughly the desired requisite properties. This method facilitates the generation of a test bed of experimental environments with the appropriate properties to differentiate the performance of the various search algorithms.

[3]Using a pseudo-random number generator. Which, of course, is not purely random, but serves our purposes.

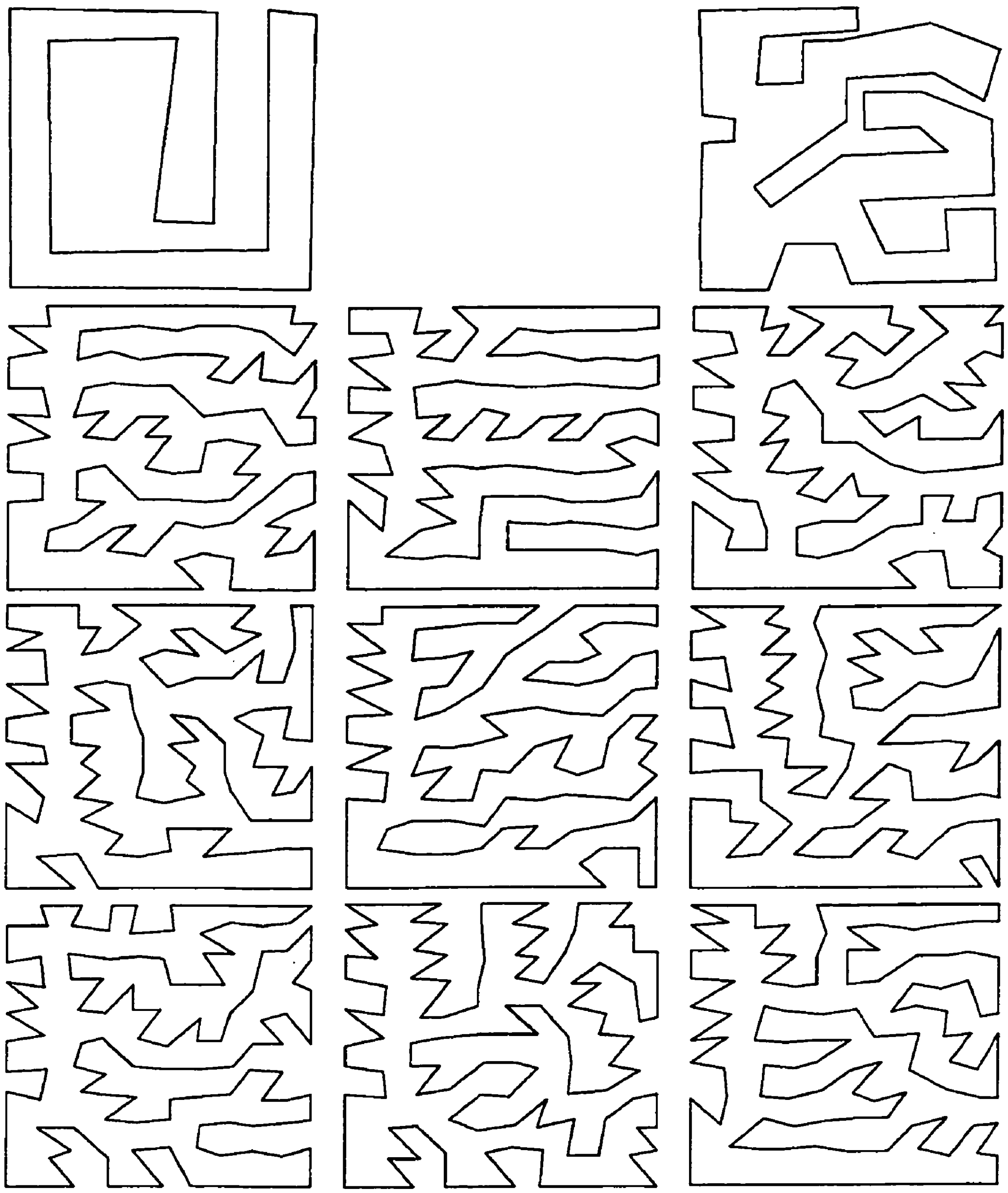


FIGURE 5.1. These are half of the Maps used. The top two are the manually generated maps. The simulated length and width of these maps would be roughly 20 meters each.

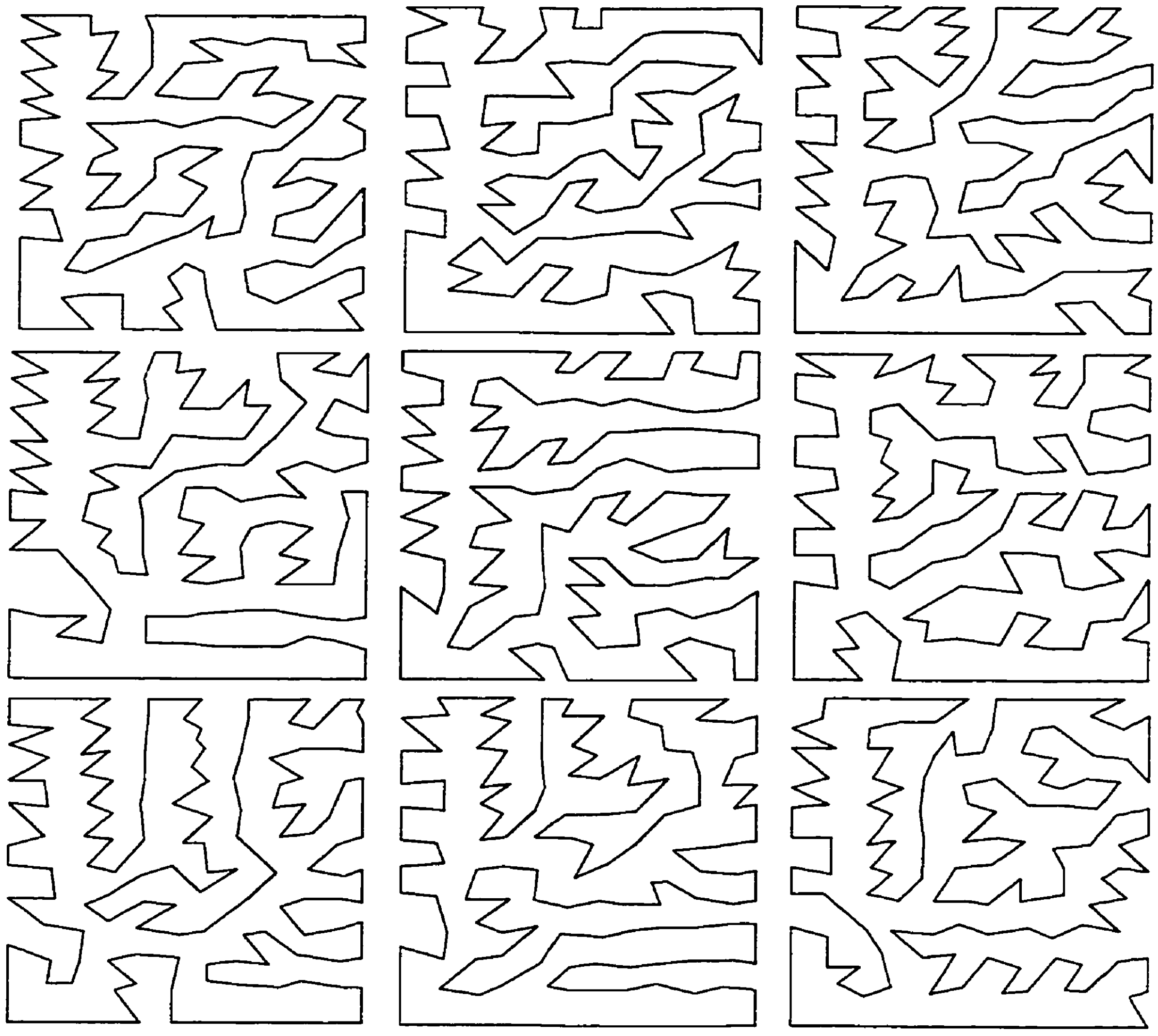


FIGURE 5.2. These are the rest of the randomly generated maps. These maps are larger than the previous ones being roughly 30 meters a side. The robot was started arbitrarily in the bottom left hand conmer of the maps.

2. Experimental Dependencies

The field of mobile robotics faces a host of difficult problems that need to be addressed in any functional robotic system. It is not our intention to solve each of these problems using the most sophisticated methods available[4]. Critically addressing each of the problems in turn would both cause intractability of the system and would distract attention from the problem at hand; the performance of search algorithms in robust environments.

Some of the well known problems in mobile robotics that are tractable, but difficult in their own right, that have direct affect on this set of experiments and have had to be addressed are as follows. Odometric drift and the accumulation of error between perceived and real movement and how to compensate for that error is an ongoing research problem in the field [**59, 60**]. The creation of internal representations of the environment for future reference are critically dependent on the details of the sensor used, and the representation of that data. Attempts to be able to reuse this acquired information require some flexibility in interpretation of current data relative to existing knowledge in order to be useful. In particular, noise and error in odometric readings serve to create divergences between perceived and real position for a mobile agent. Also reliability and repeatability is a major concern for any kind of active sensor. Work on these and related problems is common in the field [**24, 23, 66, 54, 47**]. Again noise, error in the sensor itself and the action of sensing the environment serve to create divergences between real and perceived images. Low resolution sonar is very susceptible to many distorting effects, some are well understood and some not. Despite these shortcomings, sonar has been judged to be a sufficient guidance mechanism for the purposes that we require. Such factors are anathema to the progress of the field of robotics. Particular in this work the development of robust systems that are able to function on real robots in real environments is equally inhibiting.

Dead reckoning from raw odometric data is a very easy method with which to attempt robotic navigation. In the interest of the problem at hand, some dependence on such information is being used. Efforts have been made to absolutely minimize the dependence of

[4]For a brief discussion of how the various techniques of mobile robotics blend together and constructively and destructively interfere in *robotics research integration* see [**45**].

the algorithm itself on odometric information. In fact at no time during the exploration of unknown territory or within any of the search algorithms themselves is odometric information used. The use of odometric estimation is used in the framework to facilitate motion of the robot through the already explored, known environment. In this sense, the robotic framework actually builds an odometric representation for the connectivity of the known environment. It is referred to only for navigation between reference points in the known map. Any method that can achieve this task can easily replace the use of odometric dead reckoning in the experimental framework without affecting the manner in which search is performed or without affecting the search algorithms themselves in any way other than utility.

The reliance of the experimental system upon unstable and unpredictable elements inherent in mobile robotics is minimal. Some amount of functionality is required to perform the desired tasks and so the use of odometric data and sonar dependence is justified. Careful thought has been given to avoid overt use of methods which are not easily upgradable and yet prone to failure. These dependencies are flexibly bounded in terms of being able to perform in the presence of error prone sensors and most importantly, these reliances are modular enough to be easily replaced by different methods should the need arise. At present they are sufficient for completion of the task at hand.

3. The Exploration Experiment

The experiments have been performed in simulation on a robotic system simulator. The simulation system is fully capable of locally controlling a real robot with sensors. The scaling of the experiment from simulation to an actual robot would be a trivial matter aside from the difficulties discussed at the beginning of section 2. The simulation environment allows for control over some non-trivial aspects of mobile robotics that are beyond the scope of this study such as: sonar sensor interpretation and dynamic odometric correction.

