Autonomous Robots, ?, 5–29 (19??)
© 19?? Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

A Taxonomy for Multi-Agent Robotics*

GREGORY DUDEK
dudek@cim.mcgill.edu
Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada

MICHAEL R. M. JENKIN
jenkin@cs.yorku.ca
Department of Computer Science, York University, North York, Ontario, Canada

EVANGELOS MILIOS
eem@cs.yorku.ca
Department of Computer Science, York University, North York, Ontario, Canada

DAVID WILKES
wilkes@vis.toronto.edu
Department of Computer Science, York University, North York, Ontario, Canada

Received ?. Revised ?.

Editors: ?

Abstract. A key difficulty in the design of multi-agent robotic systems is the size and complexity of the space of possible designs. In order to make principled design decisions, an understanding of the many possible system configurations is essential. To this end, we present a taxonomy that classifies multi-agent systems according to communication, computational and other capabilities. We survey existing efforts involving multi-agent systems according to their positions in the taxonomy. We also present additional results concerning multi-agent systems, with the dual purposes of illustrating the usefulness of the taxonomy in simplifying discourse about robot collective properties, and also demonstrating that a collective can be demonstrably more powerful than a single unit of the collective.

Keywords: Mobile robotics, robotic collectives.

1. Introduction and Motivation

Task oriented behaviour by groups of agents is ubiquitous in nature. How and why should multiple mobile robots be used for a task? Although most mobile robotic systems involve a single robot operating alone in its environment, a number of researchers have considered the problems and potential advantages involved in having an environment inhabited by a group of robots which cooperate in order to complete some required task. For some specific robotic tasks, such as exploring an unknown planet, pushing objects (Parker, 1994b; Mataric *et al.*, 1995; Rus *et al.*, 1995), or cleaning up toxic waste, it has been suggested that rather than sending one very complex robot to perform the task it would more effective to send a number of smaller, simpler robots. Such a collection of robots is sometimes described as a *swarm* (Beni and Wang, 1989), a *colony*, or as a *collective*, or the robots may be said to exhibit *cooperative behaviour* (Parker, 1993). Using multiple robots rather than a single robot can have several advantages and leads to a variety of design tradeoffs. Collectives of simple robots may be simpler in terms of individual physical design than a larger, more complex robot, and thus the resulting system can be more economical, more scalable and less susceptible to overall failure.

* The support of NSERC Canada is greatfully acknowledged.

There is a continuum of possible collective designs. A collective might consist of a collection of completely autonomous agents which only communicate by pairwise transfer of information. A collective might consist of a number of remotely controlled appendages so that the entire collective might more properly be described as a single large robot with distributed actuators. Both of these extremes exist in the literature. Although this later extreme might be considered as a collective, the more interesting case occurs when the elements of the collective lack any functionally relevant, permanent physical connectivity. We thus distinguish between a single, complex and possibly distributed robot R and a collective of robots $\{r_i\}$ which lack a functionally relevant, permanent physical connection.

Collectives offer the possibility of enhanced task performance, increased task reliability and decreased cost over more traditional robotic systems. Although they have this potential, many possible collective designs are neither more efficient, nor more reliable, nor more robust than a comparable single (more complex) robot. In order for a collective to have these advantages the collective must be designed with these issues in mind.

In addition to having these properties, it is essential that the collective have a collective behaviour or set of actions that accomplishes the same behaviour or action that was required of the single more complex robot. For a collective to exhibit cooperative intelligent behaviour, the members of the collective must be able to communicate with each other. This communication may take place directly via an explicit communication channel or indirectly through one robot sensing a change in other robots in its environment. Intra-collective communication presents difficulties in terms of collective efficiency, fault tolerance, and cost.

Interactions between natural organisms such as birds, ants, termites, wasps, primates, fish or wolves have been examined in the context of ethology (Tinbergen, 1951; Tinbergen, 1972; McFarland, 1989). Observations from biology and ethology have provided inspiration for developments ranging from subsumption architectures for single robots to inter-robot communication strategies for groups of robots (Anderson and Donath, 1991;Brooks, 1991). Canonical issues for biological groups include the maintenance of an appropriate distance between members of a school or flock, often via purely local communications (Partridge, 1982), or the communication of the location of a goal such as a food source. While the specific behaviours used by animals have been examined rigorously, the alternative design options for inter-agent communication has been less extensively examined.

Cao *et al.* (1995) identify Traffic Control, Box Pushing and Foraging as typical multiple robot tasks. Although these tasks have been addressed by robotic collectives, are they appropriate tasks for this type of approach? Some tasks seem ideally suited to multiple robotics. Gage (1992) identifies a number of military applications such as mine deployment, carrier deck foreign object disposal, etc., as potential applications for robotic collectives. These tasks are typified by the high potential for damage to individual collective elements, and thus it is the expendability of collective elements which is identified as the major reason for proposing robot collectives for the task, rather than any particular computational efficiency or reliability requirement. Although expendability is certainly a strong argument for collections of inexpensive robots over a single more complex expensive robot, are there computational reasons why a collective of robots should be preferred? Given a particular task T, which can be solved with either one very complex robot R, or with a collective of robots $\{r_i\}$, under what conditions should R be chosen over $\{r_i\}$? We consider several possible cases:

Tasks that ***require*** **multiple agents** Does there exist a task T which can be solved by $\{r_i\}$ but for which no R can be found? Consider the following (missile launch) example;

> There are two keys which are a large distance apart which must be turned at the same time.

Note that this task does not necessarily require multiple robots to solve it. If the keys are not too far apart then a single *large* robot can be used to solve the problem. If the keys can be turned within some small time interval of each other then a single *fast* robot R

can solve the problem. In order to exclude solutions which are based on a single robot R, the task must involve spatially separate tasks which require some sort of synchronization. This synchronization implies inter-robot communication. The robots must either have their clocks synchronized initially (which requires communication) and then plan to turn their keys at precisely the same time, or they must be able to communicate with each other in order to indicate that it is time to turn the key. A more prosaic example is a multi-robot scene exploration system that uses the motion of shadows in a scene to compute spatial occupancy. In this case, one robot uses a camera to examine the scene while a second robot moves a light source about the scene to cast appropriate shadows (Langer *et al.*, 1995).

Tasks that are traditionally multi-agent Many modern transportation, industrial, agricultural, and fishing related tasks are currently performed by a group of effectively autonomous agents. The tasks that they perform are typically parallelized with small amounts of coordinating communication at either the start (for truck delivery) or at the end (forestry). In these tasks each element of $\{r_i\}$ operates independently for the most part, utilizing inter-agent communication either initially, to parcel up the expected workload in an efficient manner, or penultimately, just before dealing with any work that was not covered during the parallel portion of the processing. From a robotic collective point of view, the computational processing is relatively straightforward due to the inherent parallelism of the tasks.

Elements of these collectives operate in effective ignorance of each other. Similar strategies have been proposed in robotic collectives work. For example, the *ignorant swarms* of Mataric (1992) and the *communication-less swarms* of Dudek *et al.* (1993) propose to solve simple, highly parallel tasks by having a number of robots solve a problem in parallel without communication. Although this approach may maximize reliability, it fails to maximize performance as members of the collective cannot be directed to uncompleted work which they cannot sense directly. If elements of the collective do not communicate at all then task completion can become probabilistic and while a probabilistic solution may be acceptable for some problems it is not in general. For some foraging or search tasks, such as finding lost children, a probabilistic solution is not appropriate and inter-robot communication must occur.

Tasks which are inherently single agent There exist tasks which do not benefit from the use of additional agents in order to solve them. Task and environment can combine to remove any benefit of the use of multiple agents. A single task at a single location does not benefit from the use of multiple robots, as a single robot is both necessary and sufficient.

Tasks that may benefit from the use of multiple agents Between these extremes exist tasks which could be performed faster, or more reliably with a collective $\{r_i\}$ rather than with a single robot R.

Consider the issue of speed. Perhaps the collective $\{r_i\}$ can perform a particular task faster than a single robot R. A typical task in this class is that of finding a particular object in a finite region. If there are n elements of $\{r_i\}$, then one should expect a speedup of at most n if we assume that each element of $\{r_i\}$ can do no more work per unit time than can R. Note that in order to obtain a speedup near n the work performed by each collective member must be well coordinated and each element of $\{r_i\}$ must have abilities near those of R. If this is not the case then there will be a loss of speedup as multiple robots will search the same area or individual elements of $\{r_i\}$ will search less efficiently. Once again, a high level of inter-robot communication is required.

It is also unlikely that individual elements of $\{r_i\}$ will be able to do the same amount of work per unit time that can be accomplished by R. Indeed, given a task T in which the only advantage of a collective is speed, then it might be worthwhile improving the performance of R, rather than constructing a reliable collective of $\{r_i\}$ to accomplish the same task.

Reliability (redundancy) is one performance measure for which collectives easily exhibit performance over that of a single robot. Failure of a single element of $\{r_i\}$ may not result in task failure. Failure of R guarantees task failure. What sort of design features should be included in $\{r_i\}$ so that the swarm exhibits reliability?

The communication mechanism utilized by the collective is critical to its practicality, efficiency and reliability. The need for effective communication is made quite clearly by Parker (1995) who performed various tasks with collectives whose members could and could not communicate with other collective members. She found that global awareness of the state of the collective members improves task efficiency. Rus *et al.* (1995) describes furniture moving experiments using centralized and distributed control, and also an experiment where communication takes place through the task itself.

Balch and Arkin (1994) have examined the extent to which various amounts of shared information facilitate certain simple multi-robot tasks. For example, if robots are grazing (i.e. consuming) some widely distributed resource to what extent is it helpful to have them explicitly transmit information on which regions have already been grazed? While it is not surprising that information facilitates certain tasks, the extent to which it does so (or sometimes fails to do so) must be carefully weighed against the additional cost of transmitting the information.

The requirements of practicality, efficiency and reliability are typically at odds with one another. Sophisticated inter-robot communication can maximize performance for many tasks, yet such communication requirements often leads to reduced reliability. If there are fixed communication topologies (e.g. Ueyama *et al.*, 1992) or controller robots (e.g. Hackwood and Beni, 1992), or other fragile communication mechanisms, then failure of these fixed links in the communications network will cause the entire collective to fail. In order to maximize the reliability of the collective, the communication mechanism between elements of $\{r_i\}$ must survive the worst possible destruction of collective elements. Communication, like action, should be distributed throughout the collective.

Many different collective architectures have been proposed. The behaviour based control strategy proposed by Brooks (1991) has become established as one possible approach for collections of simple independent robots, particularly for simple tasks. Other authors have considered how a collection of simple robots can be used to solve complex problems. Ueyama *et al.* (1992) propose a scheme whereby complex robots are organized in tree-like hierarchies with communication between robots limited to the structure of the hierarchy. Hackwood and Beni (1992) propose a model in which the robots are particularly simple but act under the influence of "signpost robots". These signposts can modify the internal state of the swarm units as they pass by. Under the action of the signposts, the entire swarm acts as a unit to carry out complex behaviours.

