Collaborative Robot Exploration and Rendezvous

Algorithms, Performance Bounds and Observations

Nicholas Roy (nicholas.roy@ri.cmu.edu)
Robotics Institute, Carnegie Mellon University
Pittsburgh, PA USA *

Gregory Dudek (dudek@cim.mcgill.ca)
Centre for Intelligent Machines, McGill University
Montreal, PQ Canada †

Abstract. We consider the problem of how two heterogeneous robots can arrange to meet in an unknown environment from unknown starting locations: that is, the problem of arranging a robot rendezvous. We are interested, in particular, in allowing two robots to rendezvous so that they can collaboratively explore an unknown environment. Specifically, we address the problem of how a pair of exploring agents that cannot communicate with one another over long distances can meet if they start exploring at different unknown locations in an unknown environment.

We propose several alternative algorithms that robots could use in attempting to rendezvous quickly while continuing to explore. These algorithms exemplify different classes of strategy whose relative suitability depends on characteristics of the problem definition. We consider the performance of our proposed algorithms analytically with respect to both expected- and worst-case behaviour. We then examine their behaviour under a wider set of conditions using both numerical analysis and also a simulation of multi-agent exploration and rendezvous. We examine the exploration speed, and show that a multi-robot system can explore an unknown environment faster than a single-agent system, even with the constraint of performing rendezvous to allow communication.

We conclude with a demonstration of rendezvous implemented on a pair of actual robots.

Keywords: multirobot, exploration, rendezvous

1. Introduction

In this paper, we consider a particular aspect of multi-robot environment exploration: how to get a pair of robots to meet one another initially in an unknown environment if they do not know one another's starting positions. The problem of *robot rendezvous* is a key step in collaborative exploration. We address this problem, once it has been formally defined, by considering it in several ways spanning closed-form analysis and real-world experimentation.

In many contexts, multiple-robot systems may be faster or more powerful than a single robot system. However, there are difficulties associated with the use of such multi-agent systems. Task division, synchronisation and coordination are significant problems, as is maximising the efficiency of the distributed team. Practical considerations can further contribute to the complexity of a multi-agent system when compared with a single agent system.

One example of such a practical limitation is inter-agent communication. Existing research indicates that multi-agent robot systems for the majority of real-life applications enjoy substantial speed gains only with some level of communication (Balch and Arkin, 1994), when compared with single-agent systems or multi-agent systems that do not communicate. Many distributed-agent algorithms, for instance dynamic path-planning, assume and rely upon instantaneous, infinite bandwidth communication between agents at all times in order to achieve promised performance levels (Brumitt and Stentz, 1996). However, most existing hardware agents are only capable of communication over short

[*] This work was carried out while the first author was at McGill University.
[†] The authors gratefully acknowledge the support of the National Science and Engineering Research Council.

distances. Environmental geometry, wireless transmission technology, power considerations and atmospheric conditions (or water conditions for underwater agents) all contribute to limitations on communication range. In the absence of sophisticated satellite receivers or high power devices, a common constraint for successful communication is maintaining "line-of-sight" between agents.

Since many realistic robots must be near one another to communicate, this implies they need to be able to rendezvous. In addition, the agents must be able to merge the maps generated by the exploration process; if the maps cannot be merged, then each agent must itself explore the environment to completion, and no task speed-up is achieved. Under many circumstances, heterogeneous agents must share a common reference point to be able to merge maps[1] - completely independent maps cannot always be merged reliably. Unless the agents start at exactly the same place in the environment, they must agree on a place to meet *a priori*, and share information. However, choosing a meeting point reliably, especially in an unknown, unconstrained environment is difficult.

1.1. Problem Statement

We are interested in multi-robot exploration using range-limited "line of sight" sensors such as vision or sonar. In practice, the particular sensing modality has numerous pragmatic implications, a major factor being the range at which the agents can either recognise one another, or any landmarks in the environment. In the context of a general rendezvous strategy, we will initially consider a generic "abstract" sensor that both

[1] It is sometimes possible to merge maps using their shapes. However, if the agents' sensors are substantially different, or there are spatial ambiguities, the merging process may fail.

allows the agents to recognise one another when they are sufficiently close together and also allows them to evaluate any point in space as to its suitability as a rendezvous point. We describe how the rendezvous task can be efficiently accomplished under various assumptions about the environment and the perceptual abilities of the agents involved.

To restate this more formally, given two heterogeneous robots equipped with noisy sensors that can be used for mapping, and a sensor that can be used for computing a local signature of the position in the environment, how can the robots both explore the map and meet in minimum time? In the simplest approach to the problem, the rendezvous will have the agents search through the environment for good meeting points, and then have them travel to the single best meeting point at a pre-arranged time. In practice, more complex approaches are required.

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

1. The first sub-problem is how to choose an appropriate rendezvous point, given an unknown environment. The ability of the agents to meet in the environment is a function of their ability to reliably choose appropriate rendezvous points. For instance, mountain tops may be a good outdoor rendezvous point.

2. The second sub-problem is that of dealing with confounding factors in the rendezvous process. One such factor is the difficulty of coming to an agreement on the location for a rendezvous. Sensor noise may cause agents to disagree; agents may not have explored the same regions of space, and therefore may choose different points. An appropriate rendezvous strategy must take into account such asymmetry between the agents' exploration. Pre-arranged behaviour must also account for such asynchronies and allow for missed rendezvous attempts.

1.2. Outline

In section 3, we formalise the parameters of the rendezvous problem that necessitate more than one attempt. In section 4, two classes of solutions are proposed and then analysed analytically. We simulate the rendezvous problem at two levels; the first level, section 5, is a purely algorithmic simulation, simply to test the efficacy of the various algorithms under the different conditions we describe. In section 6, we develop a realistic simulation, using spatial metrics and simulated sensing and motion. Finally, we demonstrate in section 7 the speed-up possible under multi-agent systems by comparing the running-time of the multi-agent system versus the single-robot system on a real multi-robot system.

2. Previous Work

The problem of rendezvous is not a new one; there exists a body of research in the optimisation and operations research community involving search problems. Rendezvous is a particular variant of the search problem, similar to games with mobile hiders, called *princess and monster games* (Alpern, 1995). There are many variants of the rendezvous problem itself, involving distinguishable (Alpern and Shmuel, 1995) or indistinguishable agents (Anderson and Essegaier, 1995) and collaborating or interfering agents. The environment may have focal points, or may be completely homogeneous.

There are a number of differences between the highly theoretical approach most prior work takes and the approach used here. In our work, the environment is not known, and one of the key problems is to *find* the focal points (what we term *landmarks*). Secondly, in prior work, the theoretical agents have perfect sensing, synchronisation, etc. Here, we are dealing with realisable agents, with the concomitant problems of noise, asynchrony, real-time travel limitations.

However, there is a key similarity between ours and the prior work, in that communication between agents is prohibited, until the agents are within a pre-determined line-of-sight range. Indeed, the graph-theoretic approach reduces this distance to 0 in many cases. It is interesting to note that one of the algorithms proposed by Alpern (Alpern, 1995) is equivalent to the first deterministic algorithm we propose in section 3.

2.1. Multi-agent robotics

Balch and Arkin (Balch and Arkin, 1994; Balch and Arkin, 1998) describe several tasks: *consuming*, *foraging* and *grazing*. The task of exploration is an example of a grazing task, in that each point in the environment needs to be covered by at least one robot's sensors, in order to acquire a complete representation. Further tasks that have been addressed in the context of multi-agent systems are box-pushing (Parker, 1994; Donald, 1995), formation holding (Beni and Liang, 1996) and exploration and mapping (Rekleitis et al., 1997; Cohen, 1996). Like Donald's box-pushing and the grazing task, many of these applications use passive sensing or implicit information to perform their task. There is a dearth of work in real applications that demand full, active communication that have been implemented on real robots. Rekleitis and colleagues' (Rekleitis et al., 1997) work on using multiple robots for exploration is very much in the spirit of this work, using multiple agents to overcome limitations in the use of a single robot for exploration. While their approach overcomes inherent limits in localisation in an unknown environment, the goal is to increase the precision of the map acquired. This work is focussed on increasing the speed of map acquisition.

A comprehensive taxonomy of the different types of multiple-agent systems, or *swarms* has been proposed (Dudek et al., 1996), including the various types of communication available. In addition, the three possible types of communication were described as

- No communication ("COM-NONE")
- limited communication ("COM-NEAR")
- full communication ("COM-INF")

As the authors point out, COM-INF "is the classical assumption, which is probably impractical if [the number of agents] $\gg$ 1." A modest understatement, given that radio communication breaks down in many situations as soon as line-of-sight is lost. Our work makes the assumption of swarms of range-limited communication, instantiated in this case using a line-of-sight constraint.

There has been considerable work in studying the range of behaviour of multiple-agent systems, especially attempting to maximise efficiency and minimise complexity (Mataric, 1992) (Hara et al., 1992). Mataric has looked at models of collaborative behaviour between mobile robots (Mataric, 1992), and examined the "emergent behaviour" properties that result. She also observed that the form of communication plays an important role in how collaborative actions proceed. Parker has developed control strategies for heterogeneous multiple robot systems, and made clear the need for effective communication (Parker, 1994).

Finally, the problem of map generation from co-operative multi-agent exploration was discussed and implemented first by Ishioka et al. (1996). Their work is a canonical example of the potential applications of the technique presented in this paper, in which co-operative heterogeneous robots generated maps of unknown environments. They did not discuss the problem of rendezvous, but focussed only on how to merge maps once the rendezvous has occurred. Later work also assumes rendezvous has occurred (Rao et al., 1996), and the latest work by Fox et. al. further develops the ideas of detecting rendezvous visually, and map merging probabilistically (Fox et al., 2000).

While it is clear from the graph-theoretical work that focal points in the environment are essential to effective rendezvous, it is Kuipers' selection of distinctive locations in a simple 2-D environment (considered previously in the context of map-making (Kuipers and Byun, 1991)), that is the basis for the landmarks in this work. The distinctive locations in that work were determined by active hill-climbing over the distinctiveness function, that is, by local gradient ascent over some function of the sensor output. The local maxima in a continuous property of the environment allowed for the conversion of a metric environment representation into a graph-like or topological one (Chatila and Laumond, 1985; Dudek et al., 1991; Kuipers and Byun, 1991; Shatkay and Kaelbling, 1997; Thrun, 1998).

3. The Rendezvous Problem

In the simplest, idealised, noise-free case, the robots have a pre-arranged notion of what constitutes a good rendezvous point. At a pre-arranged time, the robots go to the best rendezvous point, and wait for the other robot(s) to arrive. They can then fuse their maps and suitably partition any remaining exploration to be done.

This simple strategy can be decomposed into the following four steps:

1. Travel throughout environment
2. Find good rendezvous locations
3. At the pre-arranged meeting time, choose the best rendezvous location
4. Travel to that rendezvous location, and share information with the other agents

In the following sections, we describe how to choose good rendezvous locations. We then formalize what can cause a rendezvous attempt to fail, and how our rendezvous strategies recover from failures.

3.1. Defining Rendezvous Points

In the context of cultural environments, typical notions of good rendezvous locations – we refer to these points as **landmarks** – generally rely upon some *a priori* knowledge of the environment. For instance, humans often rely upon existing structures such as doors of buildings or monuments. We would like to avoid assumptions about the availability of such structures, therefore, we define the notion of **distinctiveness**, or **landmarkness**, as a value defined at every point in the environment, and use this value to find landmarks. If the distinctiveness is a function of the agent's sensor(s), then there is no issue of environmental dependence on the ability to find landmarks — every location is a potential landmark.

We refer to the scalar measure of suitability of a particular point (x, y, θ) as its distinctiveness: $D(x, y, \theta)$. The position (x, y) and orientation θ of the robot are commonly termed the *pose* of the robot, defined over the configuration space $\mathcal{C}$ of the robot. For a pose vector $\mathbf{q}$ we can define $D(\mathbf{q}) : \mathcal{C} \rightarrow \mathcal{R}$ that maps from the configuration space of the robot to a real-valued scalar.

The quantity $D(\mathbf{q})$ is implicitly a function of sensor data $\mathbf{f}(\mathbf{q})$, so we have a new distinctiveness function $\hat{D}$, such that:

$$D : (x, y, \theta) \rightarrow \mathcal{R} \tag{1}$$

$$\mathbf{f} : (x, y, \theta) \rightarrow \mathcal{S} \tag{2}$$

$$\hat{D} : \mathcal{S} \rightarrow \mathcal{R} \tag{3}$$

$$D = \hat{D} \circ \mathbf{f}(\mathbf{q}) \tag{4}$$

Some intuitive examples of environmental attributes that might serve as distinctiveness measures are spatial symmetry, distance to the nearest obstacle, or altitude (for 3D surfaces – for example humans might select hill tops). If we choose our the distinctiveness function to be orientationally invariant, then

$$D(x, y, \theta) = D(x, y) \tag{5}$$

The possible distinctiveness measures are heavily dependent on the types of sensors the robots have at their disposal. Because the robot assigns a value to every point, a good sensing modality is one that allows the distinctiveness to be defined at any location in the environment, and for which there exists some metric that can order the resultant landmarks in the environment in terms of distinctiveness. This ordering allows the landmarks to be ranked in terms of their likelihood to lead to a successful rendezvous. By far the most important feature of the distinctiveness function is that the locations of its extrema should be robust with respect to small changes in position, so that these extrema can be found again later.

It is important to note that the distinctiveness values are only computed for locations actually visited by the robot. By restricting landmarks to lie along the robot trajectory, we avoid issues of landmark visibility and viewpoint independence. Consequently, rendezvous locations need not be recognisable as such from afar, such as a mountaintop is. We recognise locations as landmarks by actually visiting them.

The function $D(x, y)$ defines a surface across the $x - y$ plane. Landmarks are defined as the local maxima (or minima, if preferred), of the distinctiveness surface. Certain generic properties apply to good distinctiveness functions, independent of the sensing modality. If $D(x, y)$ is smooth, locally convex, and has few local extrema or inflection points, then it is easy to find highly stable and mutually agreed-upon extrema. Landmarks can be found by the robots performing gradient ascent over $D(x, y)$. However, although this strategy is attractive in principle, we believe that in many real environments, occlusion, noise, and other factors may make the "distinctiveness surfaces" highly non-convex and thus complicate the process.

3.2. Finding Landmarks

One way to identify potential rendezvous points, or landmarks, is to sample the distinctiveness surface uniformly across the space, and then identify the maxima in the surface off-line. However, the task of locating landmarks for rendezvous cannot always dictate the robot trajectory. Although we are developing the technique of multi-agent rendezvous for exploration, we would like to generalise rendezvous to any multi-agent system. The constraints of some tasks may not allow the agent to suspend execution of the primary algorithm in order to follow the distinctiveness surface, hunting for landmarks. As a result, the agents must be able to identify landmarks during the execution of any task.

We therefore impose two constraints on the distinctiveness function - the function must be trajectory-independent and orientation-independent. For example, the “Northern-most” point in the already-explored environment is a poor choice. If the explored area of each robot is circular, then two robots will only have the same “northern-most” point if the environment is highly constrained or if the explored regions are very similar. In Figure 1, we see that despite having relatively similar explored regions, the robots will choose quantitatively different landmarks. If the landmarks are separated substantially, either by distance or by some obstacle such as a wall, a rendezvous at these landmarks will fail.

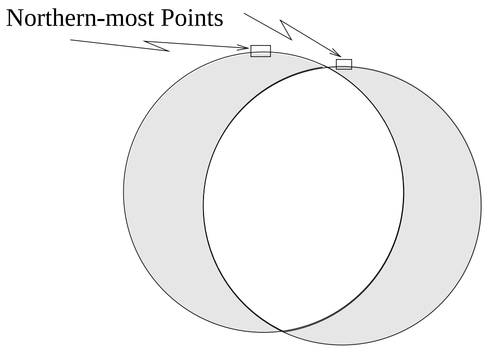


Figure 1. An example of the effect of choosing a poor measure of distinctiveness. Even though the two robots have relatively similar explored regions, the best landmark each chooses is different enough to cause rendezvous difficulties.

Similarly, an orientation-dependent distinctiveness function will give very different values for a robot looking down a corridor, as opposed to a robot looking at a wall. Unfortunately, most immediately obvious distinctiveness functions are orientation-dependent, especially those that use the sonar rings found on most mobile robots. The solution we have chosen is to sample the distinctiveness function at pre-determined orientations.

There still remains an issue of spatial sampling — as the agents travel through the environment they must sample the distinctiveness function sufficiently often to be able to specify the landmarks accurately. Coarse sampling can lead to incorrectly estimating a peak's height, or even failing to observe a good landmark entirely. One possible solution is to sample the distinctiveness surface along the trajectory as finely as possible. However, since the distinctiveness sampling is a function of the task-dependent trajectory, the problem is independent of the distinctiveness function and must be addressed in some other manner. Possible solutions to this problem will be discussed with rendezvous algorithms.

3.3. Inter-agent Differences and Sensor Noise

In addition to using the same distinctiveness function, the agents must compensate for differences in their perceptions of the environment. In order for two robots to agree on a good landmark, they must have similar perceptions of the environment or be able to convert their percepts into a common intermediate form. In the extreme case of two agents with dramatically different sensing modalities, there is essentially no way for them to rendezvous based on the recognition of environmental characteristics. Sensor noise can play a similarly problematic role. We model this aspect of the problem by parameterising the extent to which the two agents can reliably obtain the same measurement of distinctiveness at the same location.

We consider the base case, $D(x, y)$, to be "ground truth" with respect to the distinctiveness that should be measured by all the agents.

However, $D(x,y)$ is a function of the sensors the agents use:

$$\mathbf{S}_i(x,y) = \mathbf{S}(x,y) + \eta_i(x,y) + \lambda_i \quad (6)$$

$$D_i(x,y) = D(\mathbf{S}_i(x,y)) \quad (7)$$

where $\mathbf{S}(x,y)$ is the ideal perception of the environment by the given sensor, in the absence of any noise. $\mathbf{S}_1(x,y)$ is Agent 1's perception of the environment at position (x,y) that encapsulates the agent's systematic error $\eta(x,y)$ over the measurement at that position; λ_i is that agent's random sensor noise.

For the purposes of modelling the inter-agent differences, we model λ and η as scalars, and collapse them into one error term. With full generality, we can consider one of the agents as the reference perceiver (the arbiter of good taste) with a percept $D_1(x,y) = D(x,y)$ while the other robots obtain a sensor measurement which can be viewed as noisy with respect to that of the first robot:

$$D_i(x,y) - D_1(x,y) = \hat{\delta}_i\hat{\eta}_i(x,y) + \hat{\delta}_i D_1(x,y) \quad (8)$$

$$D_i(x,y) = (1-\hat{\delta}_i)D_1(x,y) + \hat{\delta}_i\hat{\eta}_i(x,y) \quad (9)$$

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. If both robots have exactly the same perceptions of the environment we have $\hat{\delta} = 0$. In the context of this formalism, $\hat{\eta}(x,y)$ combines both intrinsic sensor noise and any differences in the type of sensor used. Note that the for the purposes of modelling differences in sensor measurement across agents, the $\hat{\eta}$ and $\hat{\delta}$ parameters can be treated as a single parameter, δ.

4. Rendezvous Strategies

In the ideal case, the obvious choice is the "best" landmark, i.e., the point in the environment that has the largest known maximum of the distinctiveness function. For a variety of reasons, this simple strategy proves to be difficult or impossible to achieve in practice. Therefore, strategies must be developed to accommodate the various confounding factors that make the rendezvous problem challenging in practice.

4.1. Formal Parameters of the Rendezvous Problem

In order to estimate the effectiveness of alternative strategies for rendezvous, we have identified key attributes that must be formalised. Important attributes of the rendezvous problem are:

- Sensor noise — the distinctiveness measures observed by the two robots are unlikely to be identical. This is expressed by the $\delta(x, y)$ term of Equation 9.

- Landmark Commonality — the extent of overlap between the spatial domains of the agents.

  In the ideal case, the agents will share all landmark knowledge. More likely is that the robots have explored partially-overlapping areas, and will have some different landmarks d that are not in the common region, of a total set of n landmarks.

- Synchronisation — the level of synchronisation between the agents.

  If the agents do not agree on the rendezvous time, there is a probability that the rendezvous will be missed. Also, if an agent fails to arrive at the landmark because there of travel delays, the rendezvous will fail. The probability that a rendezvous is missed is modelled by the parameter j.

- Landmark Cardinality — the number n of points considered for rendezvous by each agent.

  If there is exactly one landmark, then the rendezvous algorithm cannot make any attempt to compensate for variations in the problem parameters. In this extreme case, the problem is "solved" simply by revisiting that one landmark. At the other extreme, if every point visited is considered as a landmark, the algorithm may be swamped, preventing it from exploiting its abilities to find the other agents.

It is assumed that if the agent roles are asymmetric that there is an *a priori* agreement of which agent will play which role. Furthermore, we assume that all agents share some notion of synchronisation — that is, all agents can agree on when rendezvous attempts should be made, however, this synchronisation may be noisy. A third assumption is that all agents have the same landmark set cardinality — they all attempt rendezvous over the same number of landmarks (even if they are not using identically the same landmarks in their sets). Finally, it is assumed that all agents are performing the same task, and using the same rendezvous strategies.

4.2. Landmark Selection Algorithms

Looking to biology, some simple algorithms for related problems are observed. One common strategy has one agent (e.g., a child lost at the zoo) wait to be found while other agents (e.g., desperate parents) cover the space, performing search. Another equally naive but much less common strategy has agents moving from landmark to landmark randomly until a rendezvous occurs.

We have developed two main classes of algorithm: deterministic and probabilistic. The deterministic class of algorithm creates a list of all possible combination of landmarks and specifies the order in which the landmarks should be visited. There is no random aspect to the landmark visit sequence, and therefore the algorithms will generate the same sequence of landmark visits for a given landmark set. The probabilistic class of algorithm does not generate an *a priori* ordering of landmarks, but simply generates probabilities for landmarks being visited at any proposed rendezvous.

4.2.1. *Deterministic Algorithms*

- **Sequential** – One agent picks a landmark and waits there for the other agent, which visits every landmark in turn. If the second agent has visited every landmark without encountering the first agent, the first agent moves to another landmark it has not yet visited.

The active agent cycles through all its landmarks, before returning to the beginning of the set. The passive agent remains at a landmark for n cycles, where n is the size of the landmark set, before moving to the next landmark. This generates a list of all pair-wise combinations of landmarks, sorted by distinctiveness. This strategy gives the agents asymmetric roles with respect to one another. The extension from a single pair of agents to an arbitrary number of agents can be easily accomplished by evenly dividing the agents into two classes of active and passive agents.

– **Smart-sequential** – Each pairwise combination of landmarks known to a robot is assigned a "goodness" value. This value is the product of the distinctiveness of the pair. The list of landmark pairs is sorted by this product, and one side of each pair is discarded, leaving an ordered list of n^2 landmarks from a set of n. The robot then visits the landmarks in this order.

The smart-sequential strategy takes into account the fact that the landmarks may be mis-ordered across agents: that is, one agent's sorted list will not quite match the other's, and that the relative mis-orderings are likely to be small (that is, one list can be regarded as an almost-sorted version of the other). The landmark orderings can be thought of as being "perturbed" rather than grossly misordered across agents. Consequently, it may make more sense to revisit highly distinct landmarks long before considering landmarks with relatively low distinctiveness. This leads to an increased probability of meeting even with substantial asynchrony between agents, or with high-valued landmarks that are unique to one agent. The smart-sequential method is tantamount to guessing where the other robot might be, given relatively similar, but not identical, landmark rankings.

4.2.2. *Probabilistic Algorithms*

The probabilistic algorithms use different probability functions to accommodate different parameters of the problem space. The landmarks are sorted with respect to their distinctiveness and then assigned a likelihood of visitation p_i for landmark i as a function of its rank in the sorted list, i.e., $p_i = f(i)$. The algorithm probabilistically selects a landmark to visit, using p_i for each landmark. For example,

- **Exponential** – The likelihood of visiting the $i-th$ best landmark is $\propto e^i$. This function has the effect of emphasising the relatively highly distinct landmarks, at the cost of landmarks with relative low distinctiveness.

- **Random** – On each attempted visit, each robot selects a landmark at random and goes there.

The particular exponential function used in the simulations was

$$P_i = \rho e^{\tau(D_1 - D_i)} \tag{10}$$

$$\tau = \frac{.25log(.001/D_1)}{D_1} \tag{11}$$

where D_i is the distinctiveness of landmark i and P_i is the probability of visiting that landmark. ρ is a normalisation constant to ensure that the probabilities for the landmark set sum to 1.0, and τ is a user-definable decay constant for tuning the exponential function response. The constants in these formulae were chosen empirically.

4.3. Analytical Results

We can make an analytical assessment of the bounds on the performance of the deterministic rendezvous algorithms, compared to the random algorithm baseline. We examine the performance of the algorithms in the limit of high heterogeneity and noise, $\delta = 1$, such that no common ordering between agents of the same landmarks can be reliably determined. The first assessment is the algorithmic time complexity, i.e., the expected time to rendezvous, for the three algorithms in the limit of $\delta = 1$. For a landmark set of size n, the failure probability of any single, random rendezvous attempt is $P_{unsuccessful} = \frac{n-1}{n}$ and if the asynchrony rate is accounted for, then the failure probability rises to $P_{unsuccessful} = \frac{n-1}{n}j$.

These facts give rise to table I. The first column refers to the case in which both robots having the same set of landmarks. The second column considers the scenario where the robots may fail to get to the appointed landmark at the same time (or fail to notice one another). This probability is the asynchrony, j. The third column deals with the case where d of each robot's n landmarks are not in the other robot's landmark set.

Table I. Expected case behaviour. The columns denote the ideal case, the case where the asynchrony $j \neq 0$ and the case where the landmark sets are not identical, but each agent has d non-common landmarks.

Algorithm	Simple	Async.	< 100% Comm.
Random	$\frac{1}{log_2(\frac{n}{n-1})}$	$\frac{1}{log_2(\frac{n}{n-1})+log_2 j}$	$\frac{1}{log_2(\frac{n}{dn-d})}$
Sequential	$n/2$	$\frac{n}{2} + j^{\frac{-1}{\log j}}$	$\frac{n}{2} + \frac{d}{n}^{\frac{-1}{\log \frac{d}{n}}}$
Smart-seq.	n	$n + j^{\frac{-1}{\log j}}$	$n + \frac{d}{n}^{\frac{-1}{\log \frac{d}{n}}}$

In the deterministic sequential algorithm, the expected time of the simplest case (identical landmark sets, no asynchrony), is very straightforward. One agent sits at a landmark, and the other agent visits every landmark in turn until they meet — on average $n/2$ landmarks. However, in the presence of asynchrony, additional sweeps of all n landmarks will have to be performed. To find the expected number, k such additional sweeps, we use $0.5 = j^k$ noting that each extra sweep i of k will reduce the probability of failure, and k such sweeps must reduce the probability of failure to 50%. Thus, on average $j^{\frac{-1}{\log j}}$ sweeps during the rendezvous will fail due to asynchrony. Similarly, for non-identical landmark sets, additional sweeps of n landmarks will have to be performed on average $\frac{d}{n}^{\frac{-1}{\log \frac{d}{n}}}$ times.

5. Numerical Simulation

Unfortunately, the analytical description of the algorithms given above does not provide a realistic picture of the the performance of the algorithms, as this bound will rarely, if ever, be attained in practice. The agent difference, λ, will likely not be extremal. Therefore, more useful than the analysis in the limit of high noise is the performance of the algorithm under conditions of worsening noise, especially under different conditions of disjoint landmark sets and asynchrony. We therefore use a numerical analysis technique to examine the algorithm under a range of conditions.

5.1. Experiment Design

Two agents were modelled as having already explored an unknown area, and having collected a set of n landmarks. The distinctiveness values of the ordered landmarks were generated with a function, $f(x)$ where x was the landmark index. $f(x)$ was a linear function for the results given here, although other functions were examined. Random noise δ as developed in equation 9 was then applied to the two sets. The appropriate rendezvous strategy was then used to generate a sequence of landmarks for the two agents, with a maximum length of n^2. The sequences were terminated at the first position with the same landmark, and the running time was considered to be the length of the sequences.

The landmark set is generated by a distinctiveness distribution model $F(i)$, with a range of values, $[0, max(F(i)]$. The noise was then modelled as a percentage δ of full scale:

$$D_i = F(i) + Random(0 : \delta \cdot \max F(i)) \qquad (12)$$

The random function was a uniform random function in the range $[0 : \delta \cdot \max F(i)]$. The distinctiveness values D_i were then re-normalised into the range $max(F(i))$.

We use the time to successful rendezvous as a measure of the algorithm's success. The length of the sub-sequences until rendezvous is used as a measure of time until successful rendezvous. Again, without noise, the deterministic algorithms (sequential and smart-sequential) are guaranteed to generate sequences of length one, that is, meet on the first try. By generating a sequence for each algorithm under different conditions, (varying δ, the asynchrony j, and the landmark set commonality d), we can measure the time to rendezvous under the various conditions.

It should be noted that the parameter space of this problem is substantial, and therefore not all aspects of the problem were explored. Only the more relevant and interesting aspects of the problem space are presented here.

5.2. Experimental Results

Each trial determined the number of rendezvous attempts, or iterations, needed to achieve rendezvous under different conditions. Measurements were taken at 14 values of δ, where each measurement was made 1000 times; these 1000 trials gave a mean number of iterations to rendezvous for a particular algorithm and a particular value of δ.

5.3. Base case: Time as a function of Noise

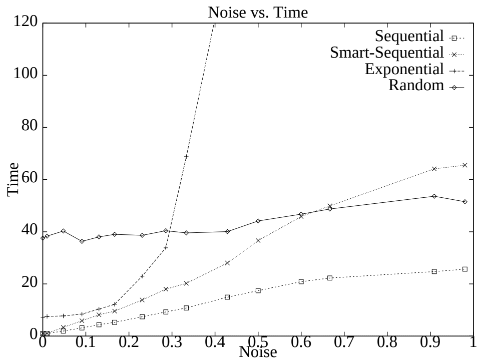


Figure 2. Baseline Performance - Time to Rendezvous as a function of Noise-level δ

The baseline simulation shows the performance of four algorithms in the face of increasing noise. The size of the landmark set is 50 landmarks, asynchrony j is 0, and the landmark sets have 100% commonality. The four algorithms are the deterministic sequential and smart-sequential algorithms, and weighted probabilistic distributions with exponential and linear probability functions. Figure 2 shows that the sequential algorithm is the best performer, especially in the face of high noise (i.e., $\delta > 0.2$) , which concurs with the analytical result. Clearly, exponential is a very fragile rendezvous scheduling function, failing catastrophically with noise, $\delta > 0.2$.

5.4. 50 % Asynchrony

In the face of asynchrony, however, the algorithms exhibit less intuitive behaviour. Asynchrony, again, is the probability that a particular rendezvous at a mutually agreed place and time actually occurs. The simulation (which created landmark sequences) implemented asynchrony as the probability that a particular sequence element could be used. Even if the pair of landmark sequences contained the same landmark at identical positions, the sequence may not have terminated there, because the asynchrony probability prevented the first pair of matching landmarks in sequence from being compared, as if the robots had failed to rendezvous successfully despite attempting to do so at the same location at the same time.

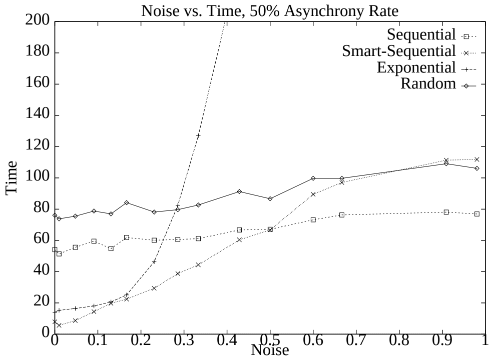


Figure 3. Performance with 50% Asynchrony rate

Figure 3 shows the performance of the algorithms given a 50% asynchrony rate, or a 50% probability of successfully making a rendezvous. In this case, the smart-sequential and exponential algorithms out-perform the sequential strategy, because the sequential form suffers from having to visit every other landmark before being able to return to the landmark that failed on a particular iteration, whereas the other two algorithms can return to landmarks relatively quickly. However, once noise dominates the values, ($\delta > 0.5$) the sequential algorithm outperforms the other algorithms because it does not rely heavily on particular landmark values — it is not returning to the same landmark over and over again.

5.5. 80 % Asynchrony

Even more interesting in the case of very high (80%) asynchrony, Figure 4 shows that the exponential probabilistic function outperforms the

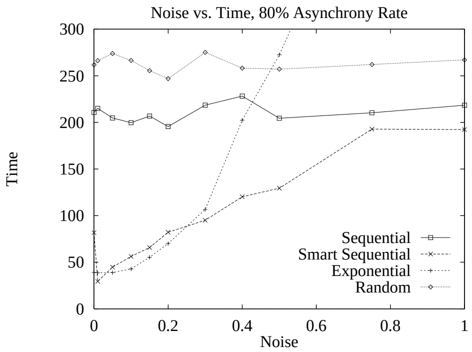


Figure 4. Performance with 80% Asynchrony rate

deterministic algorithms in the face of low noise ($0.5 < \delta < 0.25$), but again fails rapidly in the case of high noise ($\delta > 0.25$). The exponential algorithm essentially forces the robot to return to the same landmark over and over again, which is the correct strategy when asynchrony is high. However, when noise is high, the odds that the recurrent landmark is the wrong one increase, and the deterministic algorithms, which do not return to the same landmark as often, perform better.

5.6. 75 % Landmark Commonality

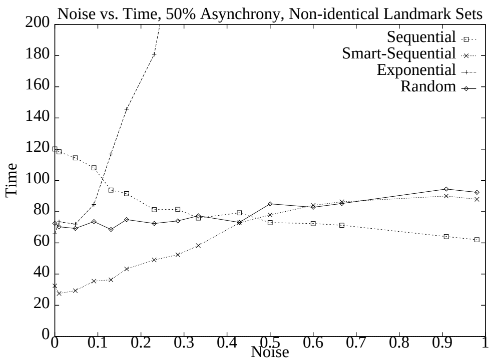


Figure 5. Performance with non-identical landmark sets, and 50% Asynchrony rate

Finally, Figure 5 shows performance for maps with only 75% of the landmarks in common and 50% asynchrony. The performance with non-identical landmark sets (akin to non-isomorphic maps) is very similar to performance under low- to medium-asynchrony. The smart-sequential algorithm performs better with low noise because it can return to landmarks faster than sequential, but in the case of high noise ($\delta > 0.35$), returning to landmarks too frequently can be costly, and the sequential algorithm again dominates.

5.7. Experimental conclusions

The sequential strategy is simple and relatively immune to sensor noise because it does not rely heavily on the relative rankings of landmarks. However, it is sensitive to asynchrony. If the two robots have the same ordered landmark sets but suffer from synchronisation problems and hence miss meetings at commonly-selected landmark, n rendezvous attempts must occur before an identical pair of landmarks occurs in the visit sequence.

Smart-sequential has its domain of superiority where the agent differences (e.g. noise) are low, but not negligible, or where the landmark sets are not identical. Although it is not a probabilistic strategy as such, it essentially groups landmarks together into types of high probability through low probability, in attempt to "guess" where the other agent(s) might be. Smart-sequential also does not perform well under conditions of high noise or high asynchrony. It suffers under conditions of high noise, because it relies upon a reasonable, if not 100% accurate knowledge of the distinctiveness surface; as noise destroys the accuracy of that measurement process, the estimates based on that knowledge become poorer.

The exponential algorithm proves to have a surprising domain of superiority in the low-noise, high-asynchrony case. When a rendezvous fails due to simply a missed attempt, the optimal behaviour is to retry the attempt regularly; it is this behaviour the exponential algorithm excels at.

6. Physically-based Simulation

Although the numerical analysis of section 5 encapsulated a number of practical issues with parameters such as sensor noise, our analysis did not address the problems of space. We therefore next simulate mobile robots in a two-dimensional environment. These experiments have two goals; the first is to determine the behaviour of the various rendezvous algorithms under different experimental conditions. The second goal is to determine the speed-up of the exploration of two robots performing rendezvous, when compared with a single robot.

6.1. Experimental Method

Figure 6 shows the map used for these experiments The first test suite, the baseline algorithm performance. In each set of 25 trials, one agent was started at the same point every trial, the **A** in Figure 6, and the other agent was put at one of 5 locations, the circles labelled **1-5** in Figure 6, for 5 trials per location. The trials were conducted for 15 values of δ.

The agents are modelled as idealised Nomad 200 robots with perfect (noise-free) sensing abilities and odometry.[2] The agents explore the unknown environment for a pre-determined length of time; at the end

[2] While we did have the ability to simulate the sonars using a more realistic sonar simulator, it was not exploited in these experiments.

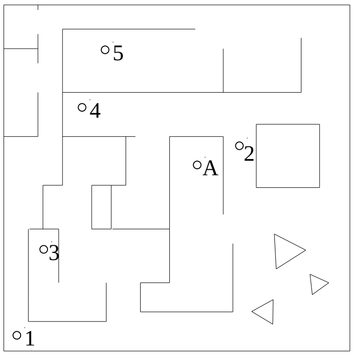


Figure 6. The map for the simulated experiments, with the starting positions marked as circles. One agent always started from the position labelled 'A', whereas the other agent started 5 times at each of the positions marked with numbers.

of this length of time the agents attempt rendezvous. The agents then take the n best landmarks seen so far, and use these for the rendezvous algorithm. Each agent is running the same rendezvous algorithm; where the algorithms demand asymmetrical agents, the agents are assigned roles randomly *ab initio*.

However, this simplistic description hides several complex issues, the first of which is choosing an appropriate distinctiveness function.

6.1.1. *The Distinctiveness Function*

Recall from section 3 that we would like a distinctiveness function that is smooth, robust, and which has few local extrema over the exploration space. Our choice of a distinctiveness function in the following experiments was inspired by human experience; we would like the function to peak in wide-open areas that correspond to large rooms, foyers, etc.

We can measure the "openness", $\mathcal{R}$, of any point in the environment simply by summing the range returned by each sensor:

$$\mathcal{R}(x,y) = \sum_{i=1}^{n} R(x,y,\theta_i) \tag{13}$$

We measure the asymmetry, $\mathcal{A}$, of each point by summing the absolute difference of diametrically-opposed pairs of sensors. If each pair of sensors measures the same, then the asymmetry falls to 0.

$$\mathcal{A}(x,y) = \sum_{i=1}^{n/2} \mid R(x,y,\theta_i) - R(x,y,\theta_{i+n/2}) \mid \tag{14}$$

Combining the openness and the asymmetry gives

$$D \;=\; \frac{\sum_{n=1}^{64} R_i}{\sum_{n=1}^{32} \mid R_i - R_{i+32} \mid} \tag{15}$$

6.1.2. *Landmark Distributions in Space*

Ideally, only landmarks which are not mutually visible should be kept in the landmark set, otherwise two landmarks (which are in reality distinctiveness maxima along the trajectory) may in fact be very proximal to one another. While this does not in principle break the rendezvous process, if the environment is large, or the area of the environment common to the agents is relatively small, then the time to rendezvous may become unreasonably large. Since the goal is to have the agents rendezvous in minimum time, it is undesirable for the agents to spend time visiting points in the environment that are close together.

There are a number of ways of dealing with this problem, for example using a sensor to test line-of-sight, or actually travelling between landmarks to test if the line-of-sight path is clear. Since the task that we are performing is exploration and an occupancy grid map is available, we use this map to test for mutual visibility.

6.1.3. *Accurate Peak Measurement*

The method described in section 6.1.2 suffices for eliminating multiple landmarks that are associated with the same structure in the distinctiveness surface. However, there still remain the issues of accurately recognising the distinctiveness peaks, and even more importantly, measuring the peak height accurately.

If the agents share the same trajectories through the environment, then this issue simply one of sensor differences. Such a situation would occur, for instance, if the agents were employing Voronoi diagrams or freeway methods for navigation. While there will in practice be some positional error across agents, this will largely be due to sensor error and can be encapsulated in the sensor model. However, if the primary task does not involve navigation along mandated trajectories, then it is likely that the agents will, while capturing the same peaks in the distinctiveness, have very different perceptions of the height of the peaks, as Figure 7 demonstrates. [3]

In practice, the measurement of landmarks can be refined by performing gradient ascent over the distinctiveness surface at each potential landmark. This could be done during the landmark acquisition process, but no longer decouples the primary task (e.g. exploration) from rendezvous. The agents could re-visit each landmark before rendezvous,

[3] This is, of course, a sampling problem. However, given the prevalence of high-frequency information in the distinctiveness surface, undersampling is inevitable without serious increases in mechanical complexity. In the worst case scenario, if the agents drastically undersample the distinctiveness surface, they will not only mis-measure the distinctiveness peaks, but miss some peaks altogether. If the distinctiveness function is also used for the primary task (as it is in this research, as the sonar is used both for the distinctiveness measurements as well as generating the map), the primary task must be aware that rendezvous is being performed, and must be willing to relinquish control of its sensors to the landmark acquisition process. This requires some coupling between the landmark acquisition process and the primary task, but the coupling can be eliminated if necessary by giving the rendezvous process a separate sensor.

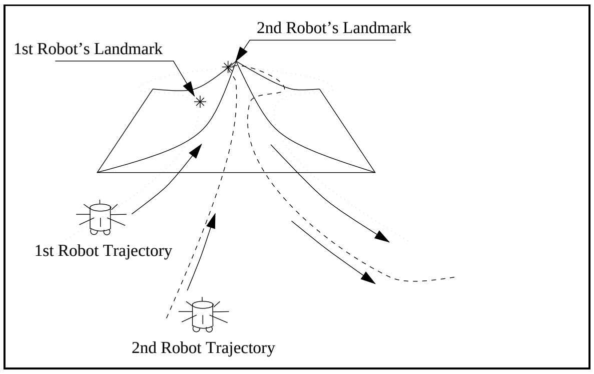


Figure 7. Two agents exploring the distinctiveness surface. Because of the nature of the exploration algorithm, one agent passes directly over top of the peak, and thus measures its height correctly. The other agent passes first to one side, and then the other, retaining only the higher of the two maximal measurements, never measuring the peak correctly at its maximum height.

but this would add a mechanical complexity to the rendezvous process. For an environment that has many, widely separated landmarks, this will be unacceptable.

The method chosen for the simulation and real robot experiments is to refine the landmarks during the rendezvous process. This has the advantage that the rendezvous process and primary task are decoupled as in the previous method, but the additional mechanical complexity is low, as in the first method. The disadvantage is that the visit sequence must be recomputed in the majority of cases if a deterministic algorithm is being used. Furthermore, if the measurements are completely wrong, the measurement may not be corrected until after a substantial number of iterations.

Figure 8 shows the result of the hill-climbing operation.

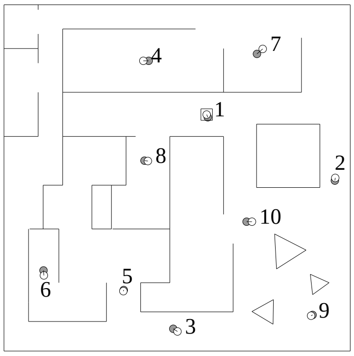


Figure 8. The result of the landmark refinement process. The grey filled circles are the initial peak estimates, acquired during the exploration process. The white circles are the final positions of the landmarks. The box is the best landmark.

6.2. Modelling Noise, Landmark Commonality and Asynchrony

In the experiments using the simulation of robot exploration and rendezvous, a noise term $\eta(x, y)$ was added to every point in space using the same model as in 5. Modelling the landmark commonality, d, and asynchrony, j, explicitly as in that section was impractical. The landmark commonality parameter is a reflection of the degree to which the trajectories of the agents overlap; this parameter is a function of the trajectories, not the inverse. Similarly, the asynchrony is a function of environmental and robot characteristics; it is extremely difficult to extract the appropriate characteristics from the single parameter.

However, the simulation did model these characteristics indirectly. The landmark commonality parameter was set by altering the size of the bounded world, and altering the time allowed between rendezvous attempts. Asynchrony was modelled using a radio communication simulator. The simulator had a locking mechanism that prevented the robots from moving to the next landmark, until both had made a communication request. By allowing the locking mechanism to operate probabilistically, the parameter j could be included in the simulation.

6.2.1. *Simulating Rendezvous*

The simulation of detecting other robots and achieving rendezvous was implemented using simulated radio communication. Requests were made by each robot to the radio simulator, and the simulator then determined, based on its knowledge of the complete map and the current positions of the two simulated robots within the map, whether or not the robots were mutually visible (line-of-sight), and whether they were in radio range of one another (13.5 m [4]).

6.3. Experimental Results

There were three main experiments performed on the simulated exploration and rendezvous, and each was a variant of a test of the rendezvous algorithm performance vs. noise. Each data point is the average of 25 trials. The trial was terminated at 100 rendezvous attempts if the agents had not achieved rendezvous by then.

6.3.1. *Baseline Performance*

Figure 9 demonstrates the performance of the 4 main algorithms, in the face of increasing noise. The size of the landmark set is 10 landmarks and asynchrony j is 0. In order to have the agents have as close to 100% landmark commonality as possible, the simulation explored for 600 seconds - this proved to be sufficient for the agents to have explored almost

[4] This number for the radio range was based on the radius of the smallest robot we used, the RWI B12. 13.5m is one hundred B12 diameters.

all of the space. The four algorithms are sequential, smart-sequential and the probabilistic functions exponential and random.

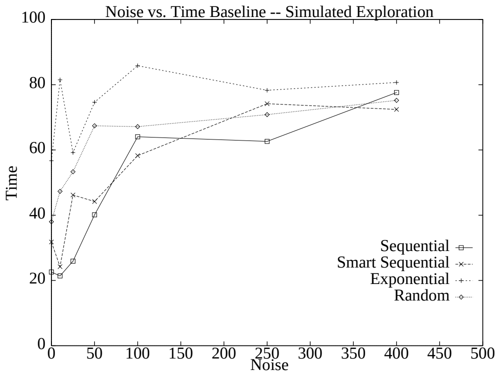


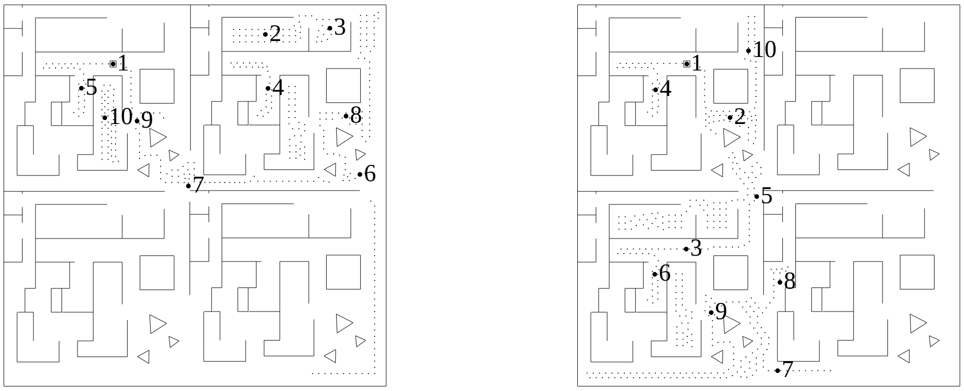
Figure 9. Baseline Performance - Time to Rendezvous as a function of Noise-level $\delta = [0, 500]$.

At the highest noise level in Figure 9, the noise is 80% of the highest noise-free peak in the environment; however, certain algorithmic characteristics manifest themselves. For example, sequential continues to out-perform all other algorithms.

6.3.2. *Disjoint Exploration Areas*

This is the second of the three experiments performed using the simulated exploration and rendezvous for the explicit case of disjoint landmark sets, representing areas of the environment explored by only one agent.

Figure 10 shows an example of the exploration carried out by two agents in this environment. Clearly, the two agents have explored the majority of the environment, and yet the overlapping areas of their trajectories is fairly minimal. This is the first experiment where the speed-up of the algorithms can be tested; the results of the speed-up of the algorithms will be in section 6.4.



a. First agent's trajectory b. Second agent's trajectory

Figure 10. Two example trajectories through a larger space. The circles indicate landmarks. Notice that the rendezvous occurred successfully, even though a large part of the trajectories were unique to the agent.

Figure 11 demonstrates the performance of the rendezvous algorithms in the face of both increasing noise and incomplete exploration. The size of the landmark set is 10 landmarks and asynchrony j is 0.

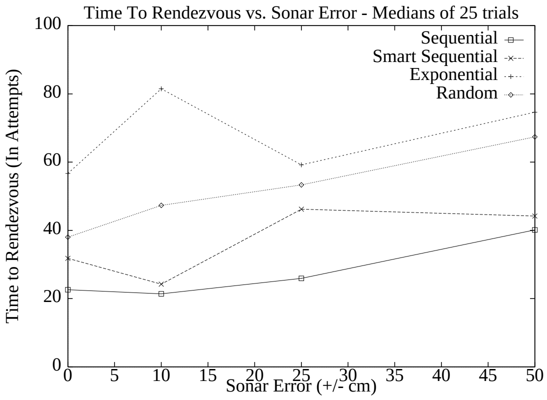


Figure 11. Non-Identical Landmark sets - Time to Rendezvous as a function of Noise-level, Low Noise $\delta = [0, 50]$.

Notice that smart-sequential is no longer the best algorithm, even in this low noise region of the parameter space. The ability of the smart-sequential algorithm to guess the location of the other agent is damaged by the incomplete knowledge that results from disjoint landmark sets.

6.3.3. *Asynchrony*

In this final experiment, we tested the ability of the agents to rendezvous under conditions where the robots would sometimes fail to meet successfully, even when at the same location. We are particularly interested in the low-noise region of the parameter space, as the numerical analysis indicated that the exponential algorithms performed best under these conditions. As Figure 12 indicates, the superior performance of the stochastic algorithms is present in the spatial simulation.

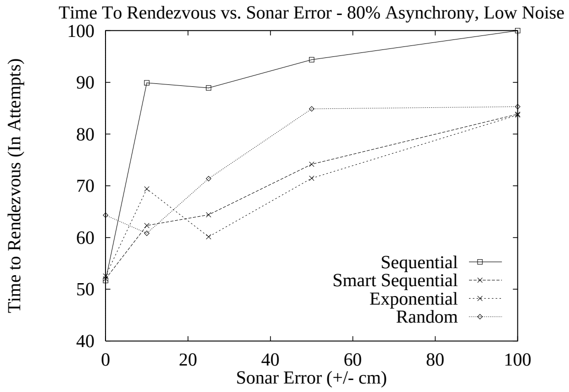


Figure 12. 80% Asynchrony - Time to Rendezvous as a function of Noise-level, Noise $\delta = [0, 100]$.

Focussing further on the region where δ is small, the exponential algorithm should be the fastest is expected. The exploration suffers only from missed meetings - both agents should have chosen the same landmarks. Since this algorithm will revisit the best landmark more often than any other algorithm, it has the best chance of overcoming the asynchrony problem. However, once any noise is present in the system, this algorithm fails rapidly.

6.4. Multi-Agent Exploration

Of particular interest in this 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. We would like to demonstrate a significant increase in exploration speed, even accounting for the time to rendezvous.

As our metric for measuring speed increase in exploration, we used the change in mapping speed, $S = \frac{A}{T}$, where A is the percentage of the environment that has been mapped, and T is the time to complete the mapping.

Since the experiment was constructed so that the occupancy grid matched the size of the bounded environment, we use the number of cells in the occupancy grid that contained information of any kind (occupied or not) as our measure of the size of the mapped environment.

The increase in speed of the mapping process is then given by Equation 18,

$$\Delta S = \frac{S_{combined} - S_{single}}{S_{single}} \tag{16}$$

$$= \frac{\frac{A_c}{T_c} - \frac{A_s}{T_s}}{\frac{A_s}{T_c}} \tag{17}$$

$$= \frac{A_c}{A_s}\frac{T_s}{T_c} - 1 \tag{18}$$

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

Once the maps from the two agents are merged, it is then possible to determine how much of the environment was explored by the two agents together, giving the increase in explored speed, compared to the efforts of a single agent. Recall that each data point in the preceding graphs represents the mean of 25 trials. The increase in explored areas over all 25 trials was a minimum of 42.8%, and on average 49.4%. If the agents were capable of merging their maps immediately after the exploration phase, then $T_c = T_s$, and the increase in area is exactly equal to the increase in speed. However, this ideal situation is equivalent to total communication, and is not realistic.

There are two possible ways to interpret the exploration speed results: the first treats each exploration iteration and rendezvous iteration as a single time increment, as if travelling through a graph where each arc is of time-length 1, and T_r is simply the number of rendezvous iterations.

Table II shows the speed increase in the algorithms in the zero-noise case, using this graph-like model of the exploration process. Each datum is the average of 25 trials; if the agents failed to meet (e.g. due to the exponential algorithm), then the change in mapping speed, ΔS was set to 1.0.

Table II. The speed increase using the graph-like model of the world, in the zero-noise case. Each number is the average of 25 trials.

Algorithm	% Speed Increase
Sequential	49.1 %
Smart-Sequential	38.1 %
Exponential	21.1 %
Random	46.7 %

Only the exploration speed of the exponential algorithm was seriously degraded by the rendezvous process. Figure 13 shows the change in exploration speed as the noise is increased.

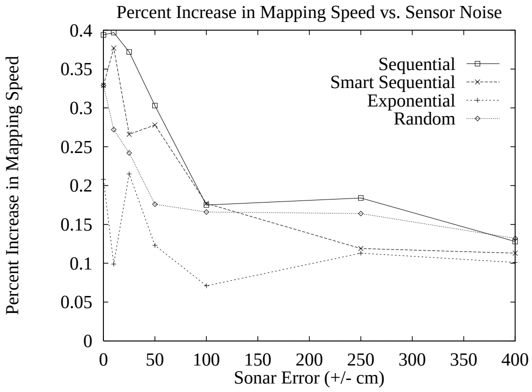


Figure 13. Increase in exploration speed as a function of noise. Environment modelled as a graph.

Characteristically, sequential performed extremely well over the majority of the noise range; smart-sequential did well in the low-noise range, however once the noise began to dominate the measurements, smart-sequential's performance was considerably degraded. These results reassuringly corroborate on a general level the numerical and simulation results. In fact, figure 13 is compelling support for multiple-agent robotics in general; an increase of speed of up to 50% in the exploration task is still available. It is a problem for future work to show that this increase in speed is possible in general.

7. Rendezvous using Real Robots

All of the prior simulation experiments assumed the simulation sensors were ideal; noise was explicitly applied in order to approximate real sensors. Odometric error was assumed to be negligible. Issues of path-planning were simplified to allow the robots to pass through each other in space, rather than investing time in allowing the simulated agents to detect each other during the exploration stage. These are all assumptions that are not valid once a real robot is being used. We therefore present a proof of concept, that, in fact, the rendezvous method is possible and useful on real robots.

7.1. Experimental Method

In this final experiment, we examine the feasibility of our rendezvous strategy on a pair of actual robots in our laboratory. The experiment was conducted using two mobile robots, a Nomad 200 and an RWI B-12. Both robots are essentially cylindrical, and quasi-holonomic, in that they are capable of turning with $0°$ radius. The Nomad 200 is 50cm in diameter, and has 16 sonar transducers equally separated by $22.5°$. The RWI B-12 is 27cm in diameter, and has 12 sonar transducers equally separated by $30°$. Although the Nomad 200 has an onboard 486 processor running Linux, all computation was performed off-board, on two SGI Indigo platforms, and a Pentium platform. The communication between the robots and their controlling platforms was wireless.

Figure 14 show the robots moving through the maze in the laboratory. The right panel of the figure shows the robots standing next to each other, having made a successfully rendezvous.

The experiment was held in a laboratory space measuring 550cm by 840cm. The walls were free-standing corrugated plastic, 60cm high, taped together for structural integrity, and stood off the floor with angle-brackets, measuring 10cm long. The total wall length, including bounding walls, was 50.4m.

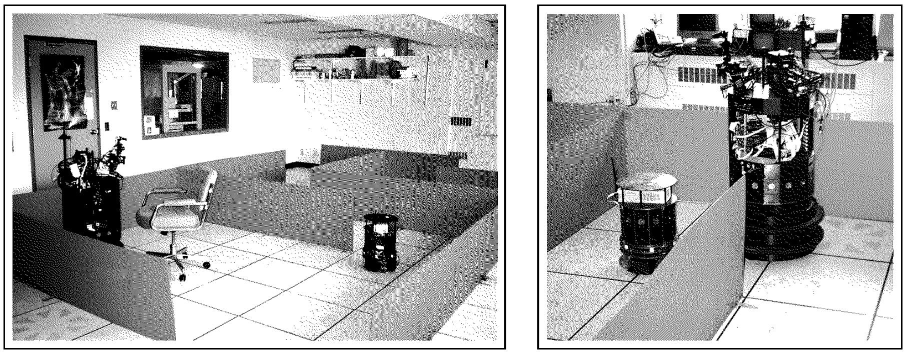


Figure 14. The robots exploring the maze, and then making a rendezvous.

7.1.1. *Sonar sensors*

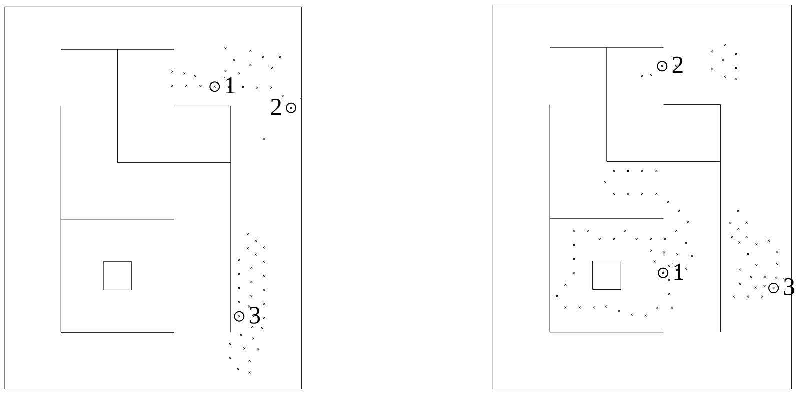
The sensor that was used throughout these experiments was the sonar sensor, which is a range sensor only. Consequently, all our distinctiveness function candidates relied upon range information only. The maximum range of the robots[5] is 8m for the Nomad, and 13m for the RWI. The range precision is ± 2.54cm for the Nomad, and ± 1.07cm for the RWI. By using the sonar to measure the distance to obstacles around it, the robot can acquire a metric map of its environment. There do exist more sophisticated sonar models such as developed by Kleeman and Kuc (Kleeman and Kuc, 1994), Wilkes (Wilkes et al., 1991), Borenstein (Borenstein et al., 1996) and Lacroix and Dudek (Lacroix and Dudek, 1997) that can recognise and deal appropriately with sonar artefacts in our model. However, our simple model of the sonar pulses, combined with some simple outlier handing, is sufficient for the limited purposes of our experiments; a more sophisticated sonar model would be more appropriate for long-term exploration and environment modelling.

[5] Assuming speed of sound at 330 m/s.

7.2. Experimental Results

7.2.1. *Trajectory*

Figure 15 shows the trajectories of the robots moving through the maze. The RWI B-12's trajectory is shown in the left panel, and the Nomad 200's trajectory is shown in the right panel. It should be emphasised that the maps were overlaid by hand for clarity, and the robots had no embedded knowledge of the layout of environment. Also overlaid on the images are the landmark positions that were chosen by the robots for rendezvous.



a. Nomad 200 trajectory   b. RWI B-12 trajectory

Figure 15. The landmark selections of the two robots overlaid on their trajectories. The triangles represent points where the robots considered potential landmarks. The ranking of the landmarks in the final landmark set is shown as well.

The trajectories consist of collections of points, separated by large areas of space. These "islands" of points were areas of the environment explored using local potential field descent. Once a local potential minimum had been reached, the robot used breadth-first search to find a new area that was known to be clear, yet low in potential (i.e., seen but unexplored).

Although gradient ascent was used in the simulations, it was not used in these experiments due to the small size of the environment.

Notice that the Nomad chose a point in the upper corridor as its best rendezvous location, whereas the RWI chose a point in the inner maze. This is no doubt due to sensor differences between the two robots.

7.2.2. *Rendezvous*

This single experiment provides the clearest support for our approach, in demonstrating a need for establishing some appropriate behaviour if the initial rendezvous attempt is unsuccessful. As Figure 15 indicates, the two robots did not choose the same point in the environment for the best rendezvous location. The robots made a successful rendezvous on the 4th attempt among the three landmarks, since they were using the sequential method of exploration.

Figure 16 shows the result of map merging. The map merging was performed manually. Although algorithms exist to merge maps gathered by heterogeneous agents (Ishioka et al., 1993), that problem is not the focus of the present work.

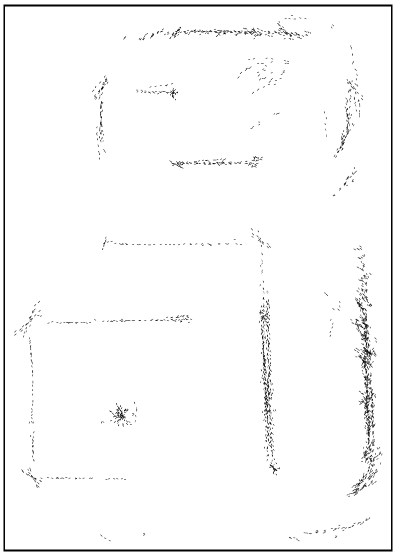


Figure 16. The final map created from the merged data acquired by the two robots.

The most important conclusion that was drawn from the experiment using the real robot is that the methodology we have chosen for achieving rendezvous is practical. The fact that the robots failed to meet on the first iteration of the rendezvous cycle is a very convincing piece of evidence that the rendezvous problem is substantially more complex than simply choosing a place to meet in the environment.

8. Conclusion

In this work, we have described the new problem of performing rendezvous between multiple mobile agents. The objective is to overcome practical communication limits by periodically having the agents converge and share information. In this manner, we increase the speed of operation of the multiple robot system compared to the single robot system, while eliminating the traditional assumption of infinite range, full bandwidth communication between agents. We are specifically interested in multiple-robot exploration of an unknown environment where communication is limited to short-range line of sight. Furthermore, we developed a methodology that does not depend on any particular task such as exploration, is trajectory independent and does not require any memory-intensive spatial representations. Although our implementation does take advantage of metric information that is provided by the exploration algorithm, our rendezvous methodology can be decoupled completely from the underlying primary task.

We divided the rendezvous problem into two separate subproblems. The first is determining what points in the environment constitute good rendezvous locations, or *landmarks*. We addressed this problem by modelling the environment as a function of the sensors; this function gave rise to a distinctiveness surface, defined over the domain of the environment. We then chose landmarks at the local extrema of the surface, limiting our knowledge of the surface only to those points that the agents have visited. Which points the robot visited was dictated by the trajectory prescribed by the underlying task, and so we demonstrated how to overcome these trajectory dependencies.

Our use of distinctive locations as landmarks is related to the psychology of human attentive vision and, in particular, to the selection of targets for pre-attentive vision. Although we have used only a sonar range-sensor throughout this work, it is easily extended to other sensor modalities, such as computer vision. For example, the notion of using distinctiveness to define domain-independent features has been employed for visual navigation (Sim and Dudek, 1998). Of course, the particular distinctiveness metric is a function of the sensor modality; the distinctiveness measure used in this paper is easily replaced with a camera-specific measure without loss of generality.

While the problem of rendezvous reduces, in the idealised case, only to the task of choosing the best point in the environment to which the robots should converge, this is in fact an inappropriate idealisation. In the formalisation of this problem, we identified 3 key parameters that characterise the problem. We showed that a number of different points in the environment must be chosen for meetings, and these points must be visited in some intelligent manner for rendezvous to be achieved reliably. These parameters we have called *sensor noise*, *map commonality*, and *asynchrony*.

This problem of which appropriate behaviour to use in choosing the landmarks to visit is the second of the two subproblems of rendezvous. We proposed two main classes of algorithms, *deterministic* and *probabilistic*, and gave examples of each class of algorithm. In order to determine the characteristics of the algorithms, we gave a closed-form analysis of the worst- and expected-case complexity of the algorithms at points in the parameter space. This closed-form analysis was complemented by a numerical description of the performance of the algorithms at a range of points in the parameter space.

Finally, we demonstrated the rendezvous algorithm in use both in simulation and on physical robots. The simulation tests were used as a confirmation of the numerical results. Within the class of deterministic algorithms, there were different regions that favoured different algorithms. These results were confirmed by both analytic closed-form solutions of section 3, and idealised numerical simulations of section 5. The physical experiments served as a proof of concept for the exploration and rendezvous algorithms, and we concluded with a map of an environment that resulted from the collaborative exploration and subsequent successful rendezvous within our laboratory of two robots.

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 - under the correct circumstances, a heretofore poor choice of algorithm can outperform the erstwhile winner. The exponential algorithm, while generally a poor choice, will outperform the other algorithms when asynchrony is a problem but sensor noise is not. It may be, however, that determining the true operating conditions is sufficiently difficult that smart-sequential is usually the best choice. Further experiments in realistic robot situations is needed to be able to tell how difficult it is to determine the operating conditions.

The physically-based simulation demonstrated that, although it is much harder to isolate the parameters in a physical sense, many of the main conclusions were upheld, despite several complicating factors that were not part of the numerical simulation. Furthermore, the physically-based simulation demonstrated that an increase of speed is still attainable with a multiple robot system using the rendezvous approach to communication. The experiments using physical robots gave a compelling demonstration that the rendezvous algorithms are an essential part of the rendezvous process; the assumption that the robots will meet on the first iteration is simply untenable. Despite very similar sensors and configurations, and a high degree of overlap between the agents, the robots required 4 rendezvous iterations before they could successfully meet and share information.

Only a small number of rendezvous algorithms were considered for this work. There is a body of literature on online search methods, of which rendezvous is a subclass. Algorithms that were not considered here may have particular utility in different regions of the problem space.

Of the analysis presented in this work, only limited but critical parts of the parameter space were examined. Further examination is necessary for examining the behaviour of the algorithms under conditions of worsening noise, worsening asynchrony, and perhaps most importantly, conditions of landmark commonality. It is likely that as time passes, the areas explored by the agents will overlap more and more; analysis of the performance of the algorithms under these conditions would be useful.

One open problem is the ability of the agents to choose the appropriate rendezvous algorithm. A major part of this problem is allowing the agents to estimate the environmental parameters, and identify the correct portion of the parameter space that identifies the environment. At no time did the agents attempt to estimate the experimental parameters; the agents did not use any environmental information in the algorithms. Allowing the agent to vary the parameters, such as constants in the stochastic algorithms, as rendezvous succeeds or fails, may have considerable power. It may also be interesting to investigate stochastic estimates of performance for the rendezvous algorithms.

The only consideration used by the algorithms for choosing which landmark to visit was the distinctiveness of the landmark. Given the sometimes substantial mechanical complexity of travelling between two landmarks, a better algorithm would consider the mechanical complexity of visiting landmarks in addition to its distinctiveness, so that of two landmarks with similar distinctiveness, the closer landmark would be visited first.

Although we have dealt primarily with two-agent systems, the work is in principle easily extended to larger collections of agents, or swarms. The probabilistic algorithms are symmetric across agents, and therefore adding new agents is trivial. The deterministic algorithms can be extended to larger swarms, simply by dividing the swarms into pairs. However, implicit in large robot swarms are complex issues of task division and interference, and so it is no way clear that the same kind of speedups that are observed for two-agent collectives will be observed for larger collectives. There may also be intelligent ways to use larger collectives to avoid long-distance travel by transmitting information from agent to agent over several scheduled rendezvous. Further experiments are required to determine how information propagation affects the speed of task completion.

References

Alpern, S. (1995). The rendezvous search problem. *SIAM Journal of Control and Optimization*, 33(3):673–683.

Alpern, S. and Shmuel, G. (1995). Rendezvous search on the line with distinguishable players. *SIAM Journal on Control and Optimization*, 33:1270–1276.

Anderson, E. J. and Essegaier, S. (1995). Rendezvous search on the line with indistinguishable players. *SIAM Journal on Control and Optimization*, 33:1637–1642.

Balch, T. and Arkin, R. C. (1994). Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52.

Balch, T. R. and Arkin, R. C. (1998). Behavior-based formation control for multiagent robot teams. *IEEE Transactions on Robotics and Automation*.

Beni, G. and Liang, P. (1996). Pattern reconfiguration in swarms - convergence of a distributed asynchronous and bounded iterative algorithm. *IEEE Transactions on Robotics and Automation*, 12(3):485–490.

Borenstein, J., Everett, H., Feng, L., and Wehe, D. (1996). Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems, Special Issue on Mobile Robots*, 14(4):231–249.

Brumitt, B. and Stentz, A. (1996). Dynamic mission planning for multiple mobile robots. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2396–2401, Minneapolis, MN.

Chatila, R. and Laumond, J. (1985). Position referencing and consistent world modelling for mobile robots. In *Proc. IEEE International Conference on Robotics and Automation*, pages 138–170.

Cohen, W. W. (1996). Adaptive mapping and navigation by teams of simple robots. *Robotics and Autonomous Systems*, (18):411–434.

Donald, B. (1995). On information invariant in robotics. *Artificial Intelligence*, (72):217–304.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1991). Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1996). A taxonomy for multi-agent robotics. *Autonomous Robots*, 3:375–397.

Hara, F. et al. (1992). Effects of population size in multi-robots cooperative behaviors. In *Proc. International Symposium on Distributed Autonomous Robotic Systems*, pages 3–9.

Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3).

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

Kleeman, L. and Kuc, R. (1994). An optimal sonar array for traget localization and classification. In *Proc. IEEE International Conference Robotics and Automation*, pages 3130–3135, San Diego.

Kuipers, B. and Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, pages 47–63.

Lacroix, S. and Dudek, G. (1997). On the identification of sonar features. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Grenoble, France.

Mataric, M. (1992). Minimizing complexity in controlling a mobile robot population. In *Proc. IEEE International Conference on Robotics and Automation*, pages 830–835.

Moravec, H. P. and Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proc. IEEE International Conference on Robotics and Automation*, pages 116–121.

Parker, L. E. (1994). *Heterogeneous Multi-Robot Cooperation.* PhD thesis, Massachusetts Institute of Technology.

Rao, N., Protopopescu, V., and Manickam, V. (1996). Cooperative terrain model acquisition by a team of two or three point-robots. In *Proc. IEEE Inter. Conf. Robotics and Automation*, volume 2, pages 1427–1433.

Rekleitis, I. M., Dudek, G., and Milios, E. E. (1997). Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proc. of the 15th International Joint Conference on Artificial Intelligence.*

Shatkay, H. and Kaelbling, L. (1997). Learning topological maps with weak local odometric information. In *Proc. International Joint Conferences on Artificial Intelligence.*

Sim, R. and Dudek, G. (1998). Mobile robot localization from learned landmarks. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, BC.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *AI Journal*, 99(1):21–71.

Thrun, S. and Buecken, A. (1996). Integrating grid-based and topological maps for mobile robot navigation. In *Proc. 13th National Conference on Artificial Intelligence*, Portland, Oregon.

Wilkes, D., Dudek, G., Jenkin, M., and Milios, E. (1991). Simulation of sonar mapping in complex environments using multiple reflecting surfaces. In *Proceedings of Vision Interface 1991*, Calgary, AB.