The experiments begin with the robot coming online with the intention of exploring the environment in which it finds itself. The robot begins with no information about the

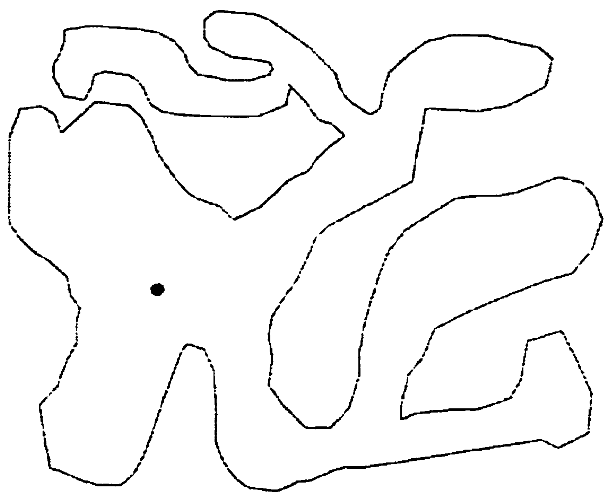


FIGURE 5.3. A possible scenario. The robot, indicated by the central dot, upon waking knows nothing of the environment.

environment. Figure 3 will begin an illustration of the search progression. The initialization the robot performs consists of taking a set of sonar samples and deciding on a "wall" to move appropriately close to. The robot employs a wall-following technique to explore the environment. The distance that the robot finds itself from the wall that it approaches initially is used to seed the robots decision of what a "unit" will be in the current metric. This method of choosing a dynamically allocated metric unit is more appropriate than using some arbitrary fixed metric in hopes that the initial surroundings of the robot may contain some sort of global scale information that could potentially aid the search. If the robot finds itself on a soccer playing field or in a partitioned office setting it is appropriate to let the environment suggest some initial metric unit. The choice of step size has low-order effect on the long term behaviour of the search algorithm although it does affect the performance of any particular trial.

Based on the assignment of the metric unit, the robot then performs a circumnavigation of the position that it has moved from. This circumnavigation is performed around a radius of twice the unit metric chosen. The purpose of this exercise is to try and establish an effective direction for the beginning of the exploration. This initialization is often enough to determine that there are some or several directions to begin the exploration. Topologically

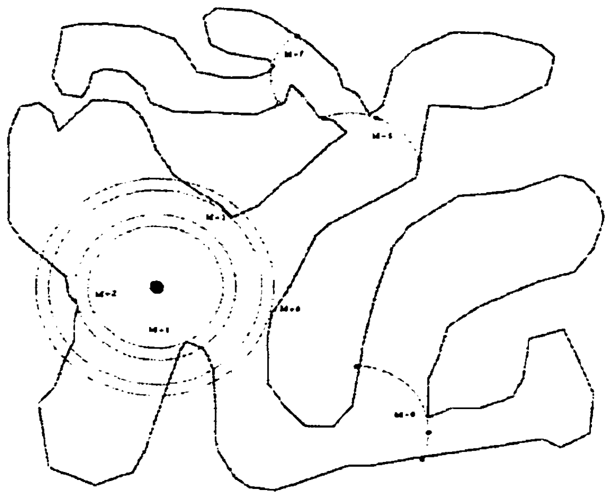


FIGURE 5.4. The concentric rings indicate the bounds of the explored area when the number of concurrent rays is increased. This occurs when an obstacle is encountered at a point in the middle of the exploration frontier within an area that was presumed to be a single ray, *ie* at a ray bifurcation.

we have identified the root of the tree that will be used to guide our search algorithms and perhaps may have acquired some information about the base of the search tree. Success at this point to determine directions for the exploration is not necessary, but tends to be the case especially considering that the initialization radius is dependent upon some rough equivalence to the size of the room. In figure 4 the inner most enclosing circle would be the seed to the determination of the unit metric distance. The circumnavigation after initialization may have the robot map a distinct portion of the initial surroundings. In the case of figure 4 the robot may perceive an exploration with four separate concurrent search branches after this initialization stage.

At this point the exploration begins. The exploration is based on a wall following method that uses the following logic. The number of "concurrent search rays" at this point is zero of greater. Until there are at least two branches to search there are no decisions to make. The number of branches at any given time is dynamic and completely dependent on the exact nature of the map, the search algorithm and the starting point of the experiment. Figure 4 serves to illustrate this point further, as the search progresses beyond initialization and down the independent branches some of the branches terminate, and some are seen to branch further. This impacts the distance down a branch that we will search according to the theoretical optimal search ratio laid out in section 16 of $f(i) = \left(\frac{m}{m-1}\right)^i$.

We are assuming that we are in an environment that has some sort of assignable topological structure. Given this assumption breadth and depth first search are already satisfiable. Spiral search, as is indicated in equation 16, requires the further assumptions to hold: That there is a branching structure that will drive the number of concurrent search branches to greater than two[5] and that we have some method whereby to determine the distance that we have traveled along a path from the origin. This second requirement is not tied necessarily to metric distance *per se* and any Lyapunov potential function [see definition: B.8] can serve this purpose equally well.

As we progress through the environment, we see where the increments take place in figure 4. These are labelled in the diagram for clarity albeit incorrectly. They are incorrect in reality since *decrements* also occur when rays of the search have been exhausted[6]. There is another issue present here that will be addressed and that is these rays are not in fact concurrent. The rays do not emanate from a common point.

This outlines an algorithm for extending the frontier of our search through a real environment. In the circumstance of concurrent rays the algorithm is designed to take advantage of what is known to be optimal behaviour in more restricted environments where quantitative evaluation is possible. The critical perception of an equivalence between the real world and the theoretical graph construct shown in chapter 4 allows for the adoption of an efficient approach to planar search in this more qualitative setting of real world structure. We have a method to adapt it to the non-concurrent situation in a manner made yet more explicit in chapter 6 section 3, in order to guide us in expanding our knowledge of the environment in a locality-centric manner.

[5]This excludes the degenerate case; the search will still proceed in such an environment, but there will be no "spiral" nature to the search.

[6]For instance the lower left region would certainly have been completely explored before the $M = 5$ vertex would be discovered and hence would have resulted in removal from the list of open rays to explore along with a decrement in the number of rays. So that by the time the $M = 5$ vertex is discovered there will actually be at most 3 active search rays, however this is a pedantic issue.

4. Exploration of Unknown Territory

When searching unexplored territory an alternating method of wall-following is used. By following the left wall for some distance then moving to the right wall and following it for a some distance, the search progresses into unknown territory in a reliable and safe way, while still being able to detect branches in the hallway. Only local sonar scans are used for this purpose and metric information is only utilized over a short range to minimize the accumulation of odometric error.

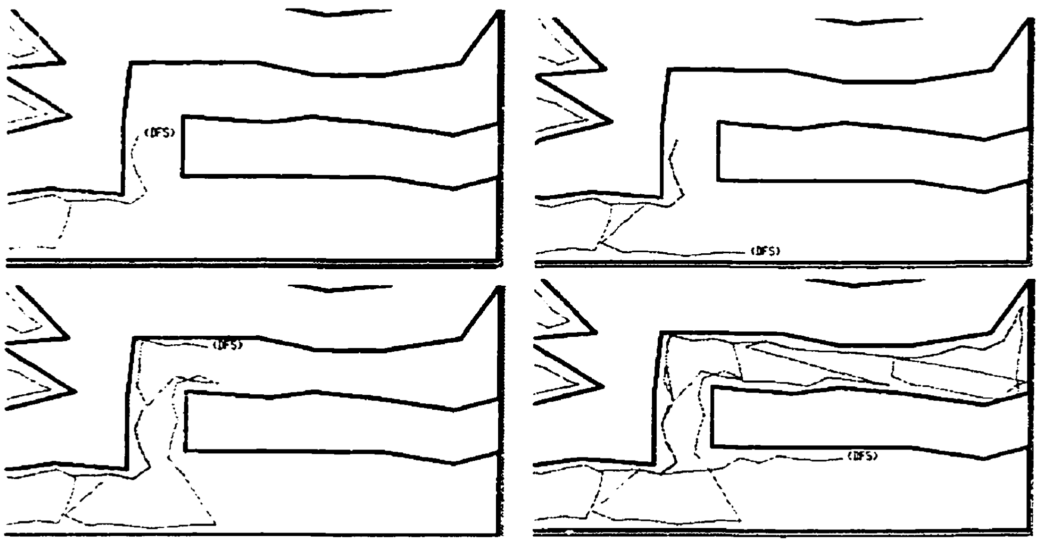


FIGURE 5.5. Exploration of unknown territory. By extending left and right searches along the walls, branches are easily detected while crossing. Detection occurs between the second and third plates.

A progression of exploration down a branch is shown in figure 5. The robot has only sonar range information about the location of the walls. By following the left and right walls alternately, exploration progresses. Between the first and second plates of figure 5 the robot has switched from the left to the right side for wall-following. Between the second and third plates a new branch in the search has been detected. This series features a depth-first searcher, hence the search continues on deeper at every opportunity. The fourth plate shows the completely searched branch and the robot resuming search in a depth first manner into unexplored territory.

Continuation of the exploration cycle ensure complete coverage of the map. With depth-first search, the robot moves through explored previously explored territory only when necessary; when termination of a branch forces traversal to a parent node. With breadth-first search the robot never searches into unexplored territory until all nodes at the current search depth have been explored. With spiral search, the decision of when to switch between branches in the topological search tree representing is a little more sophisticated. In the next chapter results on the relative performance of the three search algorithms will be presented and the advantages of spiral search will become clear.

5. Summary

Efforts have been taken to perform a set of experiments that will show the qualitative differences between a variety of search algorithms for automatic robotic exploration. To insure the accuracy of these experiments many of the dependent factors have been isolated. However, some factors are beyond easy control. Such factors that may affect the experiment have been identified and discussed.

The generation of the series of maps over which the experiments take place has been explained. The map generation has been done in a way to hopefully qualitatively separate the performance qualities of the different algorithms in an unbiased and statistically significant manner. Issues of map size, structure and topological representation have been carefully considered.

The modeling of sonar data and the accurate use of odometric information for large scale maneuvering within a real environment are difficult unreliable tasks. It is not the purpose of this thesis to address these issues in a rigorous way and the use of the simulator to control only those aspects which are otherwise not part of the dependent experimentation.

The maps that are loaded into the simulator have been "randomly" generated to have certain properties while not providing any heuristic advantages to the system. Some of the desired properties are the scale relative to the robot, the underlying topology of the map, the relative sizes and distances within the map of its constituent structures and that there is an "inside" to the map, that it is bounded for termination of each of the search algorithms.

Experimental methodology has been carefully explained. Both in the important sense of how the global search progresses in unknown environments and in the sense of how local exploration is performed within a branch of the search tree.

Having been assured of the validity of the experimental underpinnings we can progress to the results of the actual experimentation. The next chapter will reveal our findings and verify our initial presumptions of the efficacy of the spiral search technique on a real robotic system. These results have been predicted by the theory on planar search in highly constrained theoretical settings and are shown to scale to good worst-case behaviour in a real robotic system.

CHAPTER 6

Experimental Verification

Analysis of the performance of the search methods that have been the focus of this work will be shown here. The more traditional methods of search are contrasted against the performance of search being performed using an implementation of the *spiral search* technique that has been described. The setting for the experiments consists of a series of manually and randomly generated *floor map* models consisting of interconnected *hallways*[1].

1. Data Presentation

The analysis is presented using two types of measure for each map: The first graph for each map will indicate the *time of discovery* for a large number of points in each map. This will be referred to as the *point discovery* series. We assume unit speed agents in each of the experiments, so the time that a point is discovered is linearly proportional to the total distance traveled by the robot when that point is first visited.

To make sensible comparisons, the time of discovery plots order the points according to depth-first discovery. DFS search in these experiments tended to have the lowest average discovery time per point. The points chosen for the plots are points common to each of the different search algorithms corresponding to a monotonic depth-first ordering.

The second series of plots show the time taken to discover *all points at a given distance* from the starting point. This series will be referred to as the *all-points time discovery* series.

[1]These definitions are loosely used only to suggest that the search occurs in environments rich enough to encapsulate what are often referred to as *office-like* environments in mobile robotics literature.

The time depicted is the time taken to discover the *last* point that is at most a given distance from the origin of search. This number is generated by taking the set of all discovered points in a map together with the time that each is discovered and then dividing the points into *bins*. These bins are relative to the size of the individual maps. We have arbitrarily chosen to subdivide the maps into 10 bins each. Thus each bin contains points that lie within 10 percentile ranges of the longest path in the map. None of the maps used had a topological representation with a tree depth of more than 10 so this arbitrary choice of segmentation is of sufficient resolution to capture any qualitative differences in the algorithms.

For the point discovery series we know from the theory that in acyclic bounded graphs depth-first search will minimize the total distance [or time] for the graph. This is certainly expected and clearly holds true in the experiments as the mean discovery time for the DFS algorithm is seen to be consistently lowest.