Mataric (1992) describes experiments with a homogeneous population of actual robots acting under different communication constraints. The robots either act in ignorance of one another, informed by one another, or intelligently (cooperating) with one another. As intra-collective communication improves, more and more complex behaviours are possible. In the limit, in which all of the robots have complete communication, then the robots can be considered as appendages of a single larger robot (or robotic "intelligence"). One major goal of many robotic collectives is to distribute not only the sensing (and possibly actions) of the robots, but also the intelligence. What sort of processing can be accomplished by a collection of robots that cannot be accomplished by a single one? What effects do limits on communications and unit processing capabilities have on the potential actions of the collective? How do we compare the structure of various possible collectives?

The information processing ability of a collective is dependent upon a large number of factors including the number of units, their sensing abilities, their communication mechanisms, etc. (Arkin and Hobbs, 1992; Nitz and Arkin, 1993). In order to understand more fully the properties of various designs of collectives, it is instructive to group collectives into classes and to determine the capabilities of each class. It may be the case that certain collective organizations have more potential processing ability than others, and that some collective organizations may be similar to existing parallel models of computation.

In this paper we consider alternative design dimensions for the communication and coordination of a group of cooperating mobile robots. Section 2 of the paper presents a taxonomy or design space to provide a common language of discourse for work on robot collectives. Following that, in Section 3 we present selected illustrative results concerning particular collectives and problems, with the two objectives of demonstrating the usefulness of the taxonomy in simplifying collective descriptions and demonstrating that a collective can be provably more powerful than a single unit of the collective in fundamental ways. Further, these results illustrate how different points in the design space can be related to one another, for example by showing that collectives made up of robots with rudimentary computational abilities can perform more complex computations. Section 4 describes illustrative experimental results with a pair of robots demonstrating some of the behaviours discussed previously. In Section 5 we draw some final conclusions.

2. A Taxonomy of Robot Collectives

Dudek *et al.* (1993, 1993c) and independently Cao *et al.* (1995) have proposed the classification of *swarm*, *collective* or *robot collaboration* research by defining a taxonomy or collection of axes. Cao *et al.* identify group architecture, conflict resolution strategy, origins of cooperation, learning and geometric problems as 'research axes' or taxonomic axes within which cooperative robots can be compared. These axes are highly interdependent and very broad making it difficult to identify isolated sample points within the taxonomy.

Yuta and Premvuti (1992) subdivided collectives based on the interactions of collective elements; do individual elements work towards a common objective or do they work independently. Arkin *et al.* (1993) also examined different collectives along several dimensions but only in terms of a particular task. The objective of each of these taxonomies is both to clarify the strengths, constraints and tradeoffs of various designs, and also to highlight various design alternatives. Whereas (Dudek *et al.*, 1993a) and (Dudek *et al.*, 1993c) concentrated on defining a taxonomy within which different robot collectives could be compared and contrasted, (Cao *et al.*, 1995) expands the axes of comparison to include learning and the geometric structure of the problem. Following Dudek *et al.* this paper concentrates on the more restrictive taxonomic comparison.

There are several natural dimensions along which robotic collectives can be naturally classified. These dimensions address the characteristics of the collective as a whole rather then the architectural characteristics of individual robots. The dimensions follow, with key points along each dimension noted with symbolic labels. Table 1 summarizes the axes of the taxonomy. Table 2 samples the current literature and places existing collectives within this taxonomy.

By size of the collective: The number of robots in the environment.

SIZE-ALONE 1 robot. The minimal collective

SIZE-PAIR 2 robots. The simplest group.

SIZE-LIM Multiple robots. The number n is small relative to the size of the task or environment.

SIZE-INF $n \gg 1$ robots. There is effectively an infinite number of robots.

Two robots can, of course, perform problems which are impossible with a single robot. Almost any operation involving simultaneity or near simultaneity of events (such as turning two keys at the same time), is impossible with a single spatially limited robot. Multiple robots can be used to obtain speedups in terms of task performance subject to robot task synchronization.

The distinction between SIZE-LIM and SIZE-INF is a property of the size of the task. A number of robot collectives assume that the number of robots available for the task is unbounded (SIZE-INF) and this provides a number of simplifications in terms of probabilistic task completion. As a simple example, consider the task of searching a bounded environment for a lost child or robot (this is known as the "find robbie" task). Provided that the collective is SIZE-INF then one algorithm is given by flood filling the environment with the robots. Eventually every location will either be filled with a robot (robbie wasn't there) or one of the robots will find robbie. Note that the robots do not have to communicate with each other to complete this task - they just need sufficient sensing in order to be able to determine if they have found robbie and to navigate to flood the environment. For any finite sized collective

(SIZE-LIM) this same algorithm is only probabilistic.

By communication range: In most systems there are limits to the range of direct communication for any single robot. This is a function both of the communications medium and the robot distribution. We list three key classes for this dimension.

COM-NONE Robots cannot communicate with other robots directly. It is possible for robots to communicate with each other *indirectly* by observing their presence, absence or behaviour (as many animals seem to). In order to have truly "ignorant robots" (Mataric, 1992), the robots must not only not communicate with each other they must not try to signal each other through behaviour.

COM-NEAR Robots can only communicate with other robots which are sufficiently nearby. This corresponds to the communication mechanism proposed by Hackwood and Beni (1992). Distance, in this context, can be interpreted either topologically or in a Euclidean sense. A limited communication distance can occur due to physical communication constraints. For example, the power of the communication signal is often limited not only for local design reasons, but also to allow non-overlapping use of the same channel (a scarce resource) by agents in different geographical areas.

COM-INF Robots can communicate with any other robot. This is a classical assumption, which is probably impractical if $n \gg 1$. The distinction between COM-NEAR and COM-INF is analogous to the distinction between SIZE-LIM and SIZE-INF. From a practical point of view, the collective may be considered to be COM-NEAR if the communication range is smaller than the maximum separation of the robots during their execution of the task of interest, and COM-INF if the communication range is greater than this maximum separation. We identify these two points in the communication range continuum to highlight the qualitative difference in the constraints imposed on the solution of a problem as the result of differences in communication range.

Note that we have deferred issues related to having multiple robots (autonomous agents) communicating (writing) to a single robot (memory location). This is a classic problem in parallel computation (Fich *et al.*, 1988). As minor modifications in the communication design of parallel machines can result in major changes in the power of the resulting machine (Boas, 1989), we also partition the taxonomy by considering the topology of the inter-unit communication strategy utilized by the collective.

Cao *et al.* (1995) identify the communication structure as one of their taxonomic axes, and identify interaction via the environment, sensing and communications as three critical structures. The taxonomy presented here provides a finer granularity in terms of communications in order to highlight the importance of different communication strategies on the overall capability of the collective.

By communication topology: Robots may not be able to communicate with an arbitrary element of the collective regardless of its proximity. Robots may only be allowed to communicate within a particularly hierarchy (Ueyama *et al.*, 1992), or with specific controller robots (Hackwood and Beni, 1992). Individual robots may have names and messages may be sent to them directly, or messages may be broadcast to all robots. Some key variations are:

TOP-BROAD Broadcast. Every robot can communicate with all of the other robots. It is not possible to send a message only to a particular element of the collective.

TOP-ADD Address. Every robot can communicate with any arbitrary robot by name or address.

TOP-TREE Tree. Robots are linked in a tree and may only communicate through this hierarchy. This communication topology is utilized in systems with controlling robots or supervisors such as in the FIRST system (Causse and Pampagnin, 1995).

TOP-GRAPH Robots are linked in a general graph. This is a more general connectivity scheme than the tree and is more robust since redundant links can prevent the entire collective from becoming disconnected.

Communication strategies based on either tree-like or address based communication topologies are likely to be highly sensitive to failure of particular robots in the collective. Failure of a particular robot will isolate robots on either side of the failed node in the hierarchy. Addressing implies distinctive roles for individuals; resulting in reduced interchangeability, unless the robots' roles change dynamically based on actions or failures of other members of the collective. Note that the actual set of robots that can communicate directly at any time is a function both of this dimension and of the communication range and robot distribution in space.

By communication bandwidth: Communication may be inexpensive in terms of the robots' processing time, in that the robot has a special channel for communication, or it may be expensive in that the robot is prevented from doing other work while communicating. Sample points along this dimension include;

BAND-INF Communication is free. The communication bandwidth is sufficiently high that the communication cost and overhead can be ignored. This is a common assumption in theoretical computational models and can lead to robots that behave as if there was a central intelligence.

BAND-MOTION Communication costs of the same order of magnitude of the cost of moving the robot between locations. This can be thought of as being similar to the mechanism by which bees communicate by performing an intricate dance that is observed by other bees in the neighborhood. Systems such as the block moving algorithm of Brown and Jennings (1995) and some of the furniture moving approaches of Rus *et al.* (1995) use the task to signal communication. Although these may appear to be classified as BAND-ZERO (described below), they are more correctly classified as BAND-MOTION as the pushing action (motion) of one robot is communicated through the object being pushed to other robots in the collective.

BAND-LOW Very high cost. Communication costs much more than the cost of moving from one location to another. This suggests very independent robots.

BAND-ZERO No communication. Robots are unable to sense each other. As mentioned earlier, this is probably an impractical case if coordinated collective behaviour is desired.

Note that low bandwidth may be acceptable if the primary reason for using multiple robots is redundancy rather than efficiency.

Collective reconfigurability: The rate at which the collective can spatially re-organize itself; roughly equivalent to the rate at which members can move with respect to one another. For example, bees can presumably reconfigure their spatial layout with respect to one another very quickly while soldiers marching in lock-step or cars on a highway cannot. This dimension is closely related to the communication range of members of the collective. Changes in topology, however, will alter the nearest-neighbor relationships and thus are not equivalent to simple scaling of the communication range. In practice, there may be topological constraints to the allowed reconfigurations. For example, if the members of a robotic collective drive on roads, then only certain topological changes are allowed irrespective of member velocity. This can be seen in the work of Aguilar *et al.* (1995). Here global control of a collective operating within a roadway-like environment utilizes controllers at intersections to communicate with robots adjacent to and heading towards the intersection as well as other controllers. Another issue that determines reconfigurability is the possible presence of non-holonomic motion constraints on collective members: non-holonomic robots can reduce the rate of reconfiguration due to complex maneuvering that may be required, or they may render some configurations unattainable.

ARR-STATIC Static arrangement. The topology is fixed.

ARR-COMM Coordinated rearrangement. Rearrangement with members that communicate. In interesting example of this is the formation control work of Arkin *et al.* (1996) where the group of robots can sometimes change to a specified alternative topology.

ARR-DYN Dynamic arrangement. The relationship of members of the collective can change arbitrarily.

Static collective arrangement is likely to result in very fragile collectives.

The centralization/decentralization axis of Cao *et al.* (1995) includes aspects of the collective re-

Table 1. Summary of the taxonomic axes

Axis	Description
Collective Size	The number of autonomous agents in the collective.
Communication Range	The maximum distance between two elements of the collective such that communication is still possible.
Communication Topology	Of the robots within the communication range, those which can be communicated with.
Communication Bandwidth	How much information elements of the collective can transmit to each other.
Collective reconfigurability	The rate at which the organization of the collective can be modified.
Processing Ability	The computational model utilized by individual elements of the collective.
Collective Composition	Are the elements of the collective homogeneous or heterogeneous.

configurability and collective topology axes. Cao *et al.* distinguish between collectives in which there is a single controlling agent and those which do not, while the design space presented here places less emphasis on this particular dichotomy, in contrast with other options.