For the all-points discovery series the results will be discussed on a map by map basis. This measure is interesting as it indicates the worst-case time taken to discover any point in the map given its distance from the search origin. It is this worst-case behavior that this thesis was intended to address. Spiral search should provide good worst-case results for discovering points a given distance from the origin. Using the analogy of searching for a lost set of keys without being certain of the distance they lie from where they were assumed to be, then spiral search should provide a least cost approach to the search. The results from these sets of experiments reinforce our belief that spiral search achieves this primary objective. It is worth noting that the experiments show the all-points discovery time taken over a global maximum of search distance. This is important and perhaps counter-intuitive, even though spiral search takes significantly longer to find some points, the worst-case time of discovery is significantly lower than depth or breadth first search on a large category of interesting maps.

2. Underlying Observations

The nature of the "logarithmic spiral" search pattern should not be misunderstood. The spiral nature of search is not readily evident during the actual search. Nothing in the search itself seems to obviously display any spiral properties. The only use of spiral curves is in choosing the points during which the search is advanced to search other rays of the already known map.

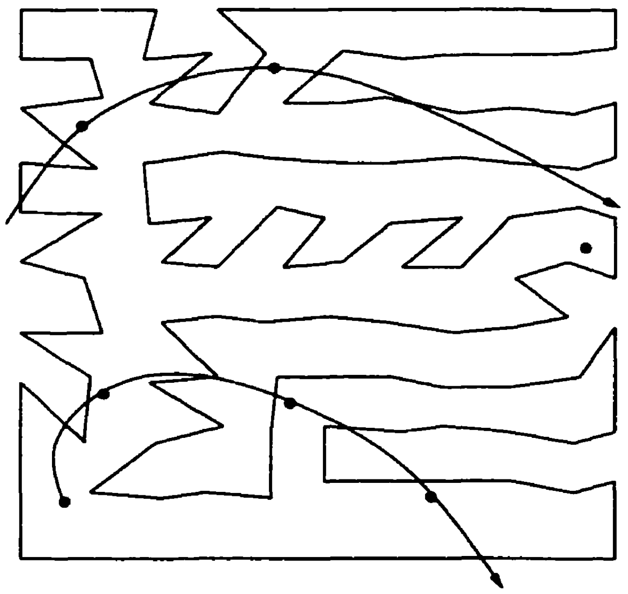


FIGURE 6.1. The nature of the underlying spiral guiding the search progression. The dots represent the turning points chosen while performing spiral search on this map.

Figure 1 shows a typical map with the turning points of spiral search depicted by dots and joined in order of occurrence. During the search, these points signify places where the search algorithm signalled the robot to abandon search and continue down another node in the underlying search tree. By joining the points it is hoped that some feel for how the spiral nature of the search development actually relates to the physical map.

Turning points are selected by their distance from the origin of the search. That distance is measured within the map, as the shortest path from the origin that the robot has taken to its current position. The number of branches that the robot will encounter during its search changes dynamically as the search progresses based on the robots current perception of the known map. The unit step distance is initialized to be roughly the width of the hallway/room in which the robot finds itself to begin. To recall from equation 16 both the unit step distance and the number of branches factor into the calculation of the next search distance.

The choice of the metric is independent from the the large scale behaviour of spiral search. Hence, use of some other metric may admit a different evaluation of *cost* during the search while maintaining the benefits gained from a spiral search approach. Any potential field may equally serve to drive the selection of the turning points. This is due to the unique fact that the family of logarithmic spirals are invariant under rotation and dilatation in the plane. As long as the function chosen to drive the search is a *Lyapunov*[2] function, then we are guaranteed that spiral search will behave with the worst case behavior that we seek. This makes spiral search a candidate for use in various other domains than just planar search. Search in higher dimensions [i.e. underwater or outer space settings] or search driven by different potential fields. Robotic application areas where this may be useful could range from volcanic exploration, where a temperature gradient potential field might be seen to be important, or nuclear reactor exploration where there may exist radiation gradients to drive the search as a potential field.

3. The Test Bed

A battery of experiments have been performed using variations in the search methods to determine the performance of these algorithms relative to each other in actual settings. Many different maps have been used in the generation of the data. For each experiment the map and the starting pose of the robot was identical. The different algorithms tested are variations to a depth-first search, a breadth-first search and spiral-search approaches.

Several examples each of a series of iterations of rounds of the experimental procedure follow. These figures show stepwise progression by one of the algorithms in question in an experiment on a hand drawn, pre-constructed map [seen in figure 2] representing perhaps a set of hallways in a building. Whereas one of our test robots has a diameter of

[2]see B.8 for a definition

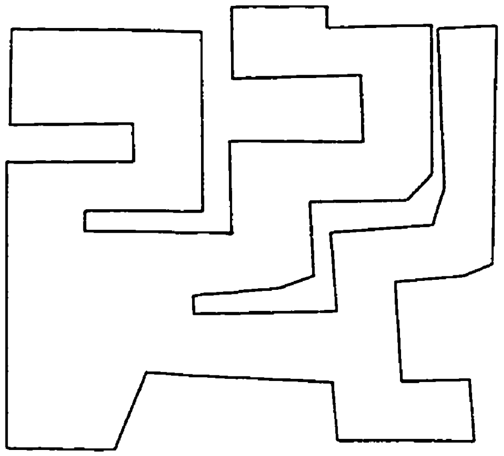


FIGURE 6.2. One of the simplest maps used for simulation testing.

50 centimeters, the relative size of the map would be roughly 25 meters. This is the rough scale of the environment to the robot used for most of the experiments.

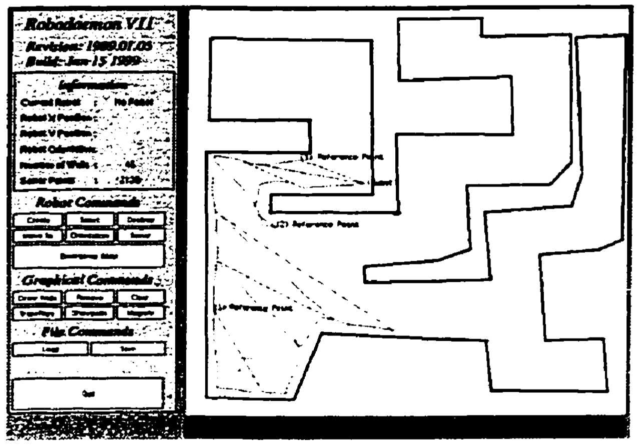


FIGURE 6.3. The robot at the end of the first iteration of search, $f(1)$. Search depth is not logged until the robot determines that there is more than one possible branch to explore, i.e. $M > 1$ [M from equation 16].

Figures 3 - 5 indicate the progress of the simulated robot at the end of each successive spiral iteration. The extent of this map is not so large that a second cycle results in complete traversal of the environment. This series is intended to indicate the nature of the progression of the algorithm in simulation. The algorithms run similarly on all of the maps.

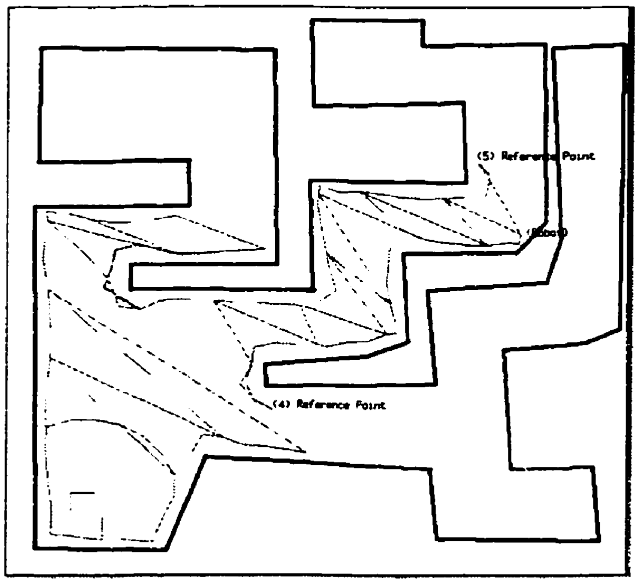


FIGURE 6.4. The robot depicted at the end of the second iteration, $f(2)$. The search will continue down the next unexplored branch in order to maintain spiral growth in explored area in a branch-wise sense.

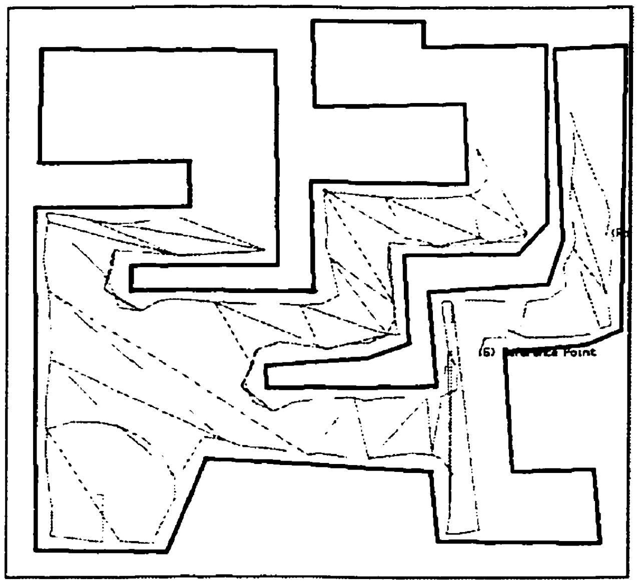


FIGURE 6.5. The explored area in the map after the third iteration, $f(3)$ steps down the third ray. $f(4)$ will prove large enough to finish exploring each of the branches completely before another spiral iteration is required.

The spiral nature of the search algorithm is again suggested by the progression of the robot down each of the branches of the map. This is gives some perspective as to how figure 1 was generated.

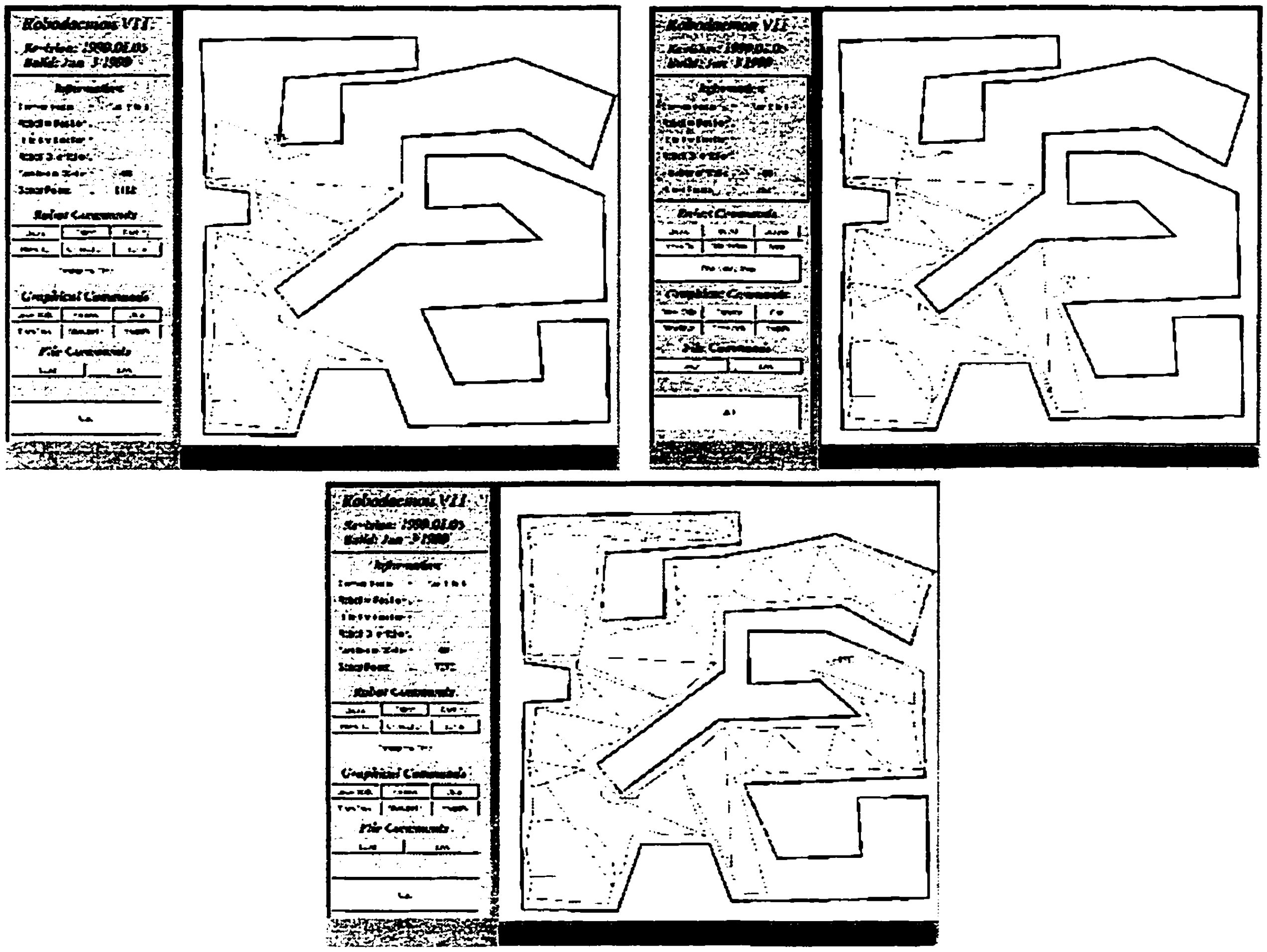


FIGURE 6.6. This series illustrates a run by the breadth first search progression.

A series of steps along the breadth-first search of a different map shown in figure 6 contrasts that of figures 3 - 5. The robot is seen to perform a branch switch every time there is a new level of branch discovered. Again this is based on the interpretation that branches in the real world correspond to vertices in the topological representation mentioned in chapter 4. BFS still corresponds to the traditional use from graph theory, where all cousin vertices are visited before any child vertices by this correspondence. This kind of behaviour will be seen to fail to be very useful in a performance sense. An intuition for why the performance in this case may be developed by gained by perceiving that the number of branch switches grows in linear proportion to the number of edges in the topological representation for a map. This is a scale independent measure. Recognizing that breadth first search does not scale with the map should give us some intuition for why the constant factor overhead of branch switching will result in unimpressive search performance.

The experiments are run on a variety of hand drawn and randomly generated maps. The results from each map are compiled and offer a large enough range and variety in findings that any qualitative performance observations that are made will accord closely to real performance of the algorithm.

4. The Experimental Results

Experimental trials verify the original postulate that spiral search has better worst-case search performance than either depth- or breadth-first search. Spiral search does not perform best in every case. Dependency on the robots starting point in a given map and the underlying topology of the map is clearly factor into overall performance. Interesting combinations of these factors directly from the experimental results will be addressed in the discussion of the experimental findings.

Both *time of discovery* and *all-points time discovery* plots will be shown for all of the experiments discussed. To reiterate, the conclusions that we expect to discover from the theory are that in bounded acyclic maps, depth-first search should explore the maps with minimum re-traversal of discovered area, resulting in a low average discovery time on a pose-per-pose basis. The second result that the theory should predict is that spiral search will perform better while attempting to discover all points that lie within a given radius of the starting point. To see how these two expectations fare under actual experimentation on a robust robotic system results are presented in our *time of discovery* and *all-points time discovery* plots to respectively uncover the desired trends.

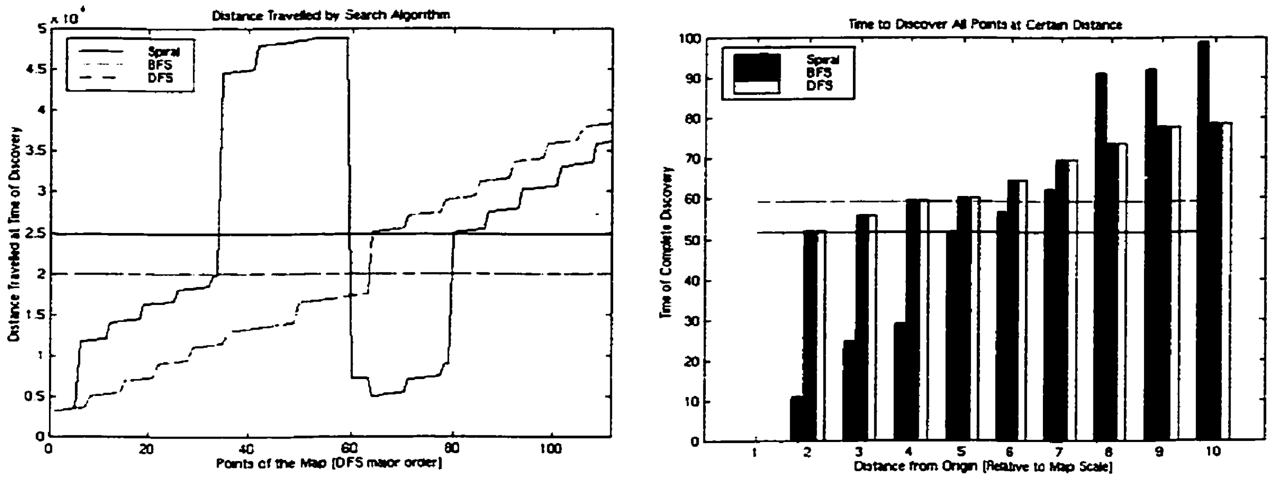


FIGURE 6.7. These experimental results from an unbranched "hallway" type map. These results would suggest an approach to looking for a light switch in a darkened hallway.

The first experimental results, shown in figure 7, are for a map of pathological simplicity. This map was manually generated and serves as a trivial case for study. The map is best described as a single, long hallway, with no branching features. The robot starts searching somewhere in the middle of the hallway and proceeds to search its length. The experiment with the robot starting at an end point of the hallway shows identical performance for the three algorithms. This case shows a minimal map and starting point configuration to differentiate between spiral and either DFS or BFS, both of the latter perform identically since there are no bifurcations in the map to separate their performance. The hallway is long enough to separate the performance of spiral search from the other algorithms, nonetheless.

This experiment is akin to looking for a light switch in a darkened hallway, given no information on the starting point of the search nor on the location or distance to the light switch. This experiment closely approximates the theoretical discussion in chapter 2 of section 2.2.5 and searching for a *point in a line* of unknown dimensions.

From the perspective of the *time of discovery per point* in the leftmost plot in figure 7 there is no difference in the performance of the DFS or BFS approaches. Nor is there any overhead in either of the algorithms of traversing already discovered area in the map

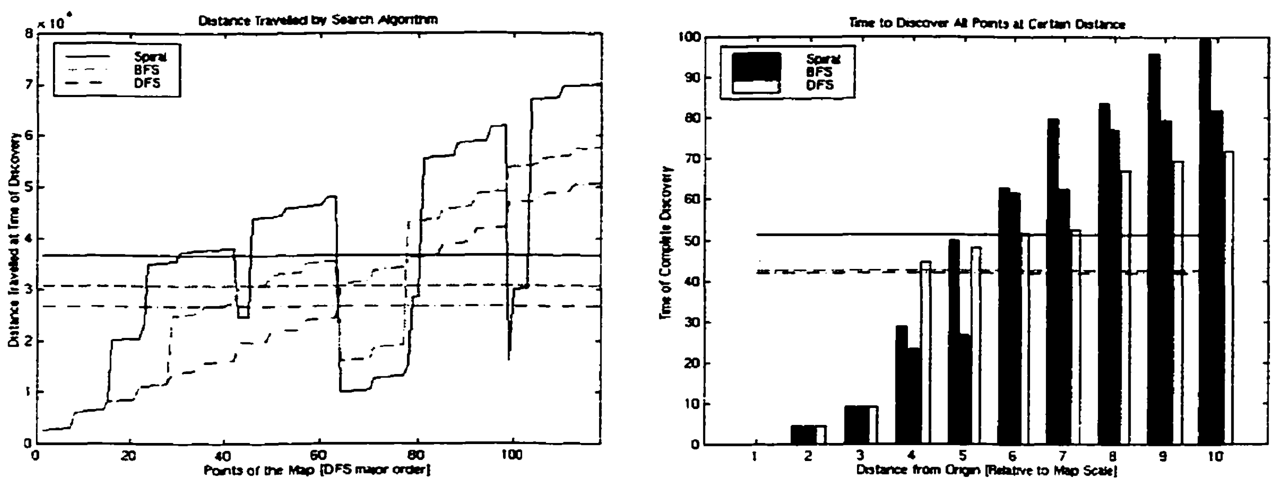


FIGURE 6.8. The graph on the left shows the time that [common] points are discovered based on algorithm. The Bar graph depicts the time at which the *last point at a given distance* is discovered based on the various search algorithms.

beyond what is essential. The alternations caused by the spiral search approach induce some such overhead in the progression of the search, resulting in a slightly higher time of discovery for roughly half of the points in the map. It is evident that the scale of the map caused the spiral search algorithm to alternate once during the search between branches, this should be obvious from the *time of discovery* plot.

As the theory suggests, however, a different truth is borne out in the *all-points discovery* plot on the right of figure 7. Indeed we see that alternating between branches results in a 12.5% lower average *all-points discovery* average in this particular case on the behalf of spiral search. This is encouraging and suggestive that DFS or BFS are clearly not capable of delivering good worst-case behaviour even in some simple situations. The difference between the performance of spiral search and DFS would continue to grow in favour of spiral search in experiments with yet greater scale. This experiment illustrates the advantages of modified spiral search but since it represents a test case closely approximating an early theoretic result it is an important experiment to test the validity of the theory.

The experiment with results shown in figure 8 was run on a small, manually generated map. The topology of the map is a complete balanced tree with depth two. Each of the hallways joining the nodes are roughly the same length. The intent behind the design of this map was to provide a minimal map such that each of the algorithms would perform exploration of the map in a manner unique to itself. Any map with less nodes in the topological representation would fail to uniquely differentiate the various search algorithms' properties.

The *time of discovery* plot on the left in figure 8 shows mean performance of DFS to be the lowest, followed by the middling performance of BFS and having spiral search with the highest of the mean performance values. These findings essentially represent the redundancy in the exploration for the different algorithms. BFS spends some time traveling through previously explored territory, while spiral search alternates even more often thus spending more time traversing already explored territory than BFS and amassing a greater overhead of travel than either of the other algorithms to perform a complete exploration.

All-points time discovery performance displayed in figure 8 on the right, bears a similar bias towards the performance of both DFS and BFS. Spiral search in this case has the highest mean time for *all-points discovery* while DFS and BFS perform nearly identically in this right. The brutal symmetry of the underlying topology and the simplicity of the map itself make either BFS or DFS equally well suited to this task at hand. Contributing to the high overhead that spiral search endures here is the fact that the map is very small.

The topological tree representing this map is only of depth two. The map is simply not large enough to allow the dividends of the spiral search approach to overcome the overhead paid in the early stages of search. This experiment is biased against worst-case performance, the symmetry of the map and its small size make it closely approximate a best-case scenario for both DFS and BFS. This is the exact reason behind the use of this map in the experimental battery and the justification for it as a manually generated test case.