By processing ability of each collective unit: Each unit of the collective has a particular model of computation. It may be useful to model individual members of the collective with a computational model that is simpler, and therefore weaker, than that of a Turing Machine. For example, if individual members of the collective are modelled as finite state machines (Hopcroft and Ullman, 1979) (operating as a function of their sensors, the current communication input, and some finite number of internal states) then it will be possible to provide formal bounds on the execution of an individual member of the collective. It is interesting to note that, even if individual members of the collective have a particular limited computational model, the entire collective may have an overall computational ability that is considerably more powerful (see §3.1). Thus, there exists the attractive possibility of having collectives where the computational power of individual units is deliberately restricted, in order to allow formal reasoning about their behaviour for example, but where the collective as a whole exhibits very general computational abilities.

For simplicity, we deal only with the common simple sequential computational models. Note that this is a non-continuous dimension.

PROC-SUM Non-linear summation unit (Hertz *et al.*, 1991). This very simple unit is used in constructing a simulated neural network but may be too simple to be a realistic model for a single robot although it illustrates the near-extremum of this dimension.

PROC-FSA Finite state automaton. This is the computational model preferred by the subsumption architecture computational systems (Brooks, 1986). Finite state models are also used for many communication protocols to facilitate proofs of correctness (Tanenbaum, 1988). It should be noted that individual units may in fact be general-purpose processors, *programmed to behave as FSAs* in order to simplify reasoning concerning their behaviour.

PROC-PDA Push-down automaton.

PROC-TME Turing machine equivalent. The computational model assumed by most robotic systems.

By collective composition: Even an ensemble of robots that is homogeneous in terms of physical structure may be differentiated by programming or behaviour. Thus, heterogeneity can be subdivided into both a physical component and a purely procedural component implemented using physically homogeneous robots. Thus, a collective may be:

CMP-IDENT Identical. The collective is made up of units that are homogeneous in both form and function (hardware and software). Note that this does not preclude differentiation in the roles assumed by members of the group based on environmental or stochastic factors.

Table 2. Full taxonomic labelling of some sample collectives

Collective	Size	Comm. range	Comm. topology	Comm. bandwidth	Reconfigur- ability	unit processing	composition
combat aircraft	SIZE-LIM	COM-LONG	TOP-BROAD	BAND-INF	ARR-DYN	PROC-TME	CMP-HET
wolf pack	SIZE-LIM	COM-NEAR	TOP-BROAD	≈ BAND-MOTION	ARR-DYN	PROC-TME	≈ CMP-HOM
automobiles	SIZE-LIM	COM-NEAR	TOP-BROAD	≈ BAND-MOTION	ARR-DYN	PROC-TME	CMP-HET
bees	SIZE-INF	COM-NEAR	TOP-BROAD	≈ BAND-MOTION	ARR-DYN	PROC-TME	CMP-HET
Turing-machine example	SIZE-INF	COM-NEAR	TOP-ADD	BAND-INF	ARR-STATIC	PROC-FSA	CMP-HOM
Graph exploration example	varies	COM-NEAR	TOP-ADD	BAND-INF	ARR-STATIC	PROC-TME	CMP-HET
Metric exploration example	SIZE-LIM	COM-NEAR	TOP-ADD	BAND-INF	ARR-DYN	PROC-TME	CMP-HOM
Reconfiguration example	SIZE-LIM	COM-NEAR	TOP-BROAD	BAND-INF	ARR-DYN	PROC-TME	CMP-HOM
Positioning example	SIZE-INF	COM-NEAR	TOP-GRAPH	BAND-INF	ARR-COMM	PROC-TME	CMP-HOM
Herding/following example	SIZE-PAIR	COM-NEAR	TOP-BROAD	varies	ARR-STATIC	PROC-TME	CMP-HET
Aguilar (1995)	SIZE-LIM	COM-NEAR	TOP-GRAPH	BAND-INF	ARR-COMM	PROC-TME	CMP-HET
Anderson (1995)	SIZE-LIM	COM-NEAR	TOP-ADD	BAND-INF	ARR-STATIC	PROC-TME	CMP-HET
Arkin (1993)	SIZE-LIM	COM-NEAR	TOP-ADD	BAND-INF	ARR-STATIC	PROC-TME	CMP-HOM
Brown (1995)	SIZE-PAIR	COM-NEAR	TOP-BROAD	BAND-MOTION	ARR-STATIC	PROC-TME	CMP-HET
Causse (1995)	SIZE-LIM	COM-INF	TOP-TREE	BAND-INF	ARR-STATIC	PROC-TME	CMP-HET
Dickson (1995)	SIZE-LIM	COM-NEAR	TOP-BROAD	BAND-LOW	ARR-DYN	PROC-TME	CMP-HOM
Habib (1992)	SIZE-LIM	COM-INF	TOP-ADD	BAND-INF	ARR-DYN	PROC-TME	CMP-HOM
Hackwood (1992)	SIZE-LIM	COM-NEAR	TOP-GRAPH	BAND-INF	ARR-DYN	PROC-TME	CMP-HET
Kurabayashi (1995)	SIZE-LIM	COM-NEAR	TOP-ADD	BAND-INF	ARR-STATIC	PROC-TME	CMP-HOM
Kurazume (1995)	SIZE-LIM	COM-INF	TOP-ADD	BAND-INF	ARR-STATIC	PROC-TME	CMP-HOM
Marapane (1995)	SIZE-LIM	COM-NEAR	TOP-BROAD	BAND-MOTION	ARR-STATIC	PROC-TME	CMP-HET
Mataric (1992)	SIZE-LIM	varies	varies	varies	ARR-DYN	PROC-TME	CMP-HOM
Mataric (1995)	SIZE-PAIR	COM-NEAR	TOP-ADD	BAND-INF	ARR-STATIC	PROC-TME	CMP-HOM
Parker (1993)	SIZE-LIM	COM-NEAR	TOP-BROAD	BAND-MOTION	ARR-DYN	PROC-TME	CMP-HOM
Parker (1995)	varies	COM-NEAR	TOP-ADD	varies	ARR-STATIC	PROC-FSM	varies
Rao (1995)	SIZE-LIM	COM-NEAR	TOP-ADD	BAND-INF	ARR-STATIC	PROC-TME	varies
Rus (1995)	varies	varies	varies	varies	varies	PROC-TME	varies
Sandini (1993)	SIZE-LIM	COM-NEAR	TOP-BROAD	BAND-INF	ARR-DYN	PROC-TME	CMP-HOM
Sekiyama (1996)	SIZE-LIM	COM-INF	varies	BAND-INF	ARR-DYN	PROC-TME	CMP-HET
Ueyama (1992)	≈ SIZE-INF	COM-NEAR	TOP-TREE	BAND-INF	ARR-STATIC	PROC-TME	CMP-HOM
Yuta (1992)	SIZE-LIM	COM-INF	TOP-ADD	BAND-INF	ARR-DYN	PROC-TME	CMP-HOM

Note, further, that assigning unique labels to elements of the collective is consistent with this classification since it could be achieved procedurally, as it is in some computer networks.

CMP-HOM Homogeneous. The collective is made up of units all with essentially the same physical characteristics.

CMP-HET Heterogeneous. The collective is made of of units that are not physically uniform. In general, this also implies difference at the behavioral level.

This axis corresponds almost exactly to the differentiation axis of Cao *et al.* (1995).

The value of the taxonomy as a language of discourse concerning swarm robotics is twofold. First, it provides for the succinct description of systems and results in the literature. Second, it maps out the space of possible designs for a collective, giving the researcher guidance and perspective when engaged in any theoretical or practical work. To illustrate the descriptive power of the taxonomy, Table 2 provides the full taxonomic labelling of some sample collectives from the literature and nature as well as the labelling of the collectives presented in the following sections.

3. The power of robot collectives: case studies

Distributed computer processing has been extensively studied by theoretical computer scientists and mathematicians, as well as by computer designers. Many models of robot collectives map onto pre-existing computational or hardware models. An example of a related computational model is PRAM (parallel random access machines) (VanLeeuwen, 1990), which is highly developed, but has significant differences from robot collectives, because the latter involve mobile processors.

The following case studies illustrate that there is something to be gained by using a collective in place of a single robot. We show that sufficiently sophisticated collectives can be used to solve particular problems and relate these collectives to the taxonomy given above. We show that the performance of a collective can be provably better than that of a single robot for certain tasks such as exploration.

The first case study involves a proof that a collective of robots operated by finite automata is as powerful as a Turing machine, which is fundamentally more powerful than each individual member of the collective. The second case study demonstrates how a robot collective can make the exploration of a graph-like world possible without the use of a movable marker, and, with enough robots, also be more efficient than using a marker. The third case study shows how a collective of robots that can sense each other can make possible the self-localization of the robots in a landmark-poor environment, where individual robots alone cannot locate themselves. The fourth case study examines another aspect of robust positioning and mapping by a collective of robots that can sense each other, without reference to external landmarks. The last case study examines reconfigurable collectives, and in particular chains of robots.

3.1. *Turing Equivalence of a Collective of Finite Automata*

An unbounded number of robots $\{A_i\}$ (INF-GROUP) whose processing abilities can be modelled individually as finite automata (PROC-FSA) with the ability to communicate their state to their neighbours (COM-NEAR,TOP-ADD,BAND-INF) may simulate an arbitrary Turing Machine. This is notable, because this fact makes it possible in principle to construct a spatially distributed intelligence from a large collection of very simple devices. The individual automata may be mobile (moving according to their own current state, as assigned by the distributed computation), and thus able to accomplish some interesting actions in the world. It is not our purpose to explore applications of the simulation constructed for the proof of this result, because it is undoubtedly the case that more efficient use could be made of the collective members by tailoring their behaviour to the particular problem of interest instead of a Turing machine simulation.

For the purposes of the exposition, the automata and Turing machine are defined using the conventions and notation of Hopcroft (1979). Let M be an arbitrary Turing machine, given by

$$M = (Q_M, \Sigma_M, \Gamma_M, \delta_M, q_{0M}, B_M, F_M)$$

where the symbols in the tuple have the following meanings:

Q_M the finite set of states of the control
Γ_M the finite set of tape symbols
B_M the blank symbol,
initially marking unused tape locations
Σ_M the finite set of input tape symbols
$(\Sigma_M \subset \Gamma_M,\ B_M \notin \Sigma_M)$
δ_M the *next move* function
$\delta_M : Q_M \times \Gamma_M \longrightarrow Q_M \times \Gamma_M \times \{L, R\}$
q_{0M} the start state. $q_{0M} \in Q_M$.
F_M the set of final states. $F_M \subseteq Q_M$.

The function δ_M is the "program" of the Turing machine, giving its behaviour on each (state, input) pair of interest. Turing machines are interesting because they seem to capture formally the informal notion of computation.