The results in figure 9 come from an experiment run on an automatically generated map. These results are fairly typical of the findings for the automatically generated maps. The size of the map is considerably larger than that of the previous manually generated

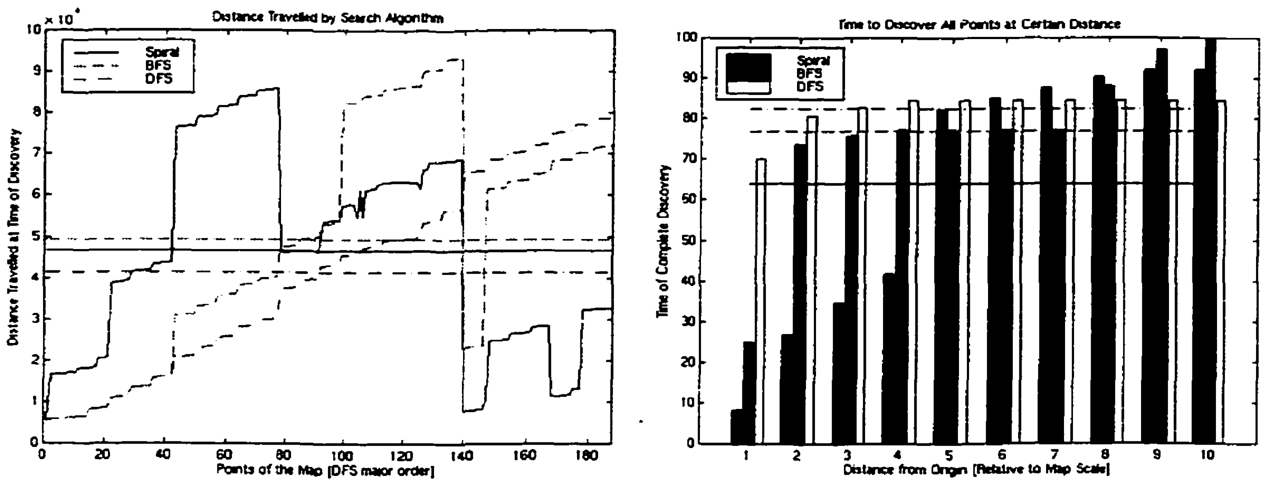
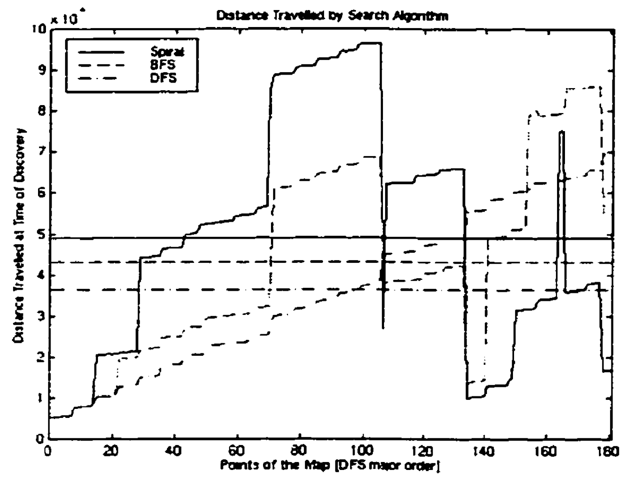


FIGURE 6.9. The first of the randomly generated maps. The area of this map is considerably greater than the manually generated maps. More complex behaviour is seen in the *time of discovery* graph on the left.

maps. This fact alone allows for more alternations for the spiral and BFS algorithms. This fact can be seen by the number of fluctuations in the left of the figure by the spiral search and BFS algorithms relative to DFS.

One interesting fact about this experiment is that spiral search actually has a lower average time of discovery than BFS. The previous experiments took place on smaller maps and so the overhead paid in redundant travel was relatively high for spiral search. In this experiment we see that as the maps get larger in size spiral search has a diminishing marginal return on redundant travel relative to BFS. DFS still has lowest time of discovery, but the margins between spiral search and DFS are closing too, although this is an asymptotic relationship.

The *all-points time discovery* plot on the right of figure 9 shows that spiral search handily outperforms both BFS and DFS having performance times of 83% and 77% respectively. DFS here exhibits a very typical failure mode. It achieves the lowest all-points discovery time because of a asymmetry in the balance of the topological tree. DFS spends a considerable amount of time searching points far from where it begins, while there are



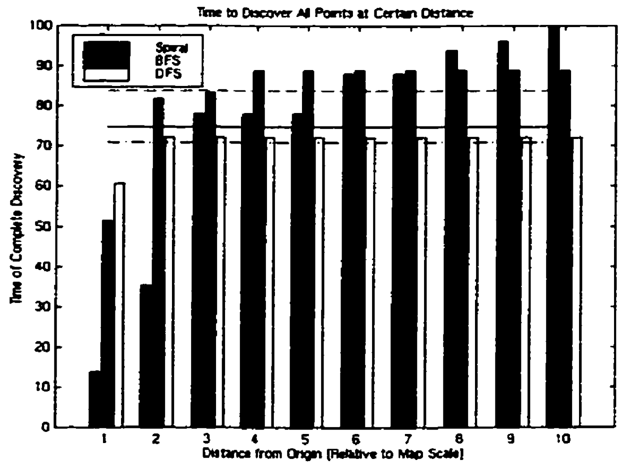


FIGURE 6.10. This is an interesting case where DFS has better mean performance time than Spiral Search. Although the results for are very close in terms of mean performance DFS prevails. This result is linked to the the map topology and initial pose used in the experiment

points very nearby the origin that remain unsearched. This contradicts the principle of locality . This fact will be further discussed in the experiment relating to figure 12 where the effect is more obvious.

Figure 10 summarizes an experiment run on another of the randomly generated maps. We see that in some cases spiral search does not perform the best in terms of *all-points discovery*. This experiment finishes with DFS having an all-points discovery time of 94% of that of spiral search and 84% that of BFS performance. The cause of this is that there is an asymmetry in the topological tree of the map that favours DFS.

The issue is particularly related to the asymmetry of the topology of the map and the relative lengths of the hallways therein. From the starting point in the bottom left corner of the map, shown on the left in figure 11, the bulk of the map is laid out in such a manner that the clockwise depth-first searcher is able to discover the large majority of the area of the map before descending into the relatively short branch from the bottom of the map.

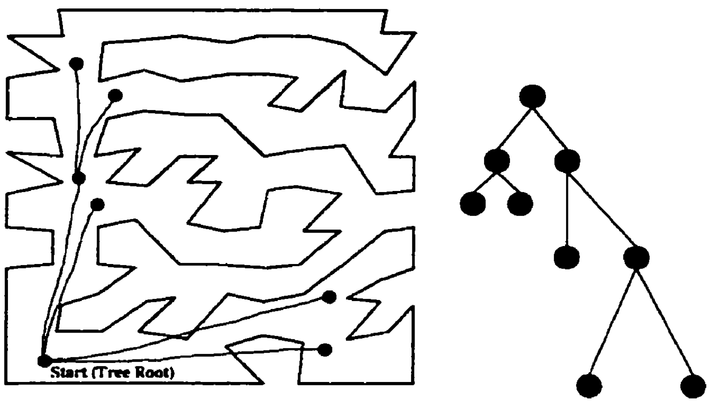


FIGURE 6.11. The topological representation on the right illustrates the map structure with edge weighting roughly corresponding to the lengths of the edges in the map on the left. This correspondence of physical environment to topological representation is nicely illustrated here and is crucial to the development of a real-world system from the theoretical primitives examined: Robots operate in the real world and algorithms operate on data. Here there is a well defined relationship between the two that can be exploited. There is a strong bias in this particular configuration that favours [clockwise] depth first search.

The topological layout as perceived from the initial pose in the bottom left hand corner is approximated on the right in figure 11. Some attempt was made to convey a sense of the relative weights of the branches according to length in this representation. The correspondence between the physical world that the robot operates in and the topological representation that allows quantitative study of search algorithms is well illustrated in the figure. Breadth-first search, for instance, is clearly applicable to a topological graph and the correspondence shown here allows a practical interpretation for the less obvious use of breadth-first search in real-world search.

This experiment also suggests the sense of dependence that the experiments have on the choice of the initial pose. In this example a decision to start somewhere else on the map would have a strong influence on the performance results. Beginning from a different location will have the effect of rotating the perceived topological tree for the map to have a different root. This would change the perceived asymmetry of the tree and of the distribution of explorable area in the map would be different. These factors all have an affect on the relative performance of the search algorithms. Although this experiment finds a lower mean *all-points discovery* time for DFS the margin of improvement is quite small, at 6%, for the gain over spiral search.

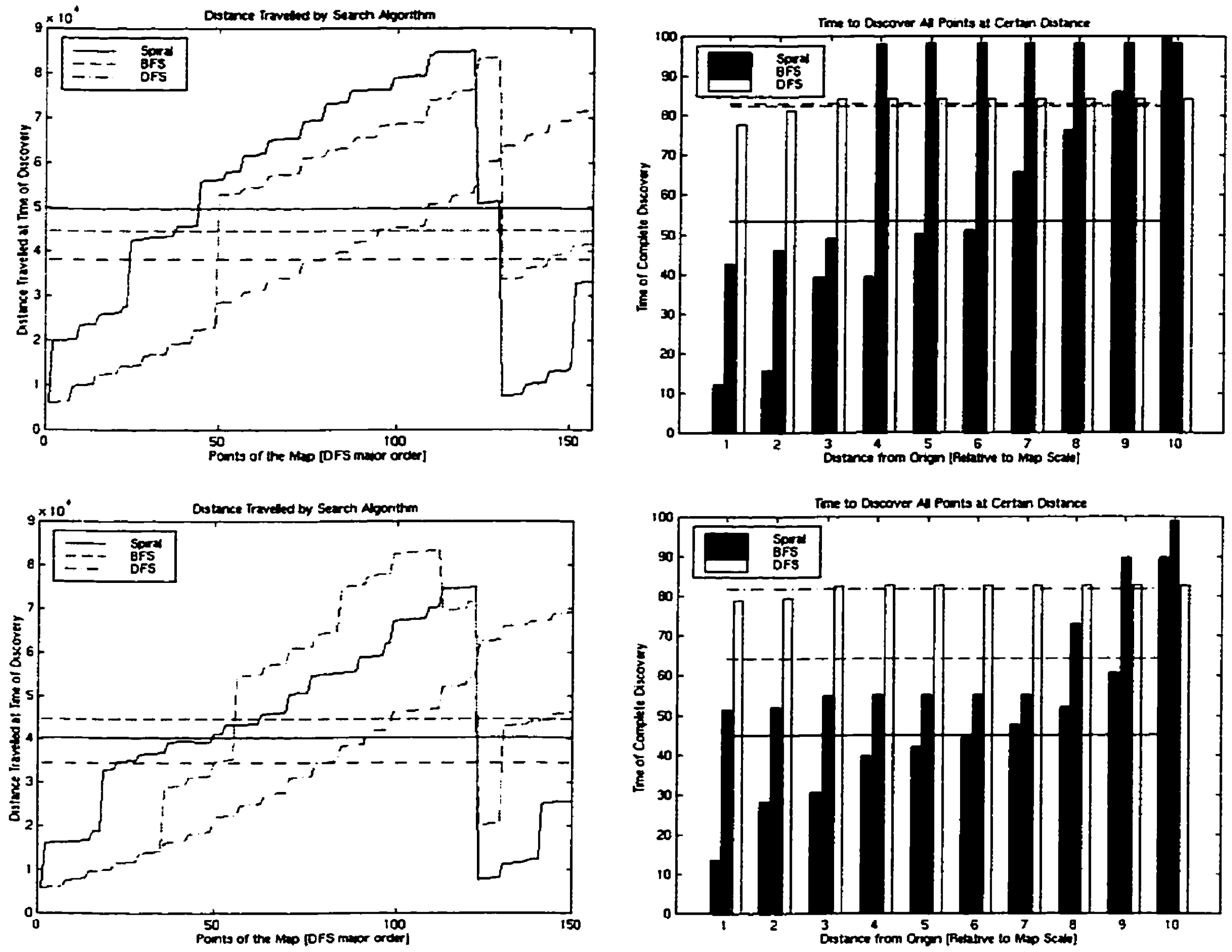


FIGURE 6.12. A common pitfall. The square shape of the *all-points distance* on the right indicates a shortcoming of depth-first search.

Both experiments shown in figure 12 have a common trait that illustrates a typical shortcoming of DFS. While DFS is very efficient, that efficiency is gained at the expense of neglect. Both of the results in the *all-points discovery* graphs on the right show poor performance for DFS. The shape of the bar graph in both experiments for DFS is nearly square. While DFS is efficiently searching points far off from the origin of search, there are points very near to the origin that are completely neglected. In both of these examples this effect is exacerbating in terms of not discovering a very large proportion of the area of the map until the very end of the search.

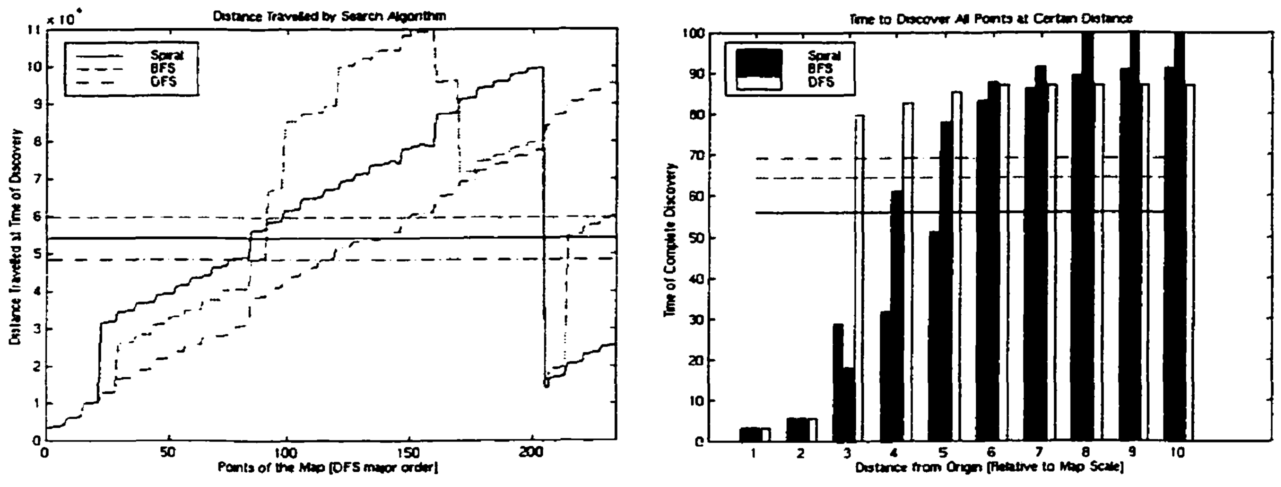


FIGURE 6.13. A fairly typical result from one of the experiments done on a map from the set of large maps. The relative mean performance of the search algorithm is staying reasonably consistent, but the scale of the dependent axis is increasing. This indicates a growing gap in absolute performance.

The results on one of the larger maps is shown in figure 13. The results over the 9 experiments performed on these largest sized maps did not vary greatly in relative terms from those results perceived in the first set of experiments on the automatically generated maps. The size of these maps would be approximately 30 meters x 30 meters assuming the robot were 0.5 meters in diameter. This corresponds to a 225% increase in the area of the maps.

The typical result shown in figure 13 shows that the relative performance of the search algorithms is remaining fairly consistent with results seen previously on the smaller maps. Certainly the absolute difference between spiral search and the performance of the other algorithms continues to grow. This same tendency is expected as the maps grow in size. The analysis that predicted these results were worst-case analysis predictions, and we see that the convergence to the worst-case behaviour has already began to show itself in our barrage of tests.

On unbounded we can predict the terminal null results for DFS. In the situation of a map with a breach in the hull, DFS will never discover the entire map. Once the depth first searcher discovers the breach to the "outer" world, it will never return to points near the root, if the "outer" world is infinite. Nor will BFS be guaranteed to search the entire of the interior. There is a guarantee that some non-trivial percentage of the map area will be discovered, but when BFS exits to the "outer" world (except for in some conditions of the "outer" world containing branching points in a specific manner) BFS will never return to the root of the search. However, we do have a guarantee that even it this terminal setting spiral search will still discover the entire area of the map eventually. Its turning points are stimulated partially by the distance traversed, so it is possible that even in an unbounded map that the area near the root of the map will be discovered. This sort of behaviour is indicative of the graceful degradation that is so highly desired in robotic system architecture.

5. Summary of Results

Presentation and analysis of the proposed experiments has been shown in this section. Extensive testing of the proposed algorithms has been performed on a simulator over a series of appropriate test maps. The data resulting from the experiments has been organized in 2 types of graphs. *Time of discovery* graphs depict the time at which a subset of regions from the environment were discovered by each of the algorithms and *all-points time of discovery* show the time taken to discover all the regions lying a certain distance from the starting point by each of the different algorithms.

Some observations have been noted regarding the interpretation of the underlying logarithmic spiral search pattern. An example from one of the experimental maps is used to elucidate the relationship between the spiral nature of the search algorithm and its observable effect during the progression of the search. A more detailed discussion of the characteristics of the experimental maps has also been made.

The predicted performance suggested by the underlying theory is indeed borne out by experimentation. The all-points discovery series is interesting as it indicates the worst-case

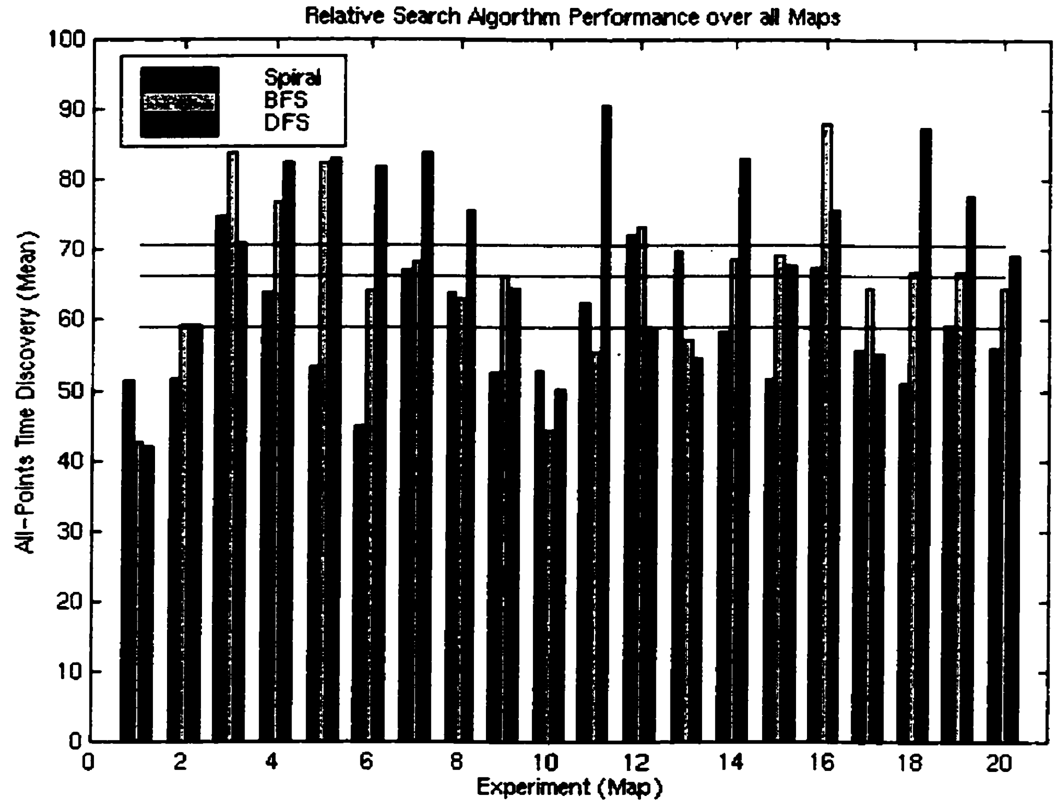


FIGURE 6.14. Summary Results

time taken to discover any point in the map given its distance from the search origin. It is this worst-case behavior that this thesis was intended to address. Spiral search has been seen to provide good worst-case results for discovering all points a given distance from the origin. Using the analogy of searching for a lost set of keys without being certain of the distance they lie from where they were assumed to be, then spiral search should provide a least cost approach to the search on average. The results from these sets of experiments reinforce our belief that spiral search achieves this primary objective. The worst-case time of discovery is significantly lower than depth or breadth first search on a large category of interesting maps.

The mean scores for the all-points discovery series have been collected and summarized from the entire set of experiments. The averages of these experiments show that spiral search indeed tends to perform significantly better at this task of all-points discovery. Spiral search over this battery of tests performs 11.9% better than BFS and performs 16.6% better than DFS in terms of average time to discover all points a certain distance from the origin. Figure 14 summarizes the *all-points discovery* mean performance for each of the experiments performed.

We have provided experimentally generated results showing the efficacy of spiral search over the more traditional types of search of depth and breadth first search. These results are drawn from a set of simulated test runs of the algorithms on interesting, non-trivial maps that are representative of office-like environments. The simulations have been built to exhibit reliable behaviour with attention paid not to impact any of the independent variables of the experiment addressed in chapter 5.

CHAPTER 7

Conclusion

In this thesis we have considered a new algorithm for efficiently searching with a moving robot in a planar environment containing obstacles. Our work integrates previous theoretical work with a functioning robotic simulation for testing and comparison of the new implementation.

Our work is founded in the theory of planar spiral search by Baeza-Yates *et al.* [11]. The techniques of spiral search that we have enumerated in chapter 2 are extensible to non-planar search, this is interesting future work. Investigations in the theory of search and search games in particular led to the sub-field of rendezvous search studied in detail by Alpern [1]. In chapter 2 we chronicle the development of a multi-robotic implementation by Roy [62] testing solutions to the rendezvous problem based on optimal theoretic results. In chapter 3 we provide analysis for the rendezvous problem in the half plane without obstacles.

Based on this theoretical work we develop a novel approach to a problem in planar search. First we construct a graph from the structured planar region that we face with the robot. Then, opposed to the more traditional methods of breadth- or depth-first graph search, we develop our brand of "modified spiral search" as an iteratively deepening depth-first search using the principles discovered in the theoretical work to ensure efficiency.

This is an extension of existing work [11] that proves the asymptotic optimality of spiral search in more abstract environments. Our work shows how to search a open or polygonal environment in a manner that is efficient in comparison to the optimal distance that would be traveling if the location of the goal were known in advance. This is accomplished by searching paths that are explored with a search radius that increases as a logarithmic spiral based also on the number of branches that have been detected in the environment. In the experimental work we present, the feasibility and typical performance is illustrated.

1. Future Work

1.1. Experimentation on an Actual Robot. The architecture the simulation has been built on is fully capable of interfacing with an actual robot. It is fully possible that experiments can be performed in actual environments given reasonable solutions to typical robotics problems.

(i) Place recognition — answering the question "Have I been here before?" is critical for traversal of previously seen territory, resolution of cycles in the map and for deciding when branches actually occur in the environment. Whether this is done using corrected odometry, dense sonar scans, laser range data, visual image methods or any combination of these, place recognition is crucial.

(ii) Localization — the ability to localize accurately the position of the robot in the environment based on sensor input and reference to an internal world representation.

(iii) Safe Navigation — navigating through the known map and into unexplored territory in a reliable and safe manner, so as not to incur damage to the robot, others in the environment, or to the environment itself.

With reasonable solutions to these problems, physical testing of the search algorithms can proceed.

1.2. Multiple Coordinated Robots. One exciting avenue for further work is examination of parallel implementations of search for use by multiple robots: this leads to issues to how to subdivide the problem and how and when to merge partial results. The rendezvous search approaches that we have studied would be ideal candidates for the implementation of multi-robotic search. The gains from work in parallel could be great given effective means of merging data and partitioning work among the agents in a dynamic way.

1.3. Variations on the Search Algorithm. The success of the modified spiral search algorithm in solving the *all-points distance* problem motivates testing with alternative search algorithms.

An alternative heuristic that seems apparent might be to use the modified spiral search approach, but attempt to make branch predictions based on suspected size of map and historical branch rate. A probability could be assigned to the length of new hallways, the distance traveled before an expected branch, the size of rooms, all based on reasonable inferences from known quantities from the map so far. It would also be interesting to try to predict the number of branches, M, in a map based on some probabilistic guess, and then have fewer updates to M dynamically during the search to see if this could be beneficial to search performance.

The accrual of positional error (for example via odometry) is a real problem. This suggests another search strategy that could involve returning to previously-explored regions slightly more frequently that might be suggested by the analysis presented here. The "localization visits" may contribute a level of reliability to the system that may facilitate portability to a real robotic system.

A related issue is that most real range sensors have an accuracy that diminishes with distance, and hence can be described in probabilistic terms. In some cases it may be preferable to search more deeply in regions were visibility is obstructed in order to assure a uniform coverage of free space along different paths. This suggests that probabilistic search may have a slightly different form than modified spiral search.

1.4. Variations on the Potential function. Perhaps search using different potential functions, rather than metric distance as used here, could make the algorithm proposed here for planar search appropriate to a larger variety of environments. Any *Lyapunov* function could act equally well as the potential function underlying the search. The hierarchy of spiral search techniques seems to prove efficient in more complex realms, as our experimentation here shows.