We define an infinite set of communicating finite automata (aka the elements of the collective) $A_i,\ i = 0, 1, 2, ...$ as

$$A_i = (Q_A, \delta_A, q_{0A}, F_A, T_A, L_A, R_A)$$

where the symbols in the tuple have the following meanings:

Q_A the finite set of states of the control
δ_A the *next state* function
$\delta_A : Q_A \times Q_A \longrightarrow Q_A$
q_{0A} the start state. $q_{0A} \in Q_A$.
F_A the set of final states. $F_A \subseteq Q_A$.
T_A the *transmitting* states. $T_A \subseteq Q_A$.
L_A the *left-receiving* states. $L_A \subseteq Q_A$.
R_A the *right-receiving* states. $R_A \subseteq Q_A$.

The function δ_A captures the notion of communication. If A_i is in a transmitting state, then the value of δ_A depends only on the current state of A_i. If A_i is in a left-receiving state, then the value of δ_A depends on the pair consisting of the current state of A_i and the current state of A_{i-1} (such states are undefined for A_0). If A_i is in a right-receiving state, then the value of δ_A depends on the pair consisting of the current state of A_i and the current state of A_{i+1}.

To simulate an arbitrary Turing machine M, we set the components of the A_i as shown below.

$$\begin{aligned}
Q_A &= Q_M \times \Gamma_M \times C \\
q_{0A} &= (q_{0M}, B_M, \lambda) \; for \; i > 0 \\
q_{0A} &= (q_{0M}, B_M, \tau) \; for \; i = 0 \\
F_A &= \{(q, X, s) | q \in F_M, X \in \Gamma_M, \\
&\quad\; s \in C\} \\
T_A &= \{(q, X, \tau) | q \in Q_M, X \in \Gamma_M\} \\
L_A &= \{(q, X, \lambda) | q \in Q_M, X \in \Gamma_M\} \\
R_A &= \{(q, X, \rho) | q \in Q_M, X \in \Gamma_M\}
\end{aligned}$$

The set $C = \{\tau, \lambda, \rho\}$ labels the communications mode (transmit, left-receive, right-receive). The basic idea is that each automaton simulates both the finite control of M and one square of M's tape. At any time, the automaton corresponding to the current tape square is in transmit mode (one of the τ states). The automata to the left of the transmit automaton are in right-receive mode (one of the ρ states). The automata to the right of the transmit automaton are in left-receive mode (one of the λ states).

The transition executed by the transmit automaton does two things. First, it causes a state change to propagate outwards from the transmit automaton to the surrounding automata. Second, it causes one of the neighbours of the transmit automaton to become the new transmit automaton, simulating a move of the read-write head of the Turing machine.

For each transition $\delta_M(q, X) = (p, Y, L)$, corresponding to a left move of the Turing machine head, we define corresponding transitions for the A_i:

$$\begin{aligned}
\delta_A((q, X, \tau), (no\ rec.\ state)) &= (p, Y, \lambda) \\
\delta_A((r, Z, \rho), (q, X, \tau)) &= (p, Z, \tau) \\
\delta_A((r, T, \rho), (p, Z, \rho)) &= (p, T, \rho) \\
\delta_A((r, Z, \lambda), (q, X, \tau)) &= (p, Z, \lambda) \\
\delta_A((r, T, \lambda), (p, Z, \lambda)) &= (p, T, \lambda)
\end{aligned}$$

The first transition specifies that the current transmit automaton goes into left-receive mode, after executing the desired state change. The automaton state records both the Turing machine state and the symbol written by the Turing machine. The second transition specifies that the automaton immediately to the left of the transmit automaton also records the Turing machine state change in its state and becomes the new transmit automaton. The tape symbol recorded in its state remains unchanged. The remaining transitions specify that all other receive automata states that will arise should lead to a propagation of the Turing machine state information, while preserving the Turing machine tape content information.

For each transition $\delta_M(q, X) = (p, Y, R)$, corresponding to a right move of the Turing machine head, there are similarly corresponding transitions for the A_i:

$$\begin{aligned}
\delta_A((q, X, \tau), (no\ rec.\ state)) &= (p, Y, \rho) \\
\delta_A((r, Z, \lambda), (q, X, \tau)) &= (p, Z, \tau) \\
\delta_A((r, T, \lambda), (p, Z, \lambda)) &= (p, T, \lambda) \\
\delta_A((r, Z, \rho), (q, X, \tau)) &= (p, Z, \rho) \\
\delta_A((r, T, \rho), (p, Z, \rho)) &= (p, T, \rho)
\end{aligned}$$

There is a delay of one state-transition time at each automaton as the encoding of the Turing machine state propagates outward from the transmit automaton. Since this delay is the same for all operations, however, it has no effect on the outcome of any computation. Delay effects would become important if the states of the automata could be affected by the world they inhabit, which is inevitable for interesting systems.

The alternative of broadcast communications (TOP-BROAD) has also been considered. In order to adapt the above simulation to this case, replace left-receive and right-receive modes with a single receive mode. In order for control to pass from one transmit automaton to a new transmit automaton, each automaton must know its own index number, and the broadcast transmission must include the index number of the transmit automaton. This is a tradeoff which exchanges the uniformity of the individual automata in favour of a machine with a synchronous update of its components.

3.2. Exploring an unknown non-metric environment

In this section, we will consider how a group of robots can explore a graph-like (topological) environment more effectively than a single robot. In earlier work (Dudek *et al.*, 1991) it was demonstrated that a single robot lacking metric information is unable to explore a graph-like environment, but that if the robot is equipped with a marker that can be put down and picked up at will then the robot can do so. Algorithms were also developed for a single robot with a large number of markers. These results can be readily extended to robotic collectives of the class (TOP-ADD, BAND-INF, ARR-COM, PROC-TME, COMP-HOM) with very limited communication distances (COM-NEAR) by replacing the markers with individual members of a robotic collective.

Model the collective's environment as an embedding of an undirected graph G: $G = (V, E)$ with set of vertices V and set of edges E. The vertices are denoted by: $V = \{v_1, ..., v_N\}$ Here the world model is restricted to graphs G that contain no cycles of length ≤ 2, i.e. the graph contains no degenerate or redundant paths. This restriction prohibits the world from having multiple edges between two vertices or an edge incident twice at the same vertex.

The definition of an edge is extended slightly to allow for the explicit specification of the order of edges incident upon each vertex of the graph embedding. This ordering is obtained by enumerating the edges in a systematic (e.g. clockwise) manner from some standard starting direction. An edge $E_{i,j}$ incident upon v_i and v_j is assigned labels n and m, one for each of v_i and v_j respectively.

A member of the collective can move from one vertex to another by traversing an edge (a *move*), and it can sense the presence or absence of a particular unit from the collective at its current location. Sensing is limited to the current nodes. The robot collective is homogeneous (but we identify a specific controller robot which will be used to herd all of the other robots).

Assume that a member of the collective is at a single vertex, v_i, having entered the vertex through edge $E_{i,l}$. In a single move, it leaves vertex v_i for vertex v_j by traversing the edge $E_{i,j}$, which is r edges after $E_{i,l}$ according to the edge order at vertex v_i. This is given by the transition function: $\delta(v_i, E_{i,l}, r) = v_j$. Assume the following property about the transition function: **if** $\delta(v_i, E_{i,l}, r) = v_j$ **and** $\delta(v_j, E_{i,j}, s) = v_k$, **then** $\delta(v_j, E_{j,k}, -s) = v_i$. This implies that a sequence of moves is *invertible*, that is, a robot can retrace the steps it has taken[1].

A single move is thus specified by the order r of the edge along which the robot exits the current vertex, where r is defined with respect to the edge along which the robot entered this vertex. Note that in the special case of a planar embedding of a graph, enumeration of edges in a clockwise fashion satisfies the above assumption.

A member of the collective's perception is of two kinds, robot-related and edge-related perception.

Robot-robot-related Perception. Assume that a robot is at vertex v_i, having arrived via edge $E_{i,j}$. The robot-related perception is a K-tuple $B_s = (bs_1, bs_2, ..., bs_K)$, where bs_k has a value from the set $\{present, not\text{–}present\}$, according to whether robot k is present at vertex v_i.

Edge-related Perception. A robot can determine the relative positions of edges incident on the vertex v_i in some consistent manner, for example it can count off the doors in a clockwise ordering relative to the one it entered by. As a result, it can assign an integer label to each edge incident on v_i, representing the order of that edge with respect to the edge enumeration at v_i. The label 0 is assigned arbitrarily to the edge $E_{i,j}$, through which the robot entered vertex v_i. The ordering is local, because it depends on the edge $E_{i,j}$. Entering the same vertex from two different edges will lead to two local orderings, one of which is a permutation of the other. Note that if the graph is planar and a spatially consistent (e.g. clockwise) enumeration of edges is used, then two permutations will be simple circular translations of each other. But this will not hold in general, and in this paper we only assume that the edges can be ordered consistently.

The sensory information that the robot acquires while at vertex v_i is the pair consisting of the marker-related perception at that vertex and the order of edges incident on that vertex, with respect to the edge along which the robot entered the vertex. If the robot visits the same vertex twice, it must relate the two different local orderings produced and unify them into a single global ordering, for example by finding the label of the 0-th edge of the second ordering with respect to the first ordering. Determining when the same vertex has been visited twice and generating a global ordering for each vertex is part of the task of the exploration algorithm.

In prior work, we demonstrated that it is not possible for a single robot to explore and map an unknown environment with this sort of limited sensory information without the ability to somehow mark locations it visits (Dudek *et al.*, 1991). This is consistent with human intuition: fairy tales and mythology are full of stories of heroes who escaped from a maze by dropping markers or unwinding string as they went. The basic problem is that, when the explorer enters an environment, he cannot always determine if he has visited this location before. In (Dudek *et al.*, 1991) it was also demonstrated that as long as the explorer had a single unique marker which could be dropped and picked up at will it was possible for the explorer (or robot) to map fully his environment in $O(N^3)$ steps. The basic technique to use the invertibility of the path taken by the robot, and the fact that the marker is unique, to disambiguate locations which could be confused. The algorithm proceeds by incrementally expanding a map of the explored part of the world (the explored subgraph). Whenever a potentially new location is encountered, the marker is dropped at that location and all nodes in the already-explored subgraph are visited. If the marker is found, then this new location corresponds to an existing location. On the other hand, if it is not found, then this location really is new. In either case, additional information about the world has been obtained, which is added to the explored subgraph. In practice, there is usually more sensory information available than this problem formulation assumes and hence better performance can be achieved.

These results can be readily extended to show that a group of robots as specified above with respect to the taxonomy can perform polynomial time exploration without any static markers. Consider the case of a collective of two robots. One robot (the controller robot) can treat the other robot like the pebble, having the pebble robot move on the controller robot's command. Environmental exploration in this sensory deprived world cannot be solved by a single robot but it can be solved by a collective of size two or larger. A larger group of robots, however, can perform more sophisticated and efficient strategies analogous to those that are possible with a large number of markers (Dudek *et al.*, 1989). One approach entails having the robots expand the boundaries of the known subgraph in parallel, hence avoiding the time consuming component of the exploration strategy that involves checking for correspondences with previously visited locations. In the extreme case, where there is a very large number of robots (more robots than nodes in the graph – essentially a group of SIZE-INF) then it is possible to have a much simpler exploration strategy. Each robot simply moves until it finds an empty node and then stays there. This may be accomplished via (for example) a breadth-first search from the robots' start position, in which the first robot into each node is responsible for directing later arrivals out of the node by the least-recently-used exit. For each node in the graph, there will exist a robot that found the node in under N steps. Robots unable to find a free node may terminate their wandering whenever they encounter a robot that they have seen before. This will happen in at most N steps. As each node is now uniquely identified, it is straightforward for a robot to begin from the start node and visit all of the (now unique) nodes in the graph by traversing every edge twice ($O(N^2)$). If members of the collective have the ability to communicate farther than the current node then even more complex and efficient strategies are possible. If robots can communicate with *every* other robot (COM-INF), then no robot steps beyond the initial N are needed to map the graph, because it is possible to keep a globally accessible adjacency matrix that is updated with each step of every robot. Note, however, that robots that are forced to maintain a static arrangement (ARR-STATIC, in terms of either position or communications) will not be able to reduce their worst-case performance since complex evolutions in the shape of the set of the nodes to be explored can be constructed.

The number of robot moves used in exploring a graph with a small fixed number of robots (much smaller than N) has a bound of $O(N^3)$ (Dudek *et al.*, 1991). This results from the need to go back and actually visit all of the locations in the graph to solve the "have I been here before" problem. With a large collective (SIZE-INF) this worst-case bound will be reduced to $O(N^2)$ (and often lower in practice, for example $O(N)$ for a planar graph) since the unbounded size of the collective makes a costly re-examination of the known graph unnecessary.

3.3. Self-location and exploration in a metric environment

The previous section considered the task of exploring a graph-like environment with a robotic collective. Metric-based representations of space are common, and exploration and self-location in metric-based environments are common mobile robot tasks. For a collective to perform self-location and exploration in a metric environment two tasks emerge: The first is to estimate the pose and heading of members of the collective with respect to the underlying metric, and the second problem involves mapping unknown landmarks using measurements made by collective elements. In this framework it is possible to encounter situations, in which the self-location problem is not solvable by a single robot, but it is solvable by a robot collective (Milios *et al.*, 1995). Such a situation can arise in an environment with few landmarks and many occluding obstacles. In such an environment there may not exist a location from which a single robot sees the minimum number of landmarks required for self-location, whereas a robot collective, the members of which see less than the minimum number of landmarks, but also see each other and exchange sensory data, may have enough information to locate itself.

The metric self-location problem for a robot collective can be formulated as an optimization problem (Milios *et al.*, 1995). Assume that the collective $\{r_i\}$ consists of n distinct robots (SIZE-LIM), that have the capability of moving, turning, recognizing landmarks and other robots, and measuring bearing and elevation angles (or equivalently range). Each element of $\{r_i\}$ has access to a global metric map of the environment where the distinct known landmarks are accurately placed. Elements of the collective are similar (COMP-HOM), can communicate with each other (COM-NEAR, TOP-ADD, BAND-INF), have local processing capability (PROC-TME) and are free to move about with respect to each other (ARR-DYN) provided that the entire collective remains in communication contact.

The state vector $\mathbf{x}$ of the problem consists of the poses of all members of $\{r_i\}$. The pose of an element of the robot collective is a set of three variables, the x and y coordinates of the robot in a global coordinate system, and the heading of the robot, i.e. the angle from the x-axis to the forward direction of the robot, a vector in the x-y plane.

To define bearing and elevation angles, assume that each robot R in $\{r_i\}$ has a local coordinate system whose x and y axes are on a horizontal plane, and the z axis is vertical. A bearing angle from R to $L(x_L, y_L, h_L)$, where L is either a known landmark or another robot, is defined as the angle from the x-axis of the measuring robot to the projection of the ray RL onto the x-y plane. An elevation angle to L is defined as $\theta = \pi/2 - \theta_p$ where θ_p is the angle from the z-axis to the ray RL. If the height h_L of L is known, then the horizontal range from the robot to L (the length of the projection of the ray RL onto the x-y plane) is related to the elevation angle by $h_L = |RL| \tan(\theta)$.

The measurement vector $\mathbf{z}$ of the problem consists of the angle measurements (bearings and/or elevations) from all the robots to all other robots and to all landmarks, provided they are visible. $\mathbf{z}$ is a nonlinear function of the problem state $\mathbf{x}$, $\mathbf{z} = \mathbf{h}(\mathbf{x})$. The components of the measurement function $\mathbf{h}(.)$ are computed from $z = \theta - \tan^{-1}((y_L - y)/(x_L - x))$ when z is a bearing, and $z = \tan^{-1}(h_L/\sqrt{(x - x_L)^2 + (y - y_L)^2})$ when z is elevation. Here $(x, y, 0)$ is the location of the measuring robot and (x_L, y_L, h_L) is the location of the landmark or the other robot, all in global coordinates.

The size of the measurement vector $\mathbf{z}$ in general changes as the collective moves among opaque obstacles, and different landmarks (and other robots) become visible by each robot. Therefore the Jacobian matrix size and value also changes.

The problem of estimating the state $\mathbf{x}$ of the collective can be expressed as the minimization of the total squared difference $Q(\mathbf{x})$ between the predicted measurement vector $\mathbf{h}(\mathbf{x})$, and the actual measurement vector $\mathbf{z}$, where $Q(\mathbf{x}) = [\mathbf{z} - \mathbf{h}(\mathbf{x})]^T W [\mathbf{z} - \mathbf{h}(\mathbf{x})]$. The diagonal weight matrix W is based solely on the (measured) geometry, and its purpose it to weigh the contribution of each error according to how much we trust that particular landmark. In practice, distant landmarks are less reliable since the same angular error will correspond to a larger positional error for a landmark that is farther away, and weight more distant landmarks accordingly.

The objective function $Q(\mathbf{x})$ is a known nonlinear function of the state vector $\mathbf{x}$, and therefore this optimization problem can be solved by using a standard numerical method, for example the conjugate gradient method (Dennis and Schnabel, 1983; Fletcher, 1987). Minimizing $Q(\mathbf{x})$ results in an estimate of the position and orientation of each of the elements in the collective.

Now consider the problem of localizing unmapped (but distinctly recognizable) landmarks (this is the metric exploration problem). In this context, the collective moves about and senses recognizable landmarks that are not in its map. The collective senses at the same time a sufficient number of known (mapped) landmarks to be able to locate itself. The objective is then to be able to locate the unmapped landmarks on the map as precisely as possible from bearing and/or elevation measurements by individual members of the collective.

One possibility would be to include the positions of the unmapped landmarks in the state vector $\mathbf{x}$, and solve for the globally optimal $\mathbf{x}$. A longer state vector would lead to a higher-dimensionality search space and a longer measurement vector would lead to a larger Jacobian matrix. In this work, we have chosen to uncouple the two problems, that of multirobot self-location and of unmapped landmark localization. This leads to two decoupled optimization problems that can be solved in sequence. The solution thus obtained may be suboptimal according to the coupled optimization criterion stated previously, but it is definitely more practical, because the two optimization problems are much “smaller” than the coupled version, especially for robot collectives with a large number of robots. Another advantage of decoupling is that the landmark localization problem can be further decoupled by constructing a separate optimization problem for each unmapped landmark.

A single unknown landmark L_u at unknown location (x, y, s) may be observed from several robot poses resulting in bearings $(\phi_1, \phi_2, ..., \phi_M)$ and elevations $(\vartheta_1, \vartheta_2, ..., \vartheta_M)$. Because the bearing measurements depend only on the (x, y) of L_u and the elevation measurements depend only on the height s of L_u, we further decouple the problem into two separate problems, the first for computing (x, y) from the bearings, and the second for computing the height s from the elevations.

Estimation of (x, y) of L_u. A bearing ϕ_i is ideally equal to $h_i(x, y)$, where the function h_i depends on the pose of the sensing member of the collective. In practice the actual measurement of bearing is only approximately equal to the predicted measurement, and $\sum_i [\phi_i - h_i(x, y)]^2$ is minimized with respect to (x, y), where the index i ranges over all elements of the collective which sense this landmark.

Estimation of height s of L_u. The height of the landmark can be estimated using the elevation measurements ϑ_i, by minimizing $\sum_i [\vartheta_i - h_i(s)]^2$ with respect to s, where the index i ranges over all elements of the collective which sense this landmark. Here $h_i(s) = \tan^{-1}(s/\sqrt{(x_i - x)^2 + (y_i - y)^2})$, where (x, y) is the estimated position of the landmark, and (x_i, y_i) the estimated position of the ith element of the collective.

The following figures show the results of simulation experiments for self-location of a robot collective. In all experiments the robots traverse polygonal paths and they stop at the vertices of their path to measure bearings and/or elevations to landmarks. The angle sensing and motion errors were normally distributed with standard deviations approximated experimentally from a Nomadic Technologies mobile robot.

Figure 1 describes scenario 1, which includes two robots, continually traversing the paths shown. Robot 1 sees three fixed landmarks, and therefore can estimate its pose by measuring the bearings to these landmarks. Robot 2 sees only two landmarks, and therefore it cannot estimate its pose by bearing only, unless it includes the bearing measurement to Robot 1.

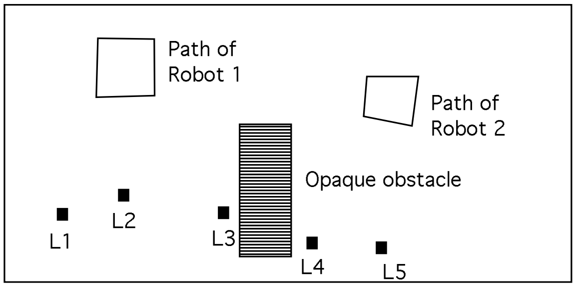


Fig. 1. The world of scenario 1, of size 6m x 3m. Five landmarks, L1..L5. Two robots traverse the paths shown and one opaque obstacle.

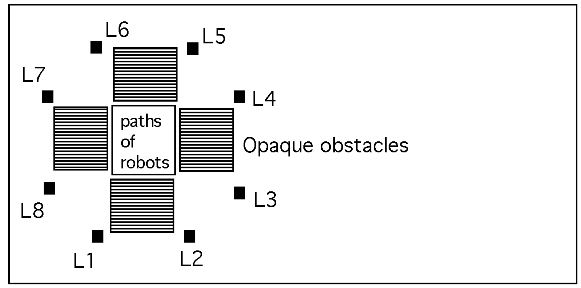


Fig. 2. The world of scenario 2, of size 6m x 3m. Eight landmarks, L1..L8. Four robots traverse the same path and follow each other at equal distances. Four opaque obstacles.

Figure 2 describes scenario 2, which includes 4 robots, continually traversing the same path and following each other at equal distances. None of these robots can estimate its pose by bearings to landmarks alone, since the robots only sense two landmarks each through the gaps between the obstacles. However, they can estimate their poses collectively, by measuring bearings towards each other, in addition to bearings towards landmarks.

The result of the experiments for scenario 1 is that without pose estimation or path correction, the maximum position and heading errors during 50 iterations of the paths were *0.5m* and *40 degrees* respectively, while with pose estimation and path correction using bearing measurements were *4cm* and *2.5 degrees*, and with pose estimation and path correction using both bearing and elevation measurements (thus implicitly estimating range) were *2.5cm* and *2.5 degrees* respectively. The result for scenario 2 is that without pose estimation or path correction the maximum position and heading errors during 50 iterations of the path were *1.4m* and *100 degrees*, whereas with pose estimation and path correction using bearing measurements the maximum errors were *2cm* and *2 degrees*. In similar simulation experiments for exploration (landmark localization), where a single robot observed both known and unknown landmarks, the maximum residual error of the position of the unknown landmarks was *8cm*, with an average residual error less than *4cm*.

3.4. Robust positioning and mapping

Although construction of a global metric representation may be useful for some applications of robotic collectives, it is also possible to use the collective itself to define a global representation of space. In (Dudek *et al.*, 1993b) we demonstrated how a collection of autonomous robots can define a mesh of local coordinate systems with respect to one another without reference to environmental positions (landmarks, markers, etc.). This simplifies tasks requiring robots to occupy or traverse a set of positions in the environment, such as mapping, conveyance and search. In this approach, sensing errors remain localized, and dead-reckoning plays no role. The approach involves a robot-based representation for the environment, in which metric information is used locally to determine the relative positions of neighbouring robots, but the global map is a graph, capturing the neighbour relations among the robots. We show that many tasks can be solved without reference to a global coordinate system, but that global metric maps may be constructed as desired, with small errors in the vicinity of any chosen position.

The problem is to use a collective of robots (CMP-IDENT, PROC-TME, ARR-COMM, BAND-INF, TOP-GRAPH, COM-NEAR, LIM-GROUP) each assigned a unique identification number, deployed on a finite 2-D surface to map the surface.

Let each robot define a unique local 2-D coordinate system in the space. The coordinate system used by robot r_i is a Cartesian system whose origin is at r_i and whose unit x-axis is the line segment joining robots r_i and $r_{j_{min}}$, where $r_{j_{min}}$ is the robot neighbouring r_i with minimum subscript value. The y-axis is chosen using the right-hand rule with the z-axis pointing up out of the plane.

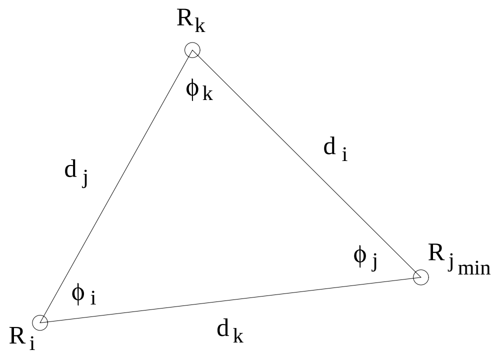


Fig. 3. Finding parameters in the local neighbourhood

Data is exchanged among robots in each local neighbourhood to determine the positions of the neighbours of each robot. A robot is considered a neighbour of another if there is reliable communication between the pair. A pair of robots may fail to be able to communicate if there is excessive distance between them or if there is an obstacle interposed between them (if communication requires a clear line-of-sight between them).

Locations of interest in space can thus be described relative to the positions of the neighbouring collective members. Positioning errors in each local neighbourhood are independent of those in other neighbourhoods, because no reference is made to a global coordinate frame. Instead, only topological (graph-like) relationships are used to relate one neighbourhood to another.

Definition of the representation The map representation kept at each r_i is illustrated in Figure 4. It consists of two graphs with labelled edges, $G_d(V, E_d)$ and $G_a(V, E_a)$. The two graphs are isomorphic but have different edge labels. Each edge $e_{ij} \in E_d$ is given the weight ∞ if robots i and j are not neighbours. Otherwise, e_{ij} is the distance between r_i and r_j, in r_i's coordinate system. Each edge $e_{ij} \in E_a$ is similarly labelled ∞ if r_i and r_j are not neighbours, but otherwise is labelled with the direction from r_i to r_j, measured counter-clockwise from the x-axis of r_i's coordinate system.

Construction of the representation Each robot r_i participates in the construction of the graph by first determining the set of robots with which it has reliable two-way communication. Then, information is exchanged with the other robots in this set to determine the labellings of edges e_{ij}, for all j, in each of E_d and E_a. Finally, this information is propagated over the entire set of robots, to be integrated into the complete edge sets. The details of the necessary communications protocols are complicated, but can be found in the existing literature, e.g. (Tanenbaum, 1988).

The details of the edge construction depend on what can be sensed by each robot. We consider two models, *neighbour distance only* and *neighbour azimuth (i.e. orientation) only*. The former corresponds to an ability to measure the signal propagation delay or phase shift with each robot in the neighbourhood. The latter corresponds to having line-of-sight communications, with direction-sensitive receivers. In both cases, our problem reduces to one of finding the unknown parameters in triangles such as the one shown in Figure 3.

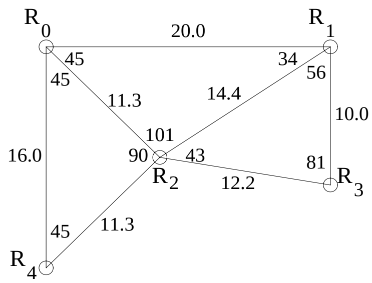
In the case in which only the distance to each neighbour is known, our unknown parameters are the angles ϕ_i, ϕ_j and ϕ_k, which may be found from the known parameters d_i, d_j and d_k.

In the case in which only the azimuth to each neighbour is known, our unknown parameters are the distances d_i, d_j and d_k. We may find only relative distances, since all similar triangles share the given values for ϕ_i, ϕ_j and ϕ_k. Setting $d_k = 1$ puts the distances in R_i-based coordinates.

Applications The main reason for using the above representation instead of a global, metric map, is that there is the potential for localization of features of the environment, without reference to fixed landmarks or dead-reckoning to learn or return to a position in the space. This is feasible because the location of a feature, like the positions of the robots, need only be determined relative to the closest robots in the space. All positioning may be done via triangulation with multiple robots.

The problems of mapping an environment with some sensors, conveying objects from place to place, and searching for a lost object, all require that at least one robot pass near each point in free space.

Figure 5 describes in high-level terms, a simple algorithm for doing this traversal. Once again, the many details of the communications necessary to coordinate the robots have been omitted. The algorithm makes greedy choices of strategy



Adjacency matrix for E_d

$$\begin{bmatrix} 0 & 1 & .57 & \text{inf} & .80 \\ 1 & 0 & .72 & .50 & \text{inf} \\ 1 & 1.27 & 0 & 1.08 & 1 \\ \text{inf} & 1 & 1.22 & 0 & \text{inf} \\ 1 & \text{inf} & .71 & \text{inf} & 0 \end{bmatrix}$$

Adjacency matrix for E_a

$$\begin{bmatrix} 0 & 0 & 315 & \text{inf} & 270 \\ 0 & 0 & 34 & 90 & \text{inf} \\ 0 & 259 & 0 & 216 & 90 \\ \text{inf} & 0 & 81 & 0 & \text{inf} \\ 0 & \text{inf} & 315 & \text{inf} & 0 \end{bmatrix}$$

Fig. 4. A sample configuration and its representation

1. Select a set of non-overlapping triangles among the robot triangles found.
2. While there remain triangles to traverse
   Do a serial traversal step as follows:
   a. while possible to do so:
      i. select an untraversed triangle with no corners selected to traverse other triangles and at least one corner free to move in this step
      ii. select a corner to traverse the interior of the triangle
      iii. disallow the remaining corners from moving in this step
   b. do the selected traversals in parallel.

Fig. 5. Traversal algorithm

at each step. As such, its performance is not optimal, but the algorithm serves as a demonstration of the principles underlying our map representation. The idea is to divide free space into non-overlapping triangles with a robot at each vertex. For each triangle, one of the robots is chosen to traverse the interior, while the other two robots maintain fixed positions. This allows the moving robot to determine its position by triangulation with the other two. Interesting locations within the triangle may be stored as a pair of angles with respect to the fixed robots. At all times, each triangle must have two fixed corners, in order that the third corner may resume its correct position following the traversal. This constrains the degree of parallelism in the traversals of different triangles.

Figure 6 illustrates the use of the algorithm on an example configuration of robots. The robots are labelled with integers, the triangles are labelled with letters. Triangles are shaded once they have been traversed. Five serial steps are required for the search of 22 triangles by 17 robots. Each step consists of the traversal, in parallel, of several triangles. The first four steps are shown in the figure. In each graph, circled nodes correspond to robots that are to move during the step; nodes enclosed in squares refer to robots that are constrained to be immobile during the step.

The network of triangles produced by the collective is a good representation of the free space in the environment, provided that the presence or absence of communication between robots is governed by the presence of absence of a free "line of sight" between the robots. Non-triangular faces in the mesh imply the existence of obstacles, while each edge in the mesh indicates an obstacle-free path segment.

The result of the above exploration process is a graph of the collective configuration labelled with local distance and/or azimuth measurements and including the relative locations of the objects of interest. All triangular faces of the graph are assumed to be part of free space (in Figure 6 there are no obstacles). As a result, the topology of free space is captured by the topology of the graph, in that non-triangular faces of the graph must contain obstacles. Features of interest in the environment are associated with the face to which they belong. Another robot collective, which returns to the same space equipped with this type of map, can use the map to find a feature of interest by expanding to occupy approximately the same positions in space as the original collective. Although the configuration cannot be duplicated accurately due to sensing errors, it will preserve the topology

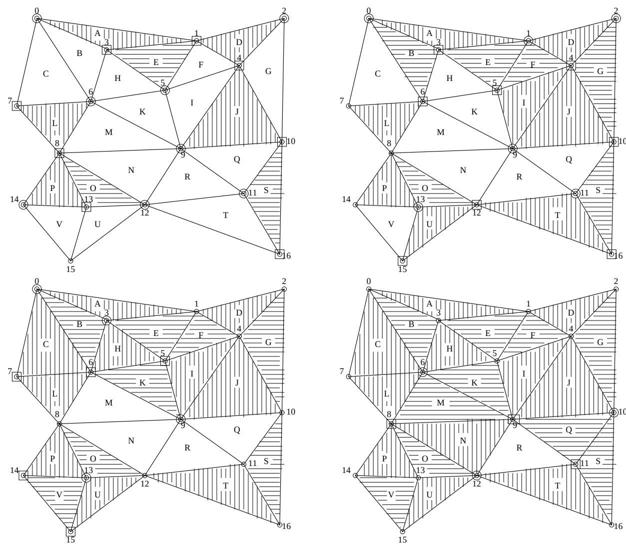


Fig. 6. Traversal example

The sequence A, B, C shows a lone randomly moving element of a collective approaching a chain. Arriving at the centre of the chain (A) it moves along the chain towards one of the ends (B) until it can join at the end becoming the new end of the chain (C).

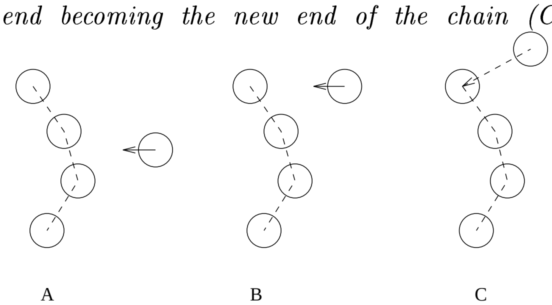


Fig. 7. A lone unit joins an existing chain

of free space, and the features of interest will not be too far from the face they are associated with. The objects of interest could now serve as landmarks to help reconstruct the original collective more accurately.

3.5. Building a reliable, reconfigurable communicating collective

Techniques such as the one presented above, for distributing sensing throughout the collective and then integrating the spatially disparate data into a global representation, require a collective connected by communication. The previous section assumed that this connectivity exists. Here we demonstrate a technique for maintaining this connectivity.

Assume that each element of $\{r_i\}$ can communicate with other nearby elements of the collective (COM-NEAR, TOP-BROAD, BAND-INF). As an example implementation, each robot could display, say by using a modulated beacon on the robot such as that proposed by Sandini *et al.* (1993) or Suzuki *et al.* (1995), information that nearby robots can sense. In addition assume that each robot is given an ID number which is unique (this is used only to break ties and its use is described later) but that otherwise the elements are identical (CMP-IDENT, PROC-TME, SIZE-LIM). Now suppose that these robots are dropped into an unknown environment and that their first goal is to form a communication network that will be used for later, more complex operations. The reconfigurable communication described here forms a connected chain of the elements of the collective. The chain may break, or reform, but the goal is to continually reform the available collective elements into a single connected component. This connected component adapts to the physical restrictions of the environment and limitations of

The sequence A, B, C, D, shows a complete chain (A) within which an element fails (B) breaking the chain into two sections. These two chains move independently (C) until they encounter each other again and rejoin (D).

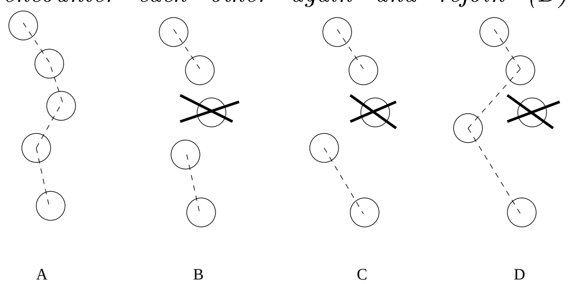


Fig. 8. Break in a chain

individual elements of the collective such as element failure or sensor limitations.

Each element of the collective is in one of two states. It is either in a chain or out of one. All robots initially start out of a chain. Robots broadcast this state information plus the ID number of the ends of the chain. (This information is continually passed up and down the chain.) Each robot's (r_i) behaviour is described as a function of its state:

- **Not in a chain:** Move randomly until r_i encounters another robot. Establish reliable communications with that robot. That robot is either in a chain or out of one. If it is out of one, the two robots form a chain. If that robot is in a chain, establish if it is an end unit or in the middle. If it is an end unit, attach r_i onto the chain and become the new end unit. If it is not an end unit move along the chain. Figure 7 illustrates a single unit attaching itself to a chain. Other strategies are possible, for example splicing the new robot into the chain. This may involve information exchange, if the individual elements of the collective have roles depending on their order along the chain.
- **In a chain:** If r_i is in the middle of the chain, pass messages up and down and broadcast the chain state. If communication fails between r_i and one of its neighbours then r_i becomes the new end unit of the chain and assert that information up or down the chain. If r_i is at the end of the chain, pass messages along the chain and broadcast the chain state. If r_i encounters the end of a different chain (or a single robot), join at your position (if it can join at more than one position it will choose one randomly). If the robot r_i joins, then r_i becomes an internal element of the chain and a new end unit will be chosen.

The unique robot identification number can be used to simplify synchronization between elements of the collective. Note that the resulting communication topology is independent of the underlying surface metric. The chain may not be physically mobile. It may simply be a communication mechanism. Mechanisms for two dimensional connectivity can be designed in a similar way (using triangular connectivity, for example).

Of course more complex operations are possible. It is possible to have robots in the chain move so as to have a regular spacing so that the chains cover more space. Also, long chains could attempt to straighten so that the chains form long lines rather than random walks.

It is interesting to note what happens in a chain when a node fails. Nodes above and below the failed node recognize that they are now the ends of two shorter strands, and begin to act accordingly. If they can communicate with each other then they can reform around their dead colleague (this is illustrated in Figure 8). If they cannot, two independent chains are formed which may or may not reconnect at some later time.

As a simple example application, suppose that a collective of robots is dropped on some planet by a landing vehicle (a lander). The lander can act as a fixed end of a robot chain. If it is not permitted to join more than one chain then it will always end up as a fixed head of a chain, and the chain of robots will grow away from the lander.

4. Some Practical Experiments

In the preceding sections we have proposed a design space or taxonomic space for multi-robot systems and considered some of theoretical implications and relationships between points in the space. In the following section, we will briefly examine the feasibility of the behaviours described and illustrate the instantiation of the taxonomy in practice. We will do this using a simple behaviour that can serve as a building block for more complex operations such as those described above.

Fig. 9. The two robots used in the experiments, Rosie (on the left) and Agamemnon (on the right).

Two RWI B-12 mobile robot bases, referred to as Agamemnon and Rosie, were used to implement a collective (SIZE-PAIR, COM-NEAR, TOP-BROAD, ARR-STATIC, PROC-TME, COMP-HET) with different communication mechanisms (BAND-INF, BAND-MOTION, BAND-ZERO) to solve the convoy task (Dudek *et al.*, 1995). There is considerable research interest in the task of having one autonomous vehicle follow another (c.f. Dickmanns and Zapp, 1987; Parker, 1994a). The task is usually implemented as only a single robot following some other autonomous agent. In particular, a variety of strategies are available for implementing this type of inter-robot collaboration. In previous approaches it has been assumed that the target to be followed does not actively aid in the processes but rather that the follower must attempt to track the leader as the leader undergoes possibly rapid random motion changes. By communicating the leader's intentions to the follower, simpler, more reliable convoy behaviours are possible. There are several natural design alternatives for this communication

Two-way communication The leader and the follower are in constant two-way communication (BAND-INF).

Explicit one-way communication The leader signals the follower(s) through some behaviour which can be sensed (BAND-MOTION).

Completely implicit behaviour The classic convoy model in which the leader ignores the follower(s) (BAND-MOTION).

Experiments have been conducted using each of these communication alternatives and are de-

Agamemnon's view of Rosie. This image shows Agamemnon's view after image processing. Horizontal lines have been coloured white, and small crosses have been placed in candidate horizontal stripes. Given the height of the top stripe, and the relative position of the third stripe, Agamemnon can compute Rosie's relative pose.

Fig. 10. Robot's sensor view

scribed in Dudek *et al.* (1995). Rather than describing each of the cases, we consider only the case of explicit one-way communication here. In the explicit one-way communication experiments Agamemnon (shown on the right in Figure 9), senses visually. The spiral pattern on Rosie's turret (see Figures 9 and 10), is designed to allow Agamemnon to compute easily the relative pose between Agamemnon and Rosie. Agamemnon can compute the distance between the two robots by locating the spiral pattern pattern in the image and computing the height of the lines in the pattern. Rosie's relative orientation is determined by the relative height of the 3rd stripe (counting from the top) in Rosie's pattern. Simple image processing is applied by Agamemnon to compute these values in his view of Rosie, as shown in Figure 10.

The first experiment involves herding: Rosie moves autonomously under external control, while Agamemnon moves to centre Rosie horizontally and to keep Rosie a particular distance away. Agamemnon implements this by defining an energy function that has a minimum when Rosie is centred and at an appropriate distance. This corresponds to the type of operation that a member of a robot collective would have to perform as part of a communication network described in section 3.5.

Given a collection of RWI's equipped with both the texture display as well as video sensors, a communication network can be established based on having robots maintain a preferred distance from other robots (sensed in the way described above), and then transmitting information based on rotational motion of each element of the collective.

In the herding experiments we found that Rosie could successfully herd Agamemnon around the lab, provided that Rosie made sufficiently small steps so that she did not move out of Agamemnon's field of view.

In a second experiment, Agamemnon followed Rosie while Rosie moved about her environment. This "following" or "convoy" task has been suggested for a number of different materials transport applications. The "following" task was implemented with two different levels of sophistication. In the first, the Herding algorithm was used. Rosie moved and Agamemnon followed, trying to maintain a preferred distance away from Rosie. This technique was successful, but as with the Herding experiment, small movements of Rosie were required in order to keep Rosie within Agamemnon's limited field of view.

In a second version of this experiment, Rosie telegraphed her movements to Agamemnon by rotating in the direction that Rosie was going to move. Agamemnon would then wait until after Rosie had moved and then move to where Rosie had been previously and then turn to have the same pose as Rosie. Using this technique, Rosie could move much farther and faster than using the herding technique above, because the field of view of Agamemnon's sensor was no longer a problem.

5. Conclusions

By providing a taxonomy for systems of multiple mobile robots, we have provided a common language for the description of seemingly disparate theoretical and practical results. The taxonomy serves the dual functions of allowing concise description of the key characteristics of different collectives, and describing the extent of the space of possible designs. As a result, we have been able to provide a succinct comparative survey of the current literature, and present our own work with reference to the taxonomy. Our results include constructive proofs of the following:

1. Collectives (SIZE-INF, COM-NEAR, TOP-ADD, BAND-INF, PROC-FSA) are equivalent to Turing machines.
2. Collectives (SIZE-INF, COM-NEAR, TOP-ADD, BAND-INF, PROC-TME, ARR-COM) can map a graph in $O(N^2)$ edge transitions per robot. If the collective is (COM-INF), then mapping can be done in $O(N)$ edge transitions per robot.
3. Collectives (SIZE-LIM, COM-NEAR, TOP-ADD, BAND-INF, PROC-TME, ARR-DYN) are able to perform mutual self-location in situations in which single robots cannot.

As well, we have presented practical algorithms for maintaining communications connectivity in the presence of possible failures, robust, accurate positioning based on the use of topologically-connected local frames of reference, and practical (BAND-MOTION/BAND-ZERO) convoying.

We are elaborating this framework to expand our understanding of the capabilities of various theoretical and practical collectives. Ongoing work relates to the formal inter-relationship between different collective classes.

Notes

1. To simplify the exposition, we also assume that there does not exist a $t \neq -s$ such that $\delta(v_j, E_{j,k}, t) = v_i$, although this could be readily accommodated.

References

Aguilar, L., Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1995). Ten autonomous mobile robots (and even more) in a route network like environment. In *Proc. IEEE/RSJ IROS*, volume 2, pages 260–267, Pittsburgh, PA.

Anderson, M. O., McKay, M. D., and Richardson, B. S. (1996). Multirobot automated indoor floor characterization team. In *Proc. IEEE R&A*, pages 1750–1753, Minneapolis, Minnesota.

Anderson, T. L. and Donath, M. (1991). Animal behavior as a paradigm for developing robot autonomy. In *Designing Autonomous Agents*, pages 145–168. MIT Press.

Arkin, R., Balch, T., and Nitz, E. (1993). Communication of behavioral state in multi-agent retrieval tasks. In *Proc. IEEE R&A*, volume 3, pages 588–594, Atlanta, GA.

Arkin, R. C. and Hobbs, J. D. (1992). Dimensions of communication and social organization in multi-agent robotics systems. In *Proc. SAB 92*.

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

Balch, T. and Arkin, R. C. (1995). Motor schema-based formation control for multiagent robot teams. In *1995 International Conference on Multiagent Systems*, pages 10–16, San Francisco, CA.

Beni, G. and Wang, J. (1989). Swarm intelligence in cellular robotic systems. In *Proc. NATO Advanced Workshop on Robotics and Biological Systems*, Il Ciocco, Tuscany, Italy.

Boas, P. (1989). Machine models and simulations. Technical Report CT-89-02, Institute for language, logic and information, University of Amsterdam.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE J. of Robotics and Automation*, 2:14–23.

Brooks, R. A. (1991). Intelligence without reason. Technical Report AI Memo No. 1293, MIT.

Brown, R. and Jennkings, J. (1995). A pusher/steerer model for strongly cooperative mobile robot manipulation. In *Proc. IEEE/RSJ IROS*, volume 3, pages 562–568, Pittsburgh, PA.

Cao, Y. U., Fukunaga, A. S., Kahng, A. B., and Meng, F. (1995). Cooperative mobile robotics: Antecedents and directions. In *Proc. IEEE/RSJ IROS*, volume 1, pages 226–234, Pittsburgh, PA.

Causse, O. and Pampagnin, L. H. (1995). Management of a multi-robot system in a public environment. In *Proc. IEEE/RSJ IROS*, volume 2, pages 246–252, Pittsburgh, PA.

Dennis, J. and Schnabel, R. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, Englewood Cliffs, NJ.

Dickmanns, E. and Zapp, A. (1987). Autonomous high speed road vehicle guidance by computer vision. In *Proc. 10th World Congree on Automatic Control*, Munich, Germany.

Dickson, W., Cannon, R., and Rock, S. (1996). Symbolic dynamic modelling and analysis of object/robot-team systems with experiments. In *Proc. IEEE R&A*, pages 1413–1420, Minneapolis, Minnesota.

Dudek, G., Jenkin, M., and a nd David Wilkes, E. M. (1989). Using multiple markers in graph exploration. In *Proc. Symposium on Advances in Intelligent Robotics Sys tems: Conference on Mobile Robotics*, Philadelphia, PA. International Society for Optical Engineering.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1991). Robotic exploration as graph construction. *IEEE Trans. on Robotics and Automation*, 7(6):859–864.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1993a). Organizational characteristics for multi-agent robotic systems. In *Proc. Vision Interface '93*, pages 91–96, North York, Canada.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1993b). Robust positioning with a multi-agent robotic system. In *Proc. IJCAI-93 Workshop on dynamically interacting robots*, pages 118–123.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1993c). A taxonomy for swarm robotics. In *Proc. IEEE/RSJ IROS*, pages 441–447, Yokohama, Japan.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1995). experiments in sensing and communication for robot convoy navigation. In *Proc. IEEE/RSJ IROS*, volume 2, pages 268–273, Pittsburgh, PA.

Fich, F. E., Radge, P. L., and Widgerson, A. (1988). Relations between concurrent-write models of parallel computation. *SIAM J. Comput.*, 17:606–627.

Fletcher, R. (1987). *Practical Methods of Optimization, 2nd Edition.* John Wiley, New York, NY.

Gage, D. W. (1992). Command control for many-robot systems. In *Proc. AUVS-92, Technical symposium of the association for unmanned vehicle systems*, pages 22–24, Huntsville, AL.

Habib, M. K., Asama, H., Ishida, Y., Matsumoto, A., and Endo, I. (1992). Simulation environment for an autonomous decentralized multi-agent robotic system. In *Proc. IEEE/RSJ IROS*, pages 1550–1557, Raleigh, NC.

Hackwood, S. and Beni, G. (1992). Self-organization of sensors for swarm intelligence. In *Proc. IEEE R&A*, pages 819–829.

Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the theory of neural computing.* Adison Wesley, Redwood City, CA.

Hopcroft, J. and Ullman, J. (1979). *Introduction to automata theory, languages and computation.* Addison Wesley, Reading, MA.

Kurabayashi, D., Ota, J., Arai, T., and Yoshida, E. (1996). Cooperative sweeping by multiple mobile robots. In *Proc. IEEE R&A*, pages 1744–1749, Minneapolis, Minnesota.

Kurazume, R., Hirose, S., Nagata, S., and Sashida, N. (1996). Study on cooperative positioning system. In *Proc. IEEE R&A*, pages 1421–1426, Minneapolis, Minnesota.

Langer, M., Dudek, G., and Zucker, S. W. (1995). Space occupancy using multiple shadowimages. In *Proc. IEEE/RSJ IROS*, pages 390–396, Pittsburgh, PA. IEEE Press.

Marapand, S. B., Trivedi, M., Lassiter, N., and Holder, M. (1996). Motion control of cooperative robotic teams through visual observation and fuzzy logic control. In *Proc. IEEE R&A*, pages 1738–1743, Minneapolis, Minnesota.

Mataric, M. (1992). Minimizing complexity in controlling a mobile robot population. In *Proc. IEEE R&A*, pages 830–835.

Mataric, M., Nilsson, M., and Simsarian, K. T. (1995). Cooperative multi-robot box-pushing. In *Proc. IEEE/RSJ IROS*, volume 3, pages 556–561, Pittsburgh, PA.

McFarland, D. (1989). *Problems of Animal Behavior.* Longman Scientific and Technical, John Wiley and Sons.

Milios, E., Wilkes, D., Jenkin, M., and Dudek, G. (1995). Multirobot landmark-based self-location and exploration. In *Proc. 3rd Int. Symposium on intelligent robotic systems'95*, Pisa, Italy.

Nitz, E. and Arkin, R. C. (1993). Communication of behavioral state in multi-agent retrieval tasks. In *Proc. IEEE R&A*, Atlanta, GA. IEEE Press.

Parker, L. (1993). Designing control laws for cooperative agent teams. In *Proc. IEEE R&A*, volume 3, pages 582–587, Atlanta, GA.

Parker, L. (1994a). ALLIANCE: An architecture for fault tolerant cooperative control of heterogeneous mobile robots. In *Proc. IEEE/RSJ IROS*, Munich, Germany.

Parker, L. (1994b). *Heterogeneous multi-robot cooperation.* PhD thesis, MIT EECS.

Parker, L. E. (1995). The effect of action recognition and robot awareness in cooperative robotic teams. In *Proc. IEEE/RSJ IROS*, volume 1, pages 212–219, Pittsburgh, PA.

Partridge, B. L. (1982). The structure and function of fish schools. *Sci. Am.*, pages 114–123.

Rao, N. S. V., Protopopescu, V., and Manickam, N. (1996). Cooperative terrain model acquisition by a team of two or three point-robots. In *Proc. IEEE R&A*, pages 1427–1433, Minneapolis, Minnesota.

Rus, D., Donald, B., and Jennings, J. (1995). Moving furniture with teams of autonomous robots. In *Proc. IEEE/RSJ IROS*, volume 1, pages 235–242, Pittsburgh, PA.

Sandini, G., Lucarini, G., and Varoli, M. (1993). Gradient driven self-organizing systems. In *Proc. IEEE/RSJ IROS*, pages 429–433, Yokohama, Japan.

Sekiyama, K. and Fukuda, T. (1996). Modeling and controlling of group behavior based on self-organizing principle. In *Proc. IEEE R&A*, pages 1407–1412, Minneapolis, Minnesota.

Suzuki, S., Asama, H., Ugeka, A., Kotosaka, S., Fujita, T., Matsumoto, A., Kaetsu, H., and Endo, I. (1995). An infra-red sensory system with local communication for cooperative multiple mobile robots. In *Proc. IEEE/RSJ IROS*, volume 1, pages 220–225, Pittsburgh, PA.

Tanenbaum, A. S. (1988). *Computer Networks, 2nd Ed.* Prentice-Hall, Englewood Cliffs, NJ.

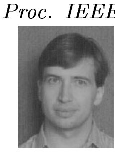
Tinbergen, N. (1951). *The study of instinct.* Oxford University Press.

Tinbergen, N. (1972). *The animal in its world; explorations of an ethologist, 1932-1972.* Harvard University Press.

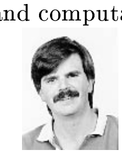
Ueyama, T., Fukuda, T., and Arai, F. (1992). Configuration of communication structure for distributed intelligent robot system. In *Proc. IEEE R&A*, pages 807–812.

van Leeuwen, J., editor (1990). *Handbook of Theoretical Computer Science: Volume A: Algorithms and Complexity.* Elsevier, Amsterdam.

Yuta, S. and Premvuti, S. (1992). Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. In *Proc. IEEE/RSJ IROS*, pages 1566–1574, Raleigh, NC.

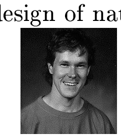


Gregory Dudek is an Assistant Professor with the School of Computer Science and a member of the Centre for Intelligent Machines (CIM). at McGill University. He obtained his PhD in computer science (computational vision) from the University of Toronto, his MSc in computer science (systems) at the University of Toronto and his BSc in computer science and physics at Queen's University. He has published research papers on subjects including visual object description and recognition, robotic navigation and map construction, distributed system design and biological perception. He research interests include perception for mobile robotics, navigation and position entimation, environment and shape modelling and computational vision.



Michael Jenkin is an Associate Professor with the Department of Computer Science, York University, Toronto. He received his Ph.D. in Computer Science from the University of Toronto in 1988 His current research interests include active stereo vision, mobile robotics, and immersive user interfaces. He is a member of the Institute for Space and Terrestrial Science Human Performance in Space Laboratory and the Institute for Robotics and Intelligent Systems.

Evangelos Milios received a diploma in Electrical Engineering from the National Technical University of Athens, Greece, and Master's and Ph.D. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, Massachusetts. While at M.I.T., he was a member of Digital Signal Processing group and he worked on acoustic signal interpretation problems at the M.I.T. Lincoln Laboratory. After spending 5 years doing research on shape analysis, sensor-based robotics, and signal processing in the Department of Computer Science, University of Toronto, he joined York University as an Associate Professor. His recent research focuses on the processing, interpretation and use of visual and range signals for landmark-based navigation and map construction in single- and multi-agent robotics. He is also interested in the application of computer vision and auditory scene analysis to the design of natural computer-human interfaces.



David Wilkes received B.Sc., M.Sc. and Ph.D. degrees in Computer Science from the University of Toronto, in 1983, 1985 and 1994 respectively, while involved in concurrent consulting and teaching activities. From 1991 through 1996 he worked on the ARK mobile robotics project, on the software architecture and navigation of a mobile robot in an uncustomized, dynamic industrial environment. He is currently working at Ontario Hydro Technologies in Toronto, Canada on problems in sensing, robotics and intelligent systems in the electrical power industry.