Perhaps adaptation to a higher dimensional search algorithm with a switch in the underlying potential function could prove to make this technique remarkably useful. The effective adaptation of spiral search to 3 dimensions may make contributions to the efficiency of underwater search or space exploration. Perhaps useful alternative potential functions may be the availability of light, for underwater visual application, or perhaps gravitational field potential for search of large areas of the galaxy.

2. Enumeration of Results

The expected results from the theory are borne out by experimentation. The all-points discovery series is interesting as it indicates the worst-case time taken to discover any point in the map given its distance from the search origin. It is this worst-case behavior that this thesis was intended to address. Spiral search has been seen to provide good worst-case results for discovering all points a given distance from the origin.

The results from experiments reinforce our belief that spiral search achieves the primary objective of efficiently searching in complex environments. The worst-case time of discovery is significantly lower than depth or breadth first search on a large category of interesting maps. The averages of these experiments show that spiral search indeed tends to perform significantly better at this task of all-points discovery. Spiral search over this series of tests performs 11.9% more efficiently than BFS and performs 16.6% more efficiently than DFS.

Modified spiral search provides also the secondary objectives that maps are searched in finite time, even if the environment is unbounded and that the search is performed in a non-neglectful manner. Any targets close to the origin will be discovered in reasonable time.

APPENDIX A

Proof of Spiral Search Optimality

1. Linear Spiral Search

THEOREM A.1 (Baeza-Yates, Culberson and Rawlins [**11**]). *Linear spiral search is optimal up to low order terms. That it is to say that the total distance walked by an agent performing linear spiral search is no more than* $9n$ *steps (see equation 14).*

PROOF. Let the point be found after the $(i+1)$th turn and before the $(i+2)$th turn, $\exists\, i$. Hence, i is such that $f(i)+1 \leq n < f(i+2)$ giving a worst case ratio of,

$$\max_{i\geq 1}\left(\frac{2\sum_{j=1}^{i+1} f(j) + f(i) + 1}{f(i)+1}\right) = 1 + 2\max_{i\geq 1}\left(\frac{\sum_{j=1}^{i+1} f(j)}{f(i)+1}\right) \qquad \text{(A.1)}$$

Since we already know that a $9n$ algorithm is possible (take $f(i) = 2^i$, see 14) suppose that f is such that

$$\frac{\sum_{j=1}^{i+1} f(j)}{f(i)+1} \leq c \qquad \forall\, i \geq 1, \qquad \text{(A.2)}$$

where c is constant. A lower bound on c will yield a lower bound of $(1+2c)n$ for the problem (from equation 23). Continuing, from 24 and the fact that f is strictly monotone increasing, so that we can always choose a large enough i such that

$$\sum_{j=1}^{i+1} f(j) - c > f(i) + 1$$

hence, we have

$$\begin{aligned} c &\geq \frac{\sum_{j=1}^{i+1} f(j)}{f(i)+1} \\ &> \frac{f(i+1)+f(i)+1}{f(i)+1} \\ &= 1 + \frac{f(i+1)}{f(i)+1} \end{aligned}$$

For a fixed and sufficiently large i, c must also satisfy the following infinite set of inequalities:

$$f(i+k) > \frac{f(i)+1+\sum_{j=1}^{k+1} f(i+j) - f(i+k)}{c-1} \qquad \forall k \geq 1$$

Together with the previous inequality on c this system of inequalities may be solved inductively for each k by deleting the $f(i+k)$ term on the right hand side, substituting the derived bound on $f(i+k)$ into the inequality for $f(i+k-1)$ and using that bound on $f(i+k-1)$, and so on.

In general, c must be such that the following polynomials are all positive:

$$g(k) = c^{k-1} \sum_{j=0}^{\infty} \binom{k-j}{j} (-1/c)^j$$

The minimal value of c for which each of these polynomials is greater than zero bounds c from below.

These polynomials obey the recurrence

$$g(k) = cg(k-1) - cg(k-2)$$

which has characteristic equation

$$\lambda^2 - c\lambda + c = 0$$

which has roots

$$\frac{c \pm \sqrt{c^2 - 4c}}{2}$$

which, having equal roots at $c = 4$ gives

$$g(k) = (k+1)2^{k+2}$$

and is positive for all $k > 0$, or, having distinct roots gives

$$g(k) = \frac{1}{c\sqrt{c^2 - 4c}} \left(\left(\frac{c + \sqrt{c^2 - 4c}}{2} \right)^{k+1} - \left(\frac{c - \sqrt{c^2 - 4c}}{2} \right)^{k+1} \right)$$

which is positive for all $k > 0$ if and only if $c > 4$.

Hence we have shown that c has a lower bound. Back-substituting into 24 we see that

$$\frac{\sum_{j=1}^{i+1} f(j)}{f(i) + 1} \leq 4,$$

and so 23 posits

$$1 + 2 \max_{i \geq 1} \left(\frac{\sum_{j=1}^{i+1} f(j)}{f(i) + 1} \right) = 1 + 2\,(4)$$

Therefore any algorithm to find a point on a line an unknown distance n, away must take at least $(1 + 2 \times 4)\, n = 9\, n$ steps. □

2. Generalized Linear Spiral Search

THEOREM A.2 (Baeza-Yates, Culberson and Rawlins [11]). *Generalized linear spiral search is optimal up to low order terms.*

PROOF. Let the point be found after the $(i + m - 1)$th turn and before the $(i + m)$th turn. The worst case ration is

$$1 + 2 \max_{i \geq 1} \left(\sum_{j=1}^{i+m-1} \frac{f(j)}{(f(i) + 1)} \right).$$

Let c be a constant such that

$$\sum_{j=1}^{i+m-1} \frac{f(j)}{(f(i)+1)} \leq c \qquad \forall i \geq 1.$$

Similar to 1 we construct an infinite sequence of functions of c (all positive), which are

$$g(k) = c^{k-1} \sum_{j=0}^{\infty} \binom{k+m-2-(m-1)j}{j} (-1/c)^j.$$

and obey the recurrence

$$g(k) = cg(k-1) - c^{m-1} g(k-m).$$

For which the characteristic equation

$$\lambda^m - c\lambda^{m-1} + c^{m-1} = 0$$

has a real positive double root at $c = m^m/(m-1)^{m-1}$ namely $\lambda = c(m-1)/m$, and all other roots are negative or imaginary. $\square$

APPENDIX B

Mathematical Concepts and Definitions

1. Group Theory

The following are some definitions from group theory that will be helpful in the general settings that we are going to be considering. For clarification on any of the following see Rudin [**64**] or any text on modern algebra.

DEFINITION B.1 (Group). *A set, G, and a binary operation, $\cdot$ (often called addition) such that*

i [Associativity] $a \cdot (b \cdot c) = (a \cdot b) \cdot c \quad \forall\, a, b, c \in G$

ii [Identity] $\exists\, e \in G$ *such that* $e \cdot a = a \cdot e = a \quad \forall\, a \in G$

iii [Inverse] $\forall\, a \in G\, \exists\, b \in G$ *such that* $a \cdot b = b \cdot a = e$ *and b is defined as "a-inverse",* $b = a^{-1}$.

DEFINITION B.2 (Abelian Group). *The group, G, is called* **Abelian** *if the operation is commutative for all the elements in the group. That is to say,*

$$a \cdot b = b \cdot a \qquad \forall\, a, b \in G.$$

DEFINITION B.3 (Isometry). *Let $\mathcal{X}, \mathcal{Y}$ be metric spaces, a bijective function, $f : \mathcal{X} \to \mathcal{Y}$, that preserves distances is an* ***isometry***.

$$d(f(x), f(y)) = d(x, y) \qquad \forall\, x \in \mathcal{X}, y \in \mathcal{Y},$$

where f is the Map and $d(a,b)$ is the distance function.

An isometry of the Plane is a linear transformation which preserves length. Isometries include Rotation, Translation, Reflection, Glides, and the Identity Map. Every isometry in the plane is the product of at most three reflections (at most two if there is a Fixed Point). Every finite group of isometries has at least one Fixed Point.

DEFINITION B.4 (Group Isomorphism). *An* isomorphism *ϕ from a group G to a group $\overline{G}$ is a one-to-one, onto mapping (or function) that preserves the group operation. That is,*

$$\phi(a \cdot b) = \phi(a) \cdot \phi(b), \quad \forall\, a, b \in G$$

If there is an isomorphism $\phi : G \to \overline{G}$ then the groups are ***isomorphic*** *and $G \approx \overline{G}$*

Although the following definition is not a group theoretic result, it is in some weak sense, the analog of the previous definition in topology and relating topological spaces rather than subsets.

DEFINITION B.5 (Homeomorphic). *Let $(\mathcal{X}, \mathcal{S}), (\mathcal{Y}, \mathcal{T})$ be topological spaces and let $h : \mathcal{X} \to \mathcal{Y}$ be bijective (1-1 and onto). The function h is a* ***homeomorphism*** *iff h is continuous and h^{-1} is continuous. If such a map exists then $(\mathcal{X}, \mathcal{S}), (\mathcal{Y}, \mathcal{T})$ are said to be homeomorphic.*

2. Metric Space and Measure Theory

DEFINITION B.6 (Distance). *A space, $\mathcal{X}$, is said to be a* metric *space if with any two points p and q from $\mathcal{X}$ there is associated a real number δ, $\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, called the distance from p to q such that*

$$\delta(p,q) > 0 \text{ if } p \neq q; \delta(p,p) = 0; \qquad \textit{(Positive Definite)}$$

$$\delta(p,q) = \delta(q,p); \qquad \textit{(Symmetric)}$$

$$\delta(p,q) \leq \delta(p,b) + \delta(b,q), \forall\, r \in \mathcal{X}. \qquad \textit{(Triangle Inequality)}$$

Any function with these three properties is called a distance function, *or a* metric *(from* [**64**]*).*

Because many properties of spaces are preserved by continuous functions, spaces related by a bijection (one-to-one and onto function) which is continuous in both directions will have many properties in common. These properties are identified as topological properties. Spaces so related are called homeomorphic. In our setting we exploit the property that two sets are homeomorphic to each other if they are *smoothly deformable* to each other without introducing or removing any fundamental characteristics of the initial set.

DEFINITION B.7 (Haar Measure). *An invariant measure on a locally compact group is often called a* ***Haar measure****. We usually want it to satisfy some regularity condition and also allow it to be complete.*

The next definition brings our attention to a much sought after class of functions important in the field of navigational methods. The generation of stable potential fields allows for the use of gradient descent navigation. Usually is it difficult to find generic, stable functions for use in any sort of complex setting. The shortcoming of gradient descent methods is clearly that the existence of local minima in the potential field result in basins of attraction that do not lead to the desired goal point. Functions that are said to be *Lyapunov functions* do not suffer from this problem of local minima.

DEFINITION B.8 (Lyapunov Function). *Any continuously differentiable, real valued function $U(x) \in C_1$ with a fixed point, x^*, in the domain of U such that*

(i) $U(x) > 0$ *for all* $x \neq x^*$ *and* $U(x^*) = 0$

(ii) $\nabla U(x) < 0$ *for all* $x \neq x^*$ *and* $\nabla U(x^*) = 0$ *(all trajectories on the domain are downhill)*

is said to be a ***Lyapunov function*** *and x^* is globally stable: for all initial conditions of vector x, $x(t) \rightarrow x^*$ as $t \rightarrow \infty$.*

REFERENCES

[1] S. Alpern and D. J. Reyniers, *Rendezvous and coordinated search problems*, Proc. IEEE Conference on Decision and Control (Piscataway, NJ), vol. 1, 1994, pp. 513–517.

[2] Steve Alpern, *The rendezvous search problem*, SIAM Journal of Control and Optimization **33** (1995), no. 3, 673–683.

[3] Steve Alpern, V.J. Baston, and Skander Essagaier, *The rendezvous search problem on a graph*, Tech. report, Center for Discrete and Applicable Mathematics, London School of Economics and Political Science, November 1997.

[4] Steve Alpern and Shmuel Gal, *Rendezvous search on the line with distinguishable players*, SIAM Journal on Control and Optimization **33** (1995), 1270–1276.

[5] Edward J. Anderson and Skander Essegaier, *Rendezvous search on the line with indistinguishable players*, SIAM Journal on Control and Optimization **33** (1995), 1637–1642.

[6] Edward J. Anderson and Sandor P. Fakete, *Asymmetric rendezvous search on the plane*, Submitted: SIAM Journal of Control and Optimization (1997), TBA.

[7] Edward J. Anderson and R. R. Weber, *The rendezvous problem on discrete locations*, Journal of Applied Probability **28** (1990), 839–851.

[8] Robert J. Anderson, *Smart: A modular control architecture for telerobotics*, IEEE Robotics and Automation Magazine **2** (1995), no. 3, 10–18.

[9] Ronald C. Arkin and J. David Hobbs, *Dimensions of communication and social organization in multi-agent robotics systems*, From animals to animats 2 : proceedings of the Second International Conference on Simulation of Adaptive Behavior (Hawaii), 1992, pp. 486–493.

[10] R. Baeza-Yates and R. Schott, *Parallel searching in the plane*, Computational Geometry: Theory and Applications **5** (1995), 143–154.

[11] Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J. E. Rawlins, *Searching in the plane*, Information and Computation **106** (1993), no. 2, 234–252.

[12] Tucker Balch and Ronald C. Arkin, *Communication in reactive multiagent robotic systems*, Autonomous Robots **1** (1994), no. 1, 27–52.

[13] ———, *Motor schema-based formation control for multiagent robot teams*, Proc. 1995 International Conference on Multiagent Systems (San Francisco), 1995.

[14] R. Bellman, *A minimization problem*, Bulletin of the American Mathematical Society **62** (1956), 270.

[15] J. L. Bentley and A. Chi-Chih Yao, *An almost optimal algorithm for unbounded searching*, Information Processing Letters **5** (1976), no. 3, 82–87.

[16] Avrim Blum, Prabhakar Raghavan, and Baruch Schieber, *Navigating in unfamiliar geometric terrain*, Proc. 23rd ACM Symposium on Theory of Computing, ACM Press, 1991, pp. 494–504.

[17] Encyclopedia Britannica, *Exploration*, internet publication, 1999, http://www.britannica.com/bcom/eb/article/4/0,5716,34034,00.html.

[18] R. Brooks, *A robust layered control system for a mobile robot*, Tech. Report AIM-864, MIT AI Lab, 1985.

[19] ———, *A layered intelligent control system for a mobile robot*, Robotics Research 3 (Faugeras and Giralt, eds.), MIT Press, 1986, pp. 365–372.

[20] B. Brumitt and A. Stentz, *Dynamic mission planning for multiple mobile robots*, Proceedings of the IEEE/International Conference on Robotics and Automation, 1996, pp. 2396–2401.

[21] Thomas Cormen, Charles Leiserson, and Ronald Rivest, *Introduction to algorithms*, McGraw Hill, 1990.

[22] Amitava Datta and Christian Icking, *Competitive Searching in a Generalized Street*, Proceedings of the Tenth Annual Symposium on Computational Geometry (Stony Brook, NY), ACM Press, June 6–8 1994, pp. 175–182.

[23] Andrew Davison and David Murray, *Mobile robot localisation using active vision*, submitted: European Conference on Computer Vision, 1998.

[24] ______, *Mobile robot localisation using active visual sensing*, submitted: IEEE Trans. PAMI, 1998.

[25] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun, *Monte carlo localization for mobile robots*, IEEE International Conference on Robotics and Automation (ICRA99), May 1999.

[26] ______, *Using the condensation algorithm for robust, vision-based mobile robot localization*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99) (1999).

[27] Xiaotie Deng and Christos Papadimitriou, *Exploring an unknown graph*, Annual Symposium on the Foundations of Computer Science, 1990, pp. 335–361.

[28] Robert Bingham Downs, *In search of new horizons : Epic tales of travel and exploration*, no. G 525 D751, Chicago : American Library Association, 1978, Adventure and adventurers, Explorers.

[29] Gregory Dudek, Paul Freedman, and Ioannis M. Rekleitis, *Just-in-time sensing: efficiently combining sonar and laser range data for exploring unknown worlds*, Proceedings of the IEEE International Conference on Robotics and Automation (Minneapolis, MN), vol. 1, April 1996, pp. 667–671.

[30] Gregory Dudek, Michael Jenkin, Evangelos Milios, and David Wilkes, *Robotic exploration as graph construction*, IEEE Transactions on Robotics and Automation **7** (1991), no. 6, 859–865.

[31] Gregory Dudek, Kathleen Romanik, and Sue Whitesides, *Localizing a robot with minimum travel*, Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM-SIAM, SIAM, Jan. 1995.

[32] Gregory Dudek, Kathleen Romanik, and Sue Whitesides, *Global localization: Localizing a robot with minimum travel*, SIAM Journal of Computation, vol. 27, SIAM, April 1998, pp. 583–604.

[33] Gregory Dudek and Chi Zhang, *Vision-based robot localization and pose estimation withou explicit landmarks*, Tech. Report XXX-xxx, Center for Intelligent Machines, McGill University, 3480 University Street, McGill University, Montreal, Que, Canada, H3A 2A7, JJJ 1994.

[34] F. Hara et al., *Effects of population size in multi-robots cooperative behaviors*, Proc. International Symposium on Distributed Autonomous Robotic Systems, 1992, pp. 3–9.

[35] Guy Foux, Michael Heymann, and Alfred Bruckstein, *Two-dimensional robot navigation among unknown stationary polygonal obstacles*, IEEE Transactions on Robotics & Automation. **9** (1993), 96–102.

[36] Schmuel Gal, *Search games*, Academic Press, New York, 1980.

[37] John L. Hennessy and David A. Patterson, *Computer architecture a quantitative approach*, Morgan Kaufmann, 1996.

[38] C. Icking and R. Klein, *Searching for the kernel of a polygon. a competitive strategy*, Proceedings of the 11th ACM Symposium on Computational Geometry, ACM, 1995.

[39] Rufus Isaacs, *Differential games*, Robert E. Kreiger Publishing Co., Huntington, New York, 1975.

[40] J. R. Isbell, *An optimal search pattern*, Naval Research Logistics Quarterly **8** (1957), 357–360.

[41] K. Ishioka, K. Hiraki, and Y. Anzai, *Coorperative [sic] map generation by heterogeneous autonomous mobile robots*, Proc. of the Workshop on Dynamically Interacting Robot (Chambery, France), 1993, pp. 58–67.

[42] R. M. Karp, M. Saks, and A. Wigderson, *On a search problem related to branch-and-bound procedures*, 27th Annual Symposium on Foundations of Computer Science (Los Angeles, Ca., USA), IEEE Computer Society Press, October 1986, pp. 19–28.

[43] R. Klein, *Walking an unknown street with bounded detour*, Computational Geometry: Theory and Applications, vol. 1, ACM, 1992, pp. 325–351.

[44] Jon Kleinberg, *On-line Search in a Simple Polygon*, Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, 1994, pp. 8–15.

[45] David Kortenkamp and Alan Schultz, *Integrating robotics research: Taking the next leap*, See AAAI Spring Symposia, http://www.aic.nrl.navy.mil/˜schultz/AR, http://tommy.jsc.nasa.gov/˜korten, http://www.aic.nrl.navy.mil:80/˜schultz, March 1998.

[46] Benjamin Kuipers, *Modelling spatial knowledge*, Cognitive Science **2** (1979), 129–153.

[47] Benjamin J. Kuipers and Y.T. Byun, *A qualitative approach to robot exploration and map-learning*, IEEE Workshop on Spatial Reasoning and Multi-sensor Fusion (Los Altos, California), IEEE, 1987, pp. 390–404.

[48] ———, *A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations*, Robotics and Autonomous Systems **8** (1991), 47–63.

[49] S. P. Lalley and H. E. Robbins, *Stochastic search in a convex region*, Probab. Theory Related Fields, 77, October 1988, pp. 99–116.

[50] John J. Leonard, Hugh F. Durrant-Whyte, and Ingemar J. Cox, *Dynamic map building for an autonomous mobile robot.*, International Journal of Robotics Research **11** (1992), 286–98.

[51] A. López-Ortiz and S. Schuierer., *Generalized streets revisited*, Proceedings of the European Symposium on Algorithms, Lecture Notes in Computer Science, vol. 1136, Springer-Verlag, 1996.

[52] ———, *Position-independent near optimal searching and on-line recognition in star polygons*, Workshop on Algorithms and Data Structures, ACM, 1997.

[53] Paul MacKenzie and Gregory Dudek, *Precise positioning using model-based maps*, Proceedings of the International Conference on Robotics and Automation (San Diego, CA), IEEE Press, 1994.

[54] M. Mataric, *Minimizing complexity in controlling a mobile robot population*, Proc. IEEE International Conference on Robotics and Automation, 1992, pp. 830–835.

[55] Christos H. Papadimitriou and Mihalis Yannakakis, *Shortest paths without a map*, Theoretical Computer Science **84** (1991), no. 1, 127–150.

[56] David Pierce and Benjamin Kuipers, *Learning to explore and build maps*, Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI, AAAI/MIT Press, 1994.

[57] Nageswara S. V. Rao, *Algorithmic framework for learned robot navigation in unknown terrains.*, Computer **22** (1989), 37–43.

[58] Eric S. Raymond, *Homesteading the noosphere*, internet publication, http://www.tuxedo.org/~esr/writings/homesteading.

[59] Ioannis Rekleitis, Gregory Dudek, and Evangelos Milios, *Multi-robot exploration of an unknown environment, efficiently reducing the odometry error*, Int. Joint Conf. on A.I. (see also CIM Tech. Report TR-CIM-97-01), Aug. 1997.

[60] ———, *Multi-robot exploration of an unknown environment, efficiently reducing the odometry error*, International Joint Conference in Artificial Intelligence

(Nagoya, Japan) (IJCAI, ed.), vol. 2, Morgan Kaufmann Publishers, Inc., August 1997, pp. 1340–1345.

[61] ———, *Reducing odometry error through cooperating robots during the exploration of an unknown world.*, Fifth IASTED International Conference Robotics and Manufacturing (Cancun, Mexico), vol. 1, IASTED, May 1997.

[62] Nicholas Roy, *Multi-agent exploration and rendezvous*, Master's thesis, Center for Intelligent Machines, McGill University, 3480 Rue University, Montréal, July 1997.

[63] Nicholas Roy, Wolfram Burgard, Dieter Fox, and Sebastian Thrun, *Coastal navigation – mobile robot navigation with uncertainty in dynamic environments*, Proc. IEEE Conf. Robotics and Automation (ICRA), May 1999.

[64] Walter Rudin, *Principles of mathematical analysis*, McGraw Hill, Montreal, 1964.

[65] T. C. Schelling, *The strategy of conflict*, Harvard University Press, Cambridge, 1960.

[66] Robert Sim and Gregory Dudek, *Learning visual landmarks for pose estimation*, Proceedings of International Conference on Robotics and Automation (Detroit, MI), May 4–6 1998.

[67] S.Thrun and A. Bücken, *Integrating grid-based and topological maps for mobile robot navigation*, Proceeding of the Thirteenth National Conference on Artificial Intelligence, AAAI-96, 1996, Portland OR, pp. 944–950.

[68] Brian Yamauchi, *A frontier-based approach for autonomous exploration*, CIRA'97, IEEE, July 10 1997, Monterey CA.

[69] Brian Yamauchi, Alan Schultz, and William Adams, *Integrating exploration and localization for mobile robots*, to appear in *Adaptive Behavior*.

[70] ———, *Mobile robot exploration and map-building with continuous localization*, IEEE International Conference on Robotics and Automation, IEEE, May 1998, pp. 3715–3720.

[71] Alexander Zelinsky, *A mobile robot exploration algorithm.*, IEEE Transactions on Robotics & Automation **8** (1992), 707–17.

Index

Document Log:

Manuscript Version 5 — 3 May 2000

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX — 3 May 2000

Graphic Figures Set by:

SCOTT BURLINGTON

electronic versions of this document are available

contact the author

CENTER FOR INTELLIGENT MACHINES (416), MCGILL UNIVERSITY, 3480 UNIVERSITY ST., MONTRÉAL (QUÉBEC), CANADA, H3A 2A7, *Tel.* : (514) 932-2186

E-mail address: `scottyb@cim.mcgill.ca`

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX