Stigmergic Navigation for Multi-Agent Teams in Complex Environments

Alfred Wurr and John Anderson

Autonomous Agents Laboratory
University of Manitoba
Winnipeg, Manitoba, Canada R3T2N2
awurr,andersj@cs.umanitoba.ca

Summary. Robotic agents in dynamic environments must sometimes navigate using only their local perceptions: for example, in strongly dynamic domains where paths are outdated quickly. Reactive navigation alone has limited power: domain features such as terrain undulation, geometrically complex barriers, and similar obstacles form local maxima and minima that can trap and hinder agents that use it exclusively. Moreover, agents navigating in a purely reactive fashion forget their past discoveries quickly. Preserving this knowledge usually requires that each agent construct a detailed world model as it explores or be forced to rediscover desired goals each time, and share elements of this knowledge explicitly in group situations. The cost of explicit communication can be substantial, however, making it desirable to avoid its use in many domains. In this chapter we present the design and implementation of cooperative methods for reactive navigation: allowing a team of agents to assist one another in their explorations through implicit (*stigmergic*) communication. These methods range from simple solutions for avoiding specific problems such as individual local maxima, to the construction of sophisticated branching trails. We evaluate these methods individually and in combination using behaviour-based robots in a complex, three-dimensional software environment.

1 Introduction

Mobile robots are being used with increasing frequency for a wide range of applications, from rescue and security to exploration of remote areas. Simulated robotic agents are also employed as characters in the dynamic and complex simulated worlds of computer games. Whether the domain in which a robot is embodied is physical or exists only in software, navigation is an essential component: robots must not only move, but explore their environment in a coherent way in order to achieve their goals [1]. The challenge of navigation increases in tandem with the topographical complexity of the environment. Navigating in a static open area with uninterrupted sight lines and few obstacles is relatively straightforward. Few physical domains are this simplistic, however: even a basic building interior is a complex environment containing many corridors, doorways, walls and obstacles. At its most basic, navigating within buildings involves finding and negotiating relatively narrow doorways and hallways and possibly locating and climbing stairs (or controlling an elevator). This is still making an assumption of regular floor surfaces. In outdoor environments, or in situations such as robotic rescue where an indoor environment is severely disturbed, domains become much more complex. Agents may be faced with terrain undulation, irregularly shaped obstacles, and winding maze-like layouts of debris that make finding and navigating to a goal problematic. Intelligent navigation in complex environments typically involves dual processes of localization – determining an agent's position in the environment, and path planning – constructing a collision-free path to an agent's goal assuming its current position is known [2]. Path planning requires either pre-existing knowledge about the environment, such as a map and the locations of objects within it [3] or the ability to acquire this information while exploring the environment. Accurate localization is essential for path planning to be successful, because if the agent is in error about its current location, the path that is calculated to get it to its destination will be likewise incorrect [4].

A variety of localization techniques exist, including methods that use odometry, global-positioning, and radio-sonar positioning [5]. In odometry-based localization, agents begin in a known starting location and continuously update their locations as they travel [5]. Global positioning, on the other hand, works by dividing the environment into a grid where locations can be identified by precise coordinates. Radio-sonar methods triangulate position by sending out sonar pings and calculating position based on differences in ping times, and an analogous approach can be used with more accurate laser equipment. Unfortunately, odometry-based navigation works only over short distances [1], because cumulative errors in odometry can quickly result in discrepancies between the agent's actual and perceived location [6]. The remaining methods can require preparation of the environment ahead of time in some approaches, but more importantly require more complex sensory equipment [5]. In addition to accurate localization, path planning algorithms require and operate on symbolic representations of the environment. These may be provided *a priori*, constructed at runtime based on an agent's perceptions, or both in combination. Storing, constructing and maintaining these maps can demand significant storage and computational resources [7]. This is one reason that for even very complex agents it is helpful to work on a simple map based on previous travels (e.g. a topological map) rather than a map showing every detail of the world around the robot [8]. To make matters worse, if the environment is dynamic and subject to extensive change, any dynamically constructed map may have only a limited viable lifetime. In such situations, path planning becomes limited in its utility, as the path planner is reduced to planning based only on the agent's immediate perceptions and/or a partial map that the agent builds as it navigates. Consequently, embodied agents must often navigate in environments that are partially (if not wholly) unknown for periods of time [9].

Re-planning based on changes in the environment is certainly possible. However, path planning is computationally expensive, and continual re-planning is not always feasible. Indeed, in a fast-changing environment, even creating a single full path that is still useful upon completion may not be possible.

These drawbacks are even more severe when agents are planning and acting as a team, because of the greater complexity of group plans and uncertainty about the possible actions of teammates [10]. Agents must be prepared to adjust to changes in the environment caused by others, as opposed to simply the spontaneous changes offered by the world itself. For example, an agent's intended path may be abruptly cut off by objects placed by others or by the bodies of other agents themselves.

All of these factors contribute to making systems that rely exclusively on path-planning impractical in many real world scenarios [11]. Even when not relied upon exclusively, the difficulties described here still hinder the system merely because path planning is present. While it brings obvious benefits where it works well, it raises the issue of balancing cost and benefit: how much can be achieved without path planning? This is one facet of the overall issue of parsimony, a very important one in robotics. The more that sophisticated hardware is relied upon, the greater the likelihood that any unreliability in that hardware can cripple the system, irrespective of the benefit it brings. For example, a system relying on GPS for localization can currently be accurate enough for many outdoor environments, but is useless in situations where interference makes GPS contact unreliable. More sophisticated sensors also make robots significantly more expensive, which is an important factor in domains where equipment may be destroyed or damaged. This also becomes an issue when attempting to work with a team of robots, in that every component that makes an individual more expensive decreases the size of an affordable team. At some point, the benefit of additional team members may outweigh the benefit of increased sensors among all individuals. Adding more sophisticated hardware also means adding more control software and adding additional levels of complexity to the entire system along with the potential for unexpected interaction between components [12, 13]. Finally, being parsimonious in terms of specialized hardware also leads to a focus on more intelligent software, which is important from the perspective of advancing artificial intelligence [13]. A desire to avoid the disadvantages of an over-reliance on path planning, combined with a desire to achieve as much as possible with very parsimonious agents, has led us and others to explore the use of reactive navigation techniques in unknown domains [9, 14]. Instead of prescriptively navigating to a desired location via path planning, reactive navigation emerges from an agent's responses to its immediate local perceptions [10]. This allows agents to react quickly to changing circumstances, does not require a map of the environment, and since agents maintain at most a limited model of the world, localization is often not as crucial [15]. Reactive navigation is also referred to as *local* navigation, since the agent continually decides on the best direction in which to move based mainly on its local perceptions [9]. Reactive control mechanisms, such as those based on potential fields [16] are highly vulnerable to local minima. Local minima situations occur when all local perceptions lead an agent to a location that is not its goal; for example, when an agent can see its goal but for reasons of geographic topology must move *away* from the goal to ultimately reach it [17]. Typically, a certain amount of noise or randomness is added to the agent's chosen movement vector (desired velocity and orientation of travel) in order to partially deal with the problem by allowing an agent to move away from a local minimum and hopefully find new perceptions that lead it to more promising places [18, 12, 17]. Agents can similarly become trapped in local maxima where they lack the stimuli needed to choose a movement vector. For example, box-canyons (Figure 1) are a common local maxima problem, where agents are unable to see a goal of any type [9] and are reduced to wandering randomly before stumbling upon an exit [7]. The constricted zone between the two open areas is also an example of a problematic bottleneck, an area that is difficult to navigate due to the restricted space, especially when it is an area through which many agents must pass.

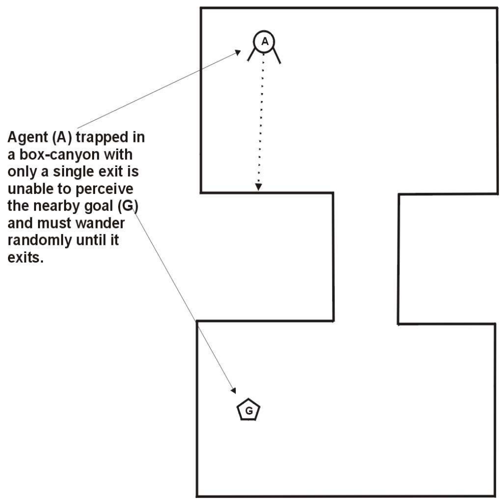

A
Agent (A) trapped in a box-canyon with only a single exit is unable to perceive the nearby goal (G) and must wander randomly until it exits.
G

Fig. 1. A box-canyon situation

The lack of goal-directed ability makes purely reactive navigation unsystematic, and agents (as individuals and as a team) do not benefit from the knowledge they have gained by exploring an environment (unless they are constructing a detailed world model as they explore) [19]. Thus, if an agent trapped in a box-canyon finds its way out and gets trapped in the same box-canyon again, it must repeat its search for an exit. In addition to bottlenecks and local minima and maxima, agents are also subject to cyclic behaviour, where they oscillate between multiple stimuli and never reach their goals (or at best waste time) [20, 21]. An ideal agent navigation system is one that has the responsiveness of reactive navigation and the proactivity of path planning methods. One partial solution to this problem is to allow agents to maintain state to remember experiences and places they have been [7]. However, this is contrary to a fundamental design goal of the purely reactive approach in that such agents are intended to maintain little or no state information [22]. Even though behaviour-based approaches relax this restriction to varying extents, it is still generally desirable to minimize state information in these agents for reasons of parsimony. As a compromise, both reactive and deliberative methods can be employed, resulting in a hybrid agent where local navigation is typically handled by a reactive subsystem while path planning methods are employed in a deliberative controller (e.g. [21, 8]). In systems that employ path planning, reactive navigation allows robots to respond to the unexpected in real time and handle aspects of the environment that have changed or were not known ahead of time [7]. Consequently, improving local navigation is desirable, whether it is used exclusively or as part of the reactive subsystem of a hybrid agent. Moreover, if reactive navigation can be made more effective, it can be employed more widely without resorting to more expensive path planning techniques. There are numerous potential improvements one can make to reactive navigation, including predicting what may occur in the environment and taking advantage of domain-specific phenomena. We are interested in ultimately supporting large teams of parsimonious agents, and so our efforts have been focused on improving reactive navigation by taking advantage of the experience of teammates in the environment. As agents explore the environment, they encounter a wealth of information that can be shared with others. With this comes a variety of decisions that must be made in any agent design: what mechanisms (both physical and linguistic) are to be used to communicate this information, what to communicate, when such information is to be shared, and with whom, for example. The principle of parsimony applies here as well: while one can develop agents that employ very elaborate communication schemes, these may not have a significant performance advantage over simpler agents once costs and benefits are weighed. Accordingly, we attempt to deal with the problems encountered in reactive navigation without the use of any explicit communication. Instead, agents assist one another in navigation by identifying desirable locations and sharing this knowledge with each other by modifying the environment. This form of implicit communication is properly referred to as *stigmergy*, a term used in biology to describe the influence that previous changes to the environment can have on an individual's current actions [23]. Naturally-occurring stigmergy is most commonly associated with the behaviours of social insects [24].

The remainder of this chapter overviews previous literature on reactive navigation and stigmergy, presents a number of increasingly sophisticated techniques for using stigmergy to assist in reactive navigation, and evaluates these in a complex three-dimensional software domain that embodies all of the classic problems associated with reactive navigation.

2 Related Literature

To date, most work with stigmergy has dealt with its effects on simple homogeneous robots collectively performing foraging or sorting tasks (e.g. [5, 23, 12, 25]). These differ from goal-directed navigation in that foraging has very general goals (e.g. find any food, which is satisfied simply by obtaining any instance) as opposed to the more specific nature of constructing a path to a single goal. Tasks such as foraging and sorting are representative of some robotic applications, and are also closely related to the behaviours of the natural creatures (e.g. ants and termites) that inspire the model. Social insects such as ants are homogeneous, and redundant in the performance of group tasks. This redundancy ensures that group goals are still met, even when an individual performing a task fails to complete it, because the sub-tasks necessary to complete a larger task do not have to be completed by a single individual. The presence of environmental cues alone eventually ensures that a complete sequence of actions is executed, even if the steps in a sequence are performed by different individuals [23].

The potential benefits of modifying the environment to affect the actions of others can also be realized in more complex domains. Werger and Mataric [5, 26] use a form of stigmergy in which robot bodies are substituted for chemical pheromone. In their work, a team of robots searches an area without global positioning and only limited sensors by forming robot chains. The primary drawback with these robot chains is that close contact between robot bodies limits parallel action. Most of the team members are prevented from actually doing useful work by the requirement that they keep their position in the chain. In addition, the area of exploration is bounded by the maximum length of the robots' bodies laid out in a line. Balch and Arkin [7] have explored using limited local spatial memory to deal with navigating box-canyons in reactive schema-based robotic agents. Agents remember and avoid visited locations by recording this information internally in a 2-dimensional integer array. Although this improved navigation, they found that the robot could become surrounded by visited locations and find no direction in which to move. Also, since the stigmergy depends on an agent maintaining a rudimentary world model, this model occasionally became inconsistent with the world, causing a divergence between the agent's perceived and actual location.

Vaughan et al. [27] implemented a team of robots capable of locating and recovering a supply of resources within a complex and initially unknown environment. Their robots generate and share waypoint coordinates via radio communication, which they maintain as an internal list of *crumbs* and *places*. These lists form trails that the agents then follow. Like Balch and Arkin's work, stigmergic markers are used here as part of an overall internal world model rather than a set of perceptible markers in the environment. Simulating stigmergy within agents' world models as opposed to marking the physical world avoids the problem of marker clutter, which would be a significant issue here because crumbs are dropped by all agents at regular intervals. However, because agents must communicate their trails explicitly, this compromise means that all the disadvantages of explicit communication remain. The method depends on agents being able to localize themselves accurately, and was found to fail after the cumulative error between the robots' coordinate records and the real world became too great for crumb trails to be shared effectively.

Parunak, Sauter, et al., [25, 28] employ ant-inspired stigmergy in a software environment that uses synthetic pheromone to coordinate unmanned aircraft employing potential field-based navigation. Their method uses a distributed network of *place* agents that are used to record and control pheromone aggregation, propagation and evaporation. Thus while the pheromone is virtual, the trails are physical in that they are maintained by embodied computational entities. These trails allow *ghost* agents to move across place agents by following the stigmergic trail. This approach is interesting but requires extensive infrastructure: place agents must first be evenly distributed throughout a predetermined region, be able to communicate reliably with each other via a wireless network, and also have appropriate sensory apparatus to detect hostile targets that are of interest to the ghost agents that traverse the network. The question of maintaining this extensive infrastructure under the conditions a combat environment presents has not yet been adequately addressed.

Kube and Bonabeau [29] examine ant-inspired coordination in a transportation task using a team of robots with decentralized control. In their work a robot team cooperatively performs a coordinated movement task (pushing a box that is too heavy for one agent toward a goal marked by a spotlight) without explicit communication, using locally sensed information only [29, 30]. When robots involved in the task detect a lack of progress (i.e. the box is not moving), they re-orient their position until box motion is once again observed. Thus, if robots are pushing from opposite ends and hinder one another's efforts, they gradually realign themselves. Directed box-pushing to a position specified by a spotlight is also achieved without explicit communication. This is accomplished by designing robots to push the box only when they are in contact with it and when the goal spotlight is not detectable. This has the effect of aligning the robots on the side of the box furthest from the goal, causing them to push it in the desired direction. Though this work focuses on stagnation recovery behaviours for a team of reactive robots, it is relevant to work in stigmergy in that the robots sense progress (or lack thereof, caused by the incompatible movements of other robots) in the environment and adapt accordingly through performance of the task itself. This emphasizes the feasibility and validity of using stigmergy to improve agent performance as individuals and a team.

Rumeliotis et al. [31] suggest a somewhat different ant-based approach to navigation using landmarks rather than pheromone for navigation. They note that desert ants navigate using a combination of visual landmarks and path integration. Path integration is the process of keeping an accurate idea of the direction of one's starting point relative to one's current position by updating this global vector as one travels with angles steered and distances covered [31]. Even for humans, path integration in unfamiliar or confusing areas can be quite difficult and easily subject to error. Consequently, desert ants learn and integrate a series of local vectors between a number of landmarks to navigate [31], rather than relying exclusively on path integration. This allows them to follow meandering and complex paths to their nest or outward from it by breaking the journey into discrete vectors between recognized landmarks. Though the ants are not explicitly creating the trail, they are actively interpreting their environment to their advantage.

Howard et al. [32] use stigmergy in a similar manner. In their work, the bodies of a team of mobile robots are used as landmarks to allow robots to localize themselves based on the relative range, bearing and orientation of other agents in a dynamic and/or hostile climate. While this shows the potential for using stigmergy by actively creating landmarks when none exist, the problem with this approach is the use of the robots themselves. A robot serving as a landmark is not likely doing useful work in such an environment.

3 Stigmergic Navigation Techniques

In contrast to these works, we employ a number of different stigmergic techniques that involve the deployment and exploitation of physical markers (as opposed to markers used only internally and communicated explicitly) in order to assist in team navigation. We refer to the process of using these marking techniques collectively to make more purposeful reactive navigation decisions in an unknown environment as *stigmergic navigation.* These techniques are described and demonstrated here in the context of reactive agents navigating in an unknown three-dimensional indoor environment. The particular agent architecture was not a strong consideration, but we chose a schema-based approach [33], one of a range of behaviour-based approaches where overall intelligent behaviour results from the complex interaction between behaviours and the world. While these techniques should be equally applicable to other types of agents, such as hybrid agents, we chose this mechanism in order to employ a very simple set of agents, so that the value of the techniques themselves could be considered separately from the sophistication of the agents used.

For stigmergy to be possible, agents need to be capable of marking their environment in a manner recognizable to others. A convenient mechanism for accomplishing this, and the one used here, is to simply place perceivable markers (i.e. physical objects) on the ground at particular locations. Our techniques make use of two types of stigmergic markers: *homogeneous* and *heterogeneous* markers. In conjunction with the agent's behaviours, each of these can be designed to permanently or temporarily attract, repel, or induce other specific behaviour in an agent.

Homogeneous markers are those where there is no distinction made between individual markers: the presence and location of a marker alone is significant. While they have numerous applications, their main limitation is that agents have no ready mechanism for mediating between markers when more than one is visible, other than aspects of the environment such as marker distance that may have no meaning to the marker's context. Heterogeneous markers, in addition to imparting information by their location, encode a *value* (perceptible by an observing agent, and possibly unique) that can serve to identify a group of markers or uniquely identify a marker, allowing markers to be grouped to serve unique purposes. In addition to encoding such labels, the value associated with heterogeneous markers can also be used as an attractive value, allowing agents to differentiate markers to follow. While it is possible in theory to use homogeneous markers in a heterogeneous fashion by defining a sufficient number of different marker types (for example, rather than encoding a value in a painted marker, different colours of paint could be employed), this may prove difficult in practice due to the number of marker types that may be necessary (e.g. the limitation on the perceivable differences in shades of paint).

Stigmergic navigation involves selecting appropriate marker types, conditions for dropping markers, and conditions for following markers under particular situations. The following subsections describe how combinations of these elements can be used to solve problems traditionally associated with reactive navigation as outlined in Section 1. These range from very simple actions that require little coordination, to more complex activities that involve a team of agents cooperatively and dynamically constructing marker trails leading to discovered goal locations. As the simpler approaches can be used to provide additional support for more complex activities, we begin with these. Further details on all techniques may be found in [14].

3.1 Marking Bottlenecks

Bottlenecks are extremely common in indoor environments. For example, any path from a given room to another must pass through some number of doorways and/or hallways, each of which forms a zone (small in the case of a doorway, but likely of much greater extent in hallways) within which agent navigation is markedly more difficult because of a lack of movement space. Moreover, each of these zones is small from the perspective of perception, and therefore difficult for a reactive agent to be drawn into. Such bottlenecks are not only constrictive, but important to the agent's successful performance. Within the open area of an empty room, for example, any one coordinate position in the enclosure is no more critical than another in terms of choosing a path to get to another room (while some may be considered better in terms of being more direct, all will be adequate). The exception to this is the position corresponding to the doorway itself - the bottleneck which must ultimately be part of the path. One of the simplest uses of stigmergic markers is the placement of heterogeneous markers in such constricted areas, in order that agents can supply each other with an attractive stimulus in the absence of pre-existing naturally-occurring stimuli. Traversing such an area is an interesting problem in that in many cases attracting the agent is not enough. In the case of a doorway, perception of a new area is immediate once one is present in the constricted area. However, in a hallway such as that forming the box-canyon shown in Figure 1, once an agent is drawn into the constricted area, measures must also be in place to ensure that the agent continues through it, rather than moving back to the area that it was exploring previously.

The technique we employ requires the use of heterogeneous markers with unique identifiers, as well as designing agents to be able to perceive markers and record recently visited markers in a short-term memory. To achieve a balance between exploring an area and moving to a new area, an agent is only attracted to markers when it has not followed a marker trail for a given period of time. In this state, the agent is considered to have *marker-affinity* and will move toward any marker of this type that it sees. As soon as the agent is successful in reaching a marker, it enters into a *marker-following* state. In this state the agent continues to be attracted to bottleneck markers that it has not visited earlier (as long as it detects that it is in a constricted area such as a doorway or hallway), but is not attracted to those markers it has already visited.

By recording and ignoring markers that have been visited while in marker following mode, the agent is drawn along by the trail of unvisited markers until it enters another open area. As soon as the agent detects that it is no longer in a narrow space (doorway or hallway) it transitions to a *marker-neutral* state during which it is no longer attracted to bottleneck markers. Each computational cycle, any markers that have not been visited within a set period of time are purged from the agent's list of visited markers. This is necessary to allow the agent to potentially traverse a previously travelled marker trail at some future time. This purging also serves to keep the agent's memory requirements very minimal. The marker-neutral state persists for a predetermined period of time, before the agent once again becomes marker-affinitive.

These bottleneck markers act to draw agents through tight spaces that are difficult for agents using local navigation to find and move through. If the room has only one entrance/exit, an agent can escape it more quickly and continue its explorations via the trail it followed to get in. Conversely, if the room has several unexplored portals the agent still has the opportunity to discover these alternate exits during the period in which it is not attracted to markers. While in this case the agents are told the perceptual features that distinguish a bottleneck, in future we intend to employ an information-theoretic measure of entropy as a criteria for dropping these markers, in order that the measurable need for a marker can be directly reflected in the likelihood of an agent placing one.

3.2 Marking Local Maxima

In addition to using markers to draw agents through constricted spaces in the environment, agents can be assisted in negotiating local maxima through stigmergic means. Our philosophy here is that agents will be attracted to local maxima already through the basic principles of reactive navigation, and thus it is worthwhile to move the agent through the environment faster by having these be more attractive. However, once an agent is drawn to a local maximum, it must be moved along through the environment in order that it does not simply stay there. Having an agent be repulsed by a local maximum would allow it to move away from the local maximum, where it would be likely to find some exit or perceive another local maximum marker (or some other stigmergic marker if multiple techniques are being combined) that would allow it to be coherently moved through the environment. We achieve this by using homogeneous markers that are initially attractive to an agent, but cause agents to be repulsed once the agent reaches a specific proximity. This avoidance state persists for a predefined period of time, allowing an agent time to exit the local maximum situation.

An agent marks a local maximum when no goals are visible from the agent's current position (the essence of a local maximum), no local maxima markers are perceivable within a given range (to minimize the number of markers), and no walls or other obstructions are nearby. The last constraint is intended to ensure that these markers are placed in the open, so that they force agents to the edges of local maxima areas when they are repelled, where the agents are more likely to happen upon an exit. This also has the effect of drawing the agent further into the open when they are attracted to the markers, where the agent is likely able to see a larger portion of its environment.

Unlike the previous marker type, in most domains these markers must vanish (or the ability to perceive them be altered) after some period of time. This is because unlike hallways, local maxima are likely due to the interaction of the environment and the agents' current goals, and must change when these goals do.

This method of marking can be likened to the use of spatial grid maps by Balch and Arkin [12]. Unlike their method, however, our agents are storing information about visited locations in the environment, rather than in an internal data structure. In addition, this information is less fine-grained, making it less verbose, and can be generated and used by multiple agents simultaneously.

3.3 Marking Local Minima

Another application of stigmergic markers is to allow agents to deal with local minima. These can be especially difficult in environments where three dimensions must be considered. For example, low lying obstacles such as a stair railing can allow an agent to perceive its goal while physically preventing the agent from moving directly toward it (as opposed to many two dimensional environments, where any obstacle is assumed to completely occlude objects beyond it).

In order to deal with this particular situation, we introduce another homogeneous marker type - *local minima* markers. Rather than attracting or repelling agents, these markers are unique in that they prescribe an *action to undertake* in response to a specific situation at a particular place in the environment.

In the indoor environment we use for our implementation, for example, these markers can be used to compel agents to jump when sufficiently close to them, allowing an agent to leap over a low lying obstacle. In order for this prescribed action to be a useful one, agents only drop these local minima markers when they perceive an attractor and a low-lying barrier in their path to it and when they are sufficiently close to the barrier itself. Unlike other markers, these are domain dependent by default, a natural consequence of encoding a particular action to take.

In our implementation, these markers are designed to disappear after an interval in the same fashion as local maxima markers. We deemed this necessary in that a local minimum in our implementation arises not solely from the environment, but from its interaction with the changing goals of agents. In domains where this is not the case, local minimum markers can be permanent.

3.4 Stigmergic Trail-Making

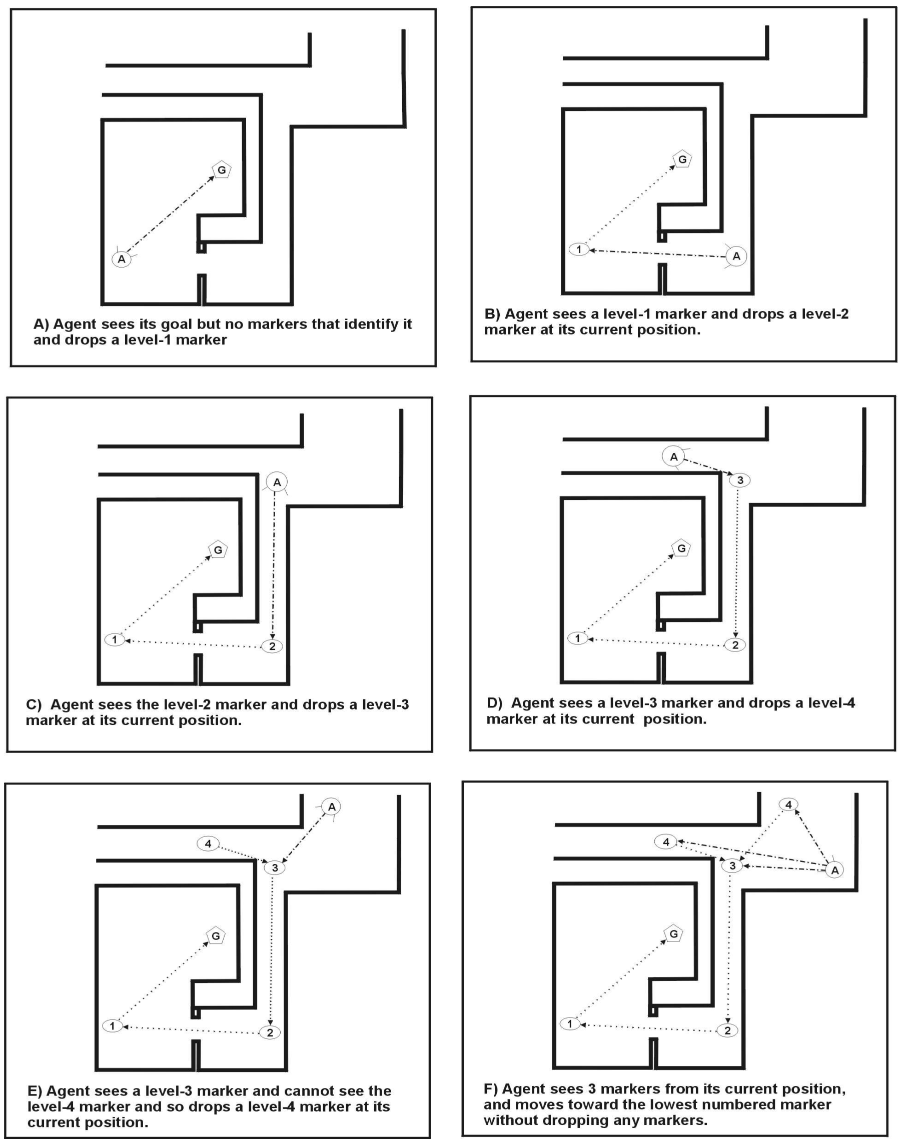
Markers in the basic techniques described above have indirect relationships to one another. For example, marking the start of a bottleneck may lead an agent further into a corridor, where it may continue to follow other markers placed by different agents at an earlier time or place markers itself. However, there are no explicitly defined relationships between these markers. While markers may even be uniquely labelled, there are no relationships between the labels used. In contrast, the markers in a real trail do not coincidentally mark individual phenomena, but have an explicit ordering to them. Analogously, *stigmergic trail-making* is the process of constructing and interpreting a purposeful trail of markers *cooperatively* to a useful location, by any number of agents over time as they explore the environment. A stigmergic trail emerges naturally out of agents' collective desire to improve the navigability of the environment. By following a stigmergic trail, agents should be able to repeatedly locate a discovered goal without internal world modelling, internal maps, or internal path planning.

The process of stigmergic trail-making is best illustrated by way of an example. Consider the situation in Figure 2. A primitive stigmergic trail is created when an agent perceives a goal and drops a marker on the ground at its own current location (Figure 2a). The dropped marker identifies a vantage point from which the goal should be visible to an agent standing at the marker's location, and on its own, would allow an agent not within perceptual distance of the marker to be able to find the goal faster. By itself, a trail this rudimentary is of limited utility, since it only serves as a visual cue to agents that have an unobscured view of the marker, up to the limits of their perception. To make the trail more sophisticated, it is necessary to extend the trail, forming a chain of markers that will ultimately end at the goal. We must be able to know the placement of each marker in the trail, and we use a numeric value associated with a heterogeneous marker to represent the marker's position in the trail (in this case, 1) to do so.

Building a trail from a goal marker involves agents successively recognizing situations where the trail can usefully be extended from what can currently be perceived. Figure 2b illustrates this extension over time as an agent (possibly the same agent that started the trail, possibly not) sees an end-point of the trail (the 1 marker) and drops a second marker. The second marker dropped is assigned a value one higher than the marker that the agent has observed (in this case, 2), indicating an extension of the trail to that point. As the process repeats, possibly through the actions of many different agents, the trail gradually lengthens (Figure 2c, d).

To minimize clutter, agents drop markers only under constrained conditions. When the agent perceives the goal (or a marker already on the trail), it checks for the presence of stigmergic markers already serving as attractors to this target. If no markers are visible, the agent drops a marker with a value one greater than that of the marker it perceives. Ultimately this can create multiple useful trails due to the limitations of agents' perception: when a new branch is created, it will be because the remainder of the trail is out of sight, and will serve to guide agents from a different location from which the trail was not previously useful. This is illustrated in Figure 2e, where an agent sees the end of a trail, by perceiving a 3 marker but no 4 marker, and so extends the trail in another direction. A 4 marker exists, but is not within the agent's perception, and the trail is now useful from two incoming directions. The increased usability of this trail arises directly out of its construction by multiple agents.

In keeping with the objective of limiting marker clutter, an agent only extends the trail when it perceives what appears to be its true end. If the agent perceives other markers from its location, it only drops a marker if the goal or a higher valued marker is not also visible. There are never any truly redundant markers, since a marker is not dropped if one serving the same purpose is perceived, and if such a marker cannot be perceived, it is necessary. For example, in Figure 2f: the agent perceives marker 3 and two marker 4's; it knows that marker 3 is not the end of the trail (because it



A) Agent sees its goal but no markers that identify it and drops a level-1 marker

B) Agent sees a level-1 marker and drops a level-2 marker at its current position.

C) Agent sees the level-2 marker and drops a level-3 marker at its current position.

D) Agent sees a level-3 marker and drops a level-4 marker at its current position.

E) Agent sees a level-3 marker and cannot see the level-4 marker and so drops a level-4 marker at its current position.

F) Agent sees 3 markers from its current position, and moves toward the lowest numbered marker without dropping any markers.

Fig. 2. Stigmergic trail making

perceives a 4 marker), and that its vantage point is not far enough away to warrant dropping another marker (since existing markers already supply pertinent navigation information from the agent's current location, or the 4 marker would not be present).

In the case where occlusion or some other factor makes perceiving a marker difficult, and another marker gets dropped because of this, this is entirely desirable: another agent in the same position is likely to suffer similar occlusion or perceptual difficulties and such a marker would be of assistance. To deal with cases where occlusion can occur due to the movement of other robots, we do wait a short time before dropping a marker to take into account the dynamic nature of the environment.

These conditions allow an agent to simply follow a trail without dropping new markers when there is enough navigation information available to do so, and assist in improving the trail when enough information is not available). Agents utilize the trail to locate the goal at the other end by following the trail of markers, always moving toward the lowest numbered marker visible. Since the markers are dropped based on visibility, the agents are able to consistently follow one marker to the next to locate the goal. This also helps minimize the length of a path followed by an agent - if an agent perceives a 3 and a 4 marker, for example, the agent will not bother moving to the 4 marker.

The two important features to emphasize in this approach are a) that this trail-building occurs collaboratively; and b) that it occurs over time while agents follow existing trails and otherwise navigate through the environment. In the case of several goals in reasonably close proximity, the trails may connect. This may cause an agent to perceive more than one lowest valued marker: in this case an agent will be attracted to the marker in closest proximity at any time, so may start following toward one goal and end up at another. Either way, however, the agent reaches a goal, which is the point of the process.

In our implementation, we did not have these markers fade, in order that we could explore the maximum advantage of being able to make and use trails. However, such markers could be built to fade where goals change. In this case, since markers are dropped at different times, with higher-numbered values extended from lower-numbered markers, the useful lifetime of a marker would have to be higher for markers that were closer to the goal (i.e., lifespan in inverse proportion to marker number). In cases where trails were broken due to marker fade, partial value could still be obtained through the portion of the trail that was remaining (that is, the agent could still be coherently led partway to the goal), and if a trail were to be rebuilt from the goal it would eventually intersect the still-existing portion. Like ant trails, mechanisms could also be put in place to allow agents to extend the lifespan of these markers as they traversed them.

The four methodologies for dropping and responding to markers described in these subsections form two complementary halves of stigmergic navigation. Bottleneck and local maxima markers serve to promote exploration and allow agents to more quickly locate an undiscovered goal somewhere in the agents' environment, while stigmergic trail and local minima markers facilitate subsequent trips to a goal once it has been discovered. Having described the basic principles followed by these techniques, we now turn to more detailed information about implementing these in a software environment.

4 Implementation

Agents in this work were implemented using a schema-based control approach. Schema-based behaviours divide control into *perceptual schemas* and *motor schemas* that are tailored to specific aspects of an agent's environment [34, 17]. Motor schemas are responsible for manipulating the agent's effectors to accomplish well-defined tasks, such as moving toward a visible attractor. When activated, a motor schema produces a vector of parameters to physically control a mechanical agent (e.g. desired direction and velocity with which the agent should move). The output of a motor schema can be weighted by the overall strength of the schema's activation, allowing a blended response where several different motor schemas may be applicable at the same time.

Motor schemas are activated either by being invoked by motor schemas that are already active (a collection of schemas properly termed an *assemblage* allows a series of activities to unfold, with each motor schema activating the next, for example, and can also act in a subroutine-like fashion), or by perceptual schemas. Perceptual schemas act as filters on information coming from sensors, looking for perceptions that will be of interest to particular motor schemas (or assemblages of schemas) to which they are connected. In our implementation, for example, the *percept-marker* perceptual schema is responsible for using an agent's sensory apparatus to detect when a marker is visible and activating interested motor schemas. Information provided about the marker, such as its location, provide an activation strength to the motor schema or assemblage, allowing it to affect the robot's effectors.

More sophisticated behaviours such as marker dropping are handled in our implementation by assemblages. For example, a *drop-marker* assemblage consists of motor schemas most relevant to placing a marker of some type on the ground, and is activated when the *percept-drop-marker* perceptual schema returns a value of *true*, and deactivated otherwise. Our implementation allows as many motor schemas as necessary to be active at any given time, and their vectors are normalized according to the activation strength of the motor schemas and combined into a single normalized vector for orientation and velocity that is then followed by the robot.

The agents operated by these control mechanisms inhabit a simulated world provided by the computer game Half-Life. This simulation provides a large-scale, three-dimensional environment with realistic physics. In addition, the environment has a first-person view that is useful for observing the agents' behaviour during experiments. To allow agents to mark their environment, a pre-existing game object (a small satchel) was adapted to function as a basic marker type. Agents were given an unlimited supply of these items to drop on the ground as required. The agents and their constituent schemas were implemented in Visual C++ as part of an overall agent architectural framework and designed to allow for dynamic reconfiguration of agent assemblages, schemas and associated gain values [14]. This object-oriented class framework allows for new motor and perceptual schemas to be developed that leverage existing code as much as possible.

An agent's decision making method is invoked by the Half-Life simulator a variable number of times per second (depending on the system load) to respond to changes based on perceptions provided by the simulator. Our implementation hooks this decision-making method to each perceptual schema, which regard the environment, invoke motor schemas and assemblages, whose outputs are combined into an overall response.

A number of perceptual schemas were developed to detect the conditions under which various marker types should be dropped. In addition, motor schemas were implemented to cause agents to react appropriately to the presence of the various marker types employed, according to the descriptions provided in Section 3. For example, the *avoid-marker-local-maxima*, alternates between moving the agent toward and away from a local maxima marker.

5 Evaluation

To examine the improvement in navigation by a team of agents using the stigmergic techniques described here, a series of experiments were conducted. In each of these, a team of six agents all employing the same combination of marking techniques were placed in a three-dimensional environment, with each agent's objective being to travel to an initially unknown goal while using its marking technique(s). Upon reaching the goal, an agent was moved back to a starting position to repeat this process, gaining the advantage of markers that had been placed by itself and others in the environment in the interim. The number of times the goal was reached in a 30 minute period was tracked, along with the time required to initially discover the goal, as well as other statistics. In this situation, a more helpful marking technique (or combination of techniques) should support a higher number of traversals to the goal over a particular time period.

The specific three-dimensional domain chosen was *Crossfire*, one of the standard maps packaged with Half-Life. A simplified two-dimensional overhead view of this domain is depicted in Figure 3. This environment was chosen because it displayed all the difficulties associated with most indoor environments. Beyond providing significant path complexity and ample opportunity for the production of incorrect paths, it includes open areas, box-canyons, stairwells, ramps and elevation changes. Since the map is made for challenging human multi-player interaction, it is designed to be non-trivial to navigate. As such it contains many areas where limited visibility makes reactive navigation by agents particularly difficult.

The size of the world defined by a Half-Life map is 8192 units in height, width, and depth. Space in this environment is measured in terms of an abstract *unit* that is 1/8192 the length of a dimension. There is thus no direct relationship to any measurement in the real world. However, an agent in Half-Life is intended to be roughly human-sized, and is defined by a bounding rectangle 72 units high, 32 units wide, and 32 units deep. Common-sense approximation to the real-world equivalent of a unit would thus be about an inch, assuming a human is six feet tall. In referring to the environment, however, we use the same unit term used in Half-Life in order to be consistent with other research employing this domain.

In order to accurately measure the impact of each marking method, the environment, goal location and agent starting positions were consistent throughout all experimental trials, and all trials were run on the same machine (a Pentium III - 933 MHZ computer, running Windows 2000 Pro Service Pack 3 with 512 MB RAM). Agents began each trial in either Room 6 or Room 7 (noted on Figure 3). Two starting rooms were used so that agents would not interfere with one another excessively due to overcrowding at the start of a trial. Since the two starting positions are positioned symmetrically, navigating to the goal from either should have no effect on results. The goal was located in a room on the far side of the environment (Room 3). As can be seen from Figure 3, agents had to navigate over ramps and stairs, and negotiate a number of doorways, open areas and box-canyons in order to find the goal. After locating and moving to within 40 units of the goal, a point was added to the running total of goals discovered in the trial. The agent discovering the goal was then transported back randomly to either Room 6 or Room 7 to attempt to find the goal again (gaining the advantage of any additional markers that had been placed in the meantime).

Experiments were performed using the simple stigmergic techniques described in Section 3, both individually and in combination, followed by more sophisticated stigmergic trail-making, and finally supplementing stigmergic trail-making with combinations of these simpler techniques.

The stigmergic techniques described in Section 3 employ a number of parameters: for example, marker lifespan and range of influence. To select appropriate values for this evaluation, we ran numerous initial trials varying these parameters. We chose the parameter values that appeared to be the most promising based on these initial trials. The specific values for these parameters depend on the marker types employed, and are detailed in the subsections below.

Each experiment involved 40 contiguous trials of 30 minutes each. In each trial, we tracked not only the real-world time involved for events such as the initial discovery of the goal, but also the number of agent decision making invocations - that is, the number of times the simulator invoked each agent to make a decision on an action. This was used as a secondary measure of time

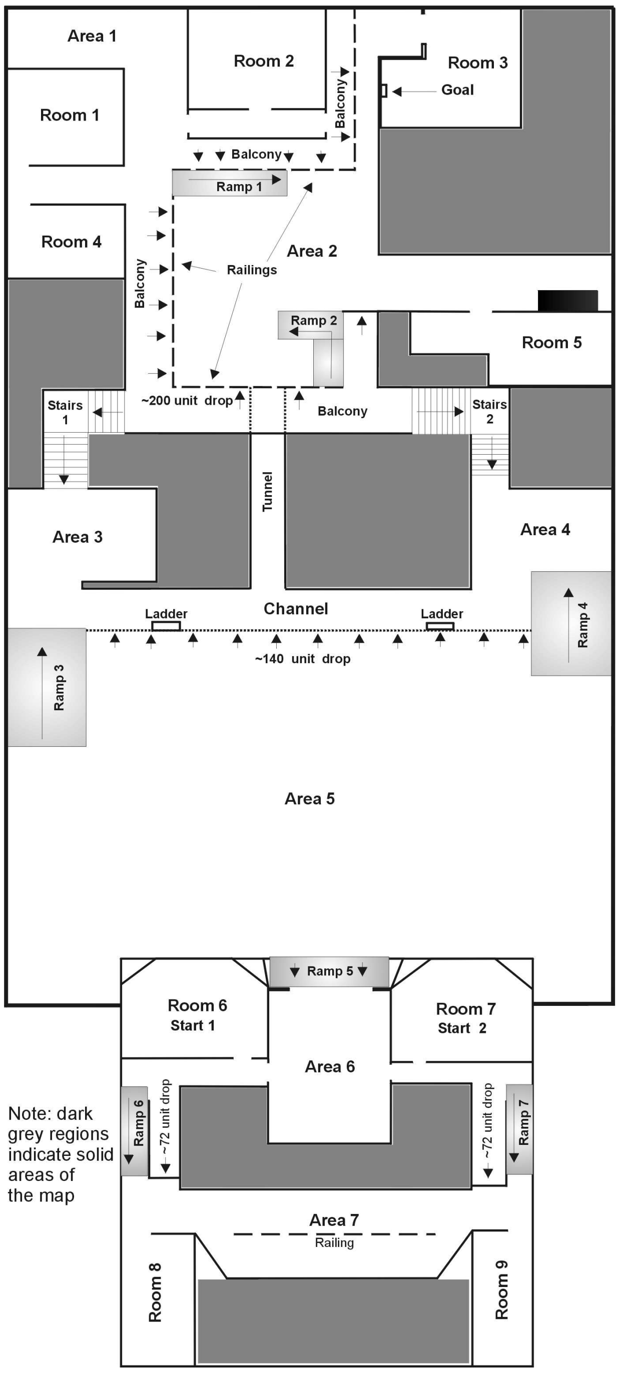


Fig. 3. Map of Experimental Environment

passing in the system, because in practice we found that the computational overhead of simulating some of the more complex marker types in combination had a side-effect of slowing down the entire simulation, resulting in fewer agent decision-making invocations over the same time period. This additional measure allows a comparison with issues of variability in simulation overhead removed.

The details of the specific conditions for each experiment and associated results are described in the following subsections. A table of all experimental results follows in Section 5.4.

5.1 Simple Marking Schemes

We began by experimenting with simple marking schemes and combinations of these. To establish a baseline, we ran a set of trials with a team of reactive agents using no markers and relying entirely on random search to locate the goal. In this case agents discovered the goal 161 times in 40 trials for an average of 4.03 goals per trial (with a standard deviation for goals reached in a trial of 1.67). It took them on average 592.35 seconds (with a standard deviation of 362.25 seconds) and an average of 57982.88 agent decision-making invocations to initially discover the goal. The agents' best performance in locating the goal repeatedly in a single trial was 8. The best time to initial discovery was 89.98 seconds over 8820 decision cycles.

We then performed experiments examining the simple marking techniques, and combinations of these, in order to compare them to this baseline. The results of these experiments along with detailed experimental conditions are presented in the subsections that follow, and are summarized in Figure 4.

Bottleneck Markers

Having established a baseline, we then explored the effects of agents that employed bottleneck markers (Section 3.1), to lead agents out of box-canyons (Section 1). The *percept-drop-marker-bottleneck* perceptual schema was configured in this experiment to return a true value whenever the agent was standing in a doorway or a hallway, unless a marker could already be seen within 60 units of the agent's current position. This perceptual schema was used to activate the *motor-drop-marker-bottleneck* motor schema, which caused the agent to drop a marker on the ground at its current position.

Another perceptual schema, *percept-marker-bottleneck*, was configured to maintain a dynamic list of markers visited by the agent. A marker was considered to be visited when the agent was standing within 40 units of the marker. These visited markers were purged from the short-term list after 30 seconds had elapsed between the agent's visit to the marker. This was intended to discourage cyclic behaviour and cause the agent to proceed along a trail of markers more deliberately (as described in Section 3.1). Otherwise, an agent

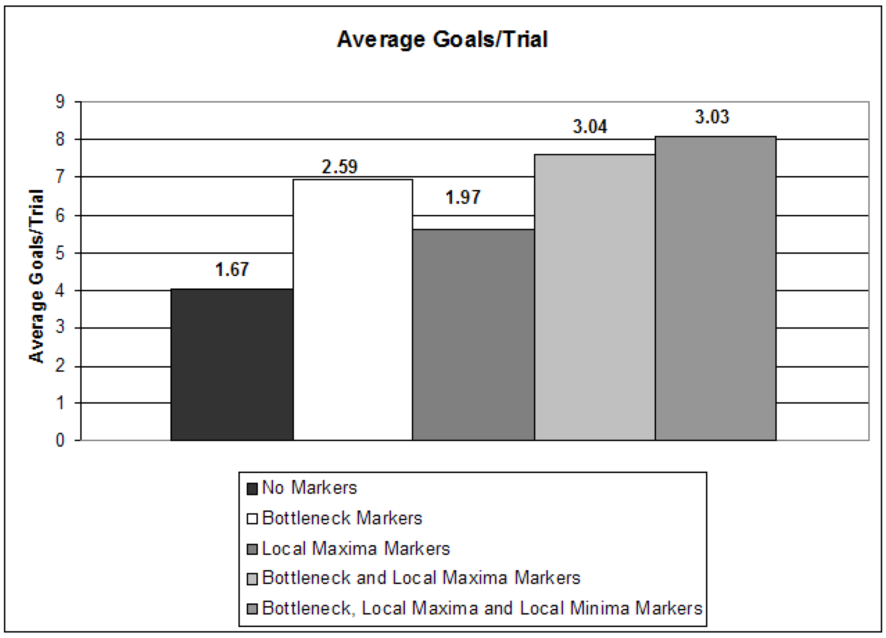


Fig. 4. Comparison of average goals/trial (the standard deviation for the number of goals/trial is indicated atop each bar)

might move back and forth between markers if it happened to turn around and see a previously visited marker again.

The *motor-move-to-marker-bottleneck* motor schema was also configured to cause the agent to move toward a visible bottleneck marker until it was within 40 units of the marker. At the point that the first marker was reached, the motor schema continued to move the agent toward any subsequent markers as long as the agent detected that it was still in a doorway or hallway. As soon as the agent detected that it was no longer in a narrow passageway (after reaching a marker), the *motor-move-to-marker-bottleneck* motor schema was suspended for a period of 30 seconds (during which any markers in its list of visited markers are gradually purged).

The agents in this set of experiments located the goal 278 times in 40 test runs for an average of 6.95 goals per trial, with a standard deviation of 2.59 (Figure 4). On average these agents discovered the goal for the first time in a trial within 463.67 seconds (with a standard deviation of 241.87 seconds), taking 43917.58 agent decision-making invocations to do so. The minimum time to initially locate the goal in any trial was 211.86 seconds and 20074 decision-making invocations. The most goals reached in a single trial was 12.

As can be seen, just marking bottlenecks alone resulted in significant improvements in the total goal count, the average goals/trial, and even the average time to initially locate the goal. The average time to locate the goal for the first time by any agent in the group was reduced by 21.72 percent. The average number of decision-making invocations to discover the goal was reduced by 24.26 percent.

The average goals discovered per trial was increased by 72.67 percent. Accordingly, the total goals for all trials was increased by the same percentage. In addition, the team of agents using bottleneck markers located the goal more often in the majority of trials. The second bar from the left in Figure 4 illustrates these findings in comparison to other techniques.

The standard deviation for goals per trial, at 2.59, was wider than that of agents not employing markers. The increased variability in goals is perhaps due in part to the fact that these markers are dropped without the agents knowing where the goal is located in the environment. As a result, the possibility exists that many (or all) of the agents might be initially led in a completely incorrect direction, slowing their time to find the goal. However, since a marker no longer affects an agent once the agent has visited it (that is, once the agent comes within a few units), the effect of being misled would be temporary. In any case, it is equally probable that the marker would in fact lead the agent in a better direction than a worse one (accounting for the variability in goals found).

Local Maxima Markers

We then examined the effect of solely using local maxima markers as described in Section 3.2. Agents were compelled to drop a local maxima marker at their current position when no goal was visible, no walls were nearby (within 160 units), and no local maxima markers were already present within a range of 500 units. A dropped marker was set to have a lifespan of 150 seconds.

Agents alternated between moving toward local maxima markers and avoiding them. When first perceiving a local maxima marker, while in a local maxima marker-affinitive state, an agent moved up to the marker's position. Once the agent physically reached the marker (moved within 40 units), the agent transitioned to avoiding local maxima markers. The local maxima marker *avoidance* state persisted for 30 seconds, during which the agent had an opportunity to find an exit from its current area (since it had nothing interesting in it). To allow for a certain freedom of movement, agents are only repelled by local maxima markers when they are in within 120 units. This has the effect of keeping them to the edge of the area they are in and closer to any exits from it, without trapping them against one particular side. If the agent does not find an exit, it is eventually drawn back into the open to try again upon becoming, once again, local maxima marker-affinitive.

The use of local maxima markers improved performance over using no markers in the majority of trials. The agents in this experiment discovered the goal a total of 224 times in 40 trials, for an average of 5.6 goals per trial with a standard deviation of 1.97 (the middle bar in Figure 4). The highest number goals found in a single trial was 11, and the average time taken to discover the goal for the first time was 533.48 seconds (with a standard deviation of 242.33) over 52135 decision-making cycles. The best time to reach the goal over all trials was 173.77 seconds over 17029 decision-making invocations.

In addition, the time to initially locate the goal was improved on average by 9.94 percent over using no markers. The average number of agent decision-making invocations to discover the goal was reduced by 10.09 percent. The number of goals discovered in total was increased by 39.13 percent over no markers. This was not as strong an improvement as gained by marking bottlenecks. However, this is not surprising, since local maxima markers work more indirectly, by pushing agents away from uninteresting places rather than actively drawing agents to new regions.

Bottleneck and Local Maxima Markers

We then examined the combined effect of bottleneck and local maxima markers, by allowing agents to employ both as per the conditions of the previous two experiments. Agents followed the additional restriction of not dropping local maxima markers within 150 units of bottleneck markers, and were configured to value bottleneck markers more strongly than local maxima markers via a higher gain value, so that the presence of a local maxima marker would not prevent them from moving to a simultaneously visible bottleneck marker.

A team of such agents located the goal 305 times in 40 trials for an average of 7.63 goals per trial, with a standard deviation of 3.04 (second bar from the right in Figure 4). The highest number of goals in a single trial was 13, and the best time to initial discovery was 244.12 seconds with an average time taken to discover the goal for the first time at 567.02 seconds (with a standard deviation of 222.61 seconds) or 51697.6 agent decision-making invocations.

The average goals per trial was increased over no markers by 89.44 percent using both marker types. This is compared to 72.67 percent for bottleneck markers and 39.13 percent for local maxima markers.

Adding Local Minima Markers

To this point, we have not yet introduced the use of local minima markers. This is because in the placement of the goal in the environment (Figure 3), there is no immediately occurring local minimum, making it unlikely that such markers would be helpful alone. Other marker types, however, form multiple intermediate goals that dynamically create local minimum situations, and so local minimum markers were expected to be more suited to augmenting these other types. To study this, we added local minima markers to the previous combination, allowing all of the simple techniques to be examined in tandem. Like local maxima markers, the fade period for local minimum markers was set to 150 seconds.

Unlike the other types of markers, local minima markers do not just generally attract or repel agents. Instead they instruct the agent to jump in the air upon moving sufficiently close to these markers. This is because the major local minima condition occurring in this domain were railings and other forms of low obstacles. For example, the balconies overlooking Area 2 of Figure 3, are lined by railings that can block agents attempting to move toward markers lying in Area 2. Agents coming up the stairs from Area 3 can sometimes see an attractive marker placed by another agent in Area 2 below them. However, when the agent attempts to move toward the attractive marker, the railing blocks the agent's path. In order to handle this and other low obstacle situations, it is possible for an agent to jump over the railing rather than remaining stuck or having to navigate the long way around the railing.

Using bottleneck, local maxima, and local minima markers in combination resulted in 323 goals in 40 trials for an average of 8.08 goals per trial (with a standard deviation of 3.03). As the rightmost bar in Figure 4 shows, this was a significant improvement over agents using no markers, but only a small improvement over that of the other marking techniques.

5.2 Stigmergic Trail-Making

Having examined the effects of the simplest techniques, we can now compare these to the more sophisticated trail-making technique explained in Section 3.4. As illustrated by the fourth bar from the right in Figure 5 (the bars before this are those from Figure 4 reproduced proportionally), stigmergic trail markers increased performance substantially over agents using no markers and that of all marking techniques examined previously. The average goals per trial for a team of agents using these markers resulted in more than 10 times the number of goals than a team of agents using no markers at all.

A team of agents using stigmergic trail markers located the goal 1828 times in 40 trials for an average of 45.7 goals per trial, with a standard deviation of 48.14 goals. On average they located the goal for the first time in 462.19 seconds of first appearing in the environment, with a standard deviation of 229.9 seconds. The most goals reached in a single trial was 182, and the fastest time to discover the goal was 103.62 seconds or 10213 decision making cycles for the agent that discovered the goal.

During the course of the experiment, a trail of markers was gradually built up, working backward from the goal after it was first discovered, and the trail branched appropriately where its end could not be perceived. Consequently, subsequent agents were eventually able to follow a trail of markers directly to the goal almost from the moment they appeared at the starting point (and improve the trail where they came upon its end from a novel location). The eventual result was a parade of agents heading (more or less) directly to the goal. This method exceeded the performance of agents using no markers in all trials and that of the combination of bottleneck and local maxima markers in 36 out of 40 trials.

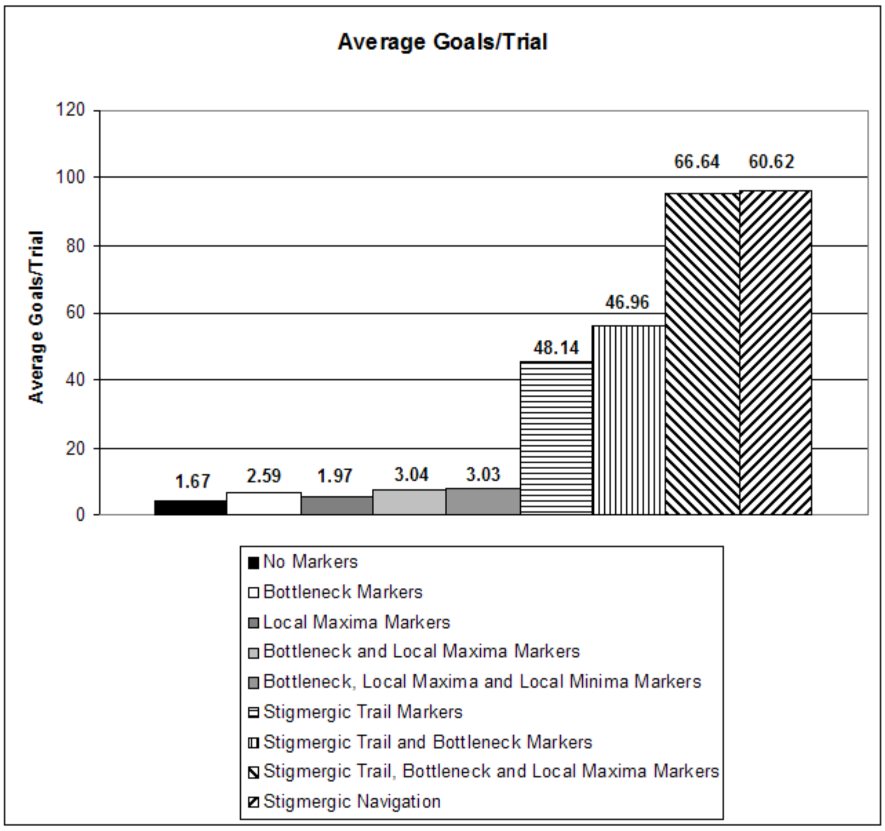


Fig. 5. Comparison of average goals/trial (the standard deviation for the number of goals/trial is indicated atop each bar)

As one would expect, the greatest factor affecting performance in these trials was how quickly the marker trail gets created. Once the marker trail was in place, the agents were able to follow it repeatedly and consistently until time expired. The poorer performances only occurred when this trail was not constructed early enough for the agents to benefit. When the agents failed to locate the goal early enough or frequently enough, their performance was comparable to using no markers at all. However, once the trail extended far enough back from the goal, the agents rapidly outpaced all others methods by a wide margin.

The average time to initial goal discovery was also improved by almost 22 percent over agents using no markers. One possible reason for this is that the first stigmergic trail marker dropped ensured that the agent discovering the goal did not wander inadvertently back out of the room in which the goal is situated. Without the use of these markers, this counterproductive activity was observed repeatedly (usually due to the influence of the *motor-noise* motor schema, which adds some randomness to an agent's movements but also on occasion when obstacle avoidance schemas directed the agent away from the goal when obstacles were present) With stigmergic trails, the agent drops a stigmergic trail marker upon first seeing the goal, allowing it to recover from even counterproductive behaviour more quickly. This is because even after wandering out of the goal room, the agent can often still see the trail marker it just dropped and is thus drawn back into the room directly.

5.3 Improving Stigmergic Trail Making with Other Marker Types

We then examined the integration of the previous stigmergic techniques to examine how they could improve stigmergic trail-making. The third bar from the right in Figure 5 illustrates the improvement gained by adding only bottleneck markers to the stigmergic trail-making process. Experimental data showed that this combined approach yielded better results than agents using no markers in 39 of 40 trials. At the same time, it resulted in a higher number of goals than stigmergic trails alone in 25 of 40 trials. This is also shown in that a total of 2253 goals were reached across all trials for an average of 56.33 goals per trial. The standard deviation for goals reached per trial was 46.96. The average time to discover the goal for the first time was 508.15 (with a standard deviation of 259.44) over an average of 46896 decision-making invocations. This is a 14.21 percent decrease in time and 19.12 percent decrease in decision-making invocations to discover the goal compared to using no markers.

Using both markers in combination resulted in 13 times (1299 percent) more goals being located on average per game versus agents not using markers, compared to 10 times more on average using stigmergic trail markers by themselves. The improvements gained here over using stigmergic trail markers alone are due to the bottleneck markers promoting greater coverage of the environment by drawing agents to different areas periodically. Once the stigmergic trail has lengthened a certain distance back from the goal, the faster other agents notice the trail, the better. Bottleneck markers help in this regard by drawing agents back and forth from room to room more regularly than they do when wandering randomly. Thus, once the trail is sufficiently long, other agents see its end sooner and the complete trail gets built earlier as a result. Since the trail gets built earlier, the agents have a longer period of time in which to benefit from it, resulting in a higher performance overall. The second bar from the right in Figure 5 illustrates the very dramatic improvement that results from adding local maxima markers to stigmergic trails and bottleneck markers. The combination of these three marker types resulted in 3817 goals being reached in total over all 40 trials: 22.7 times more goals than a team of agents using no markers and an average of 95.43 goals per game (with a standard deviation of 66.64 goals). The average time to initially discover the goal in a trial was 525.3 seconds (with a standard deviation of 291.27 seconds) over an average of 30869.03 decision-making invocations. This is an 11.32 percent reduction in average time to locate the goal in a trial. More significantly, it is a 46.76 percent reduction on average in the number of decision-making invocations required to locate the goal. The large difference in these percentages is primarily due to the additional processor load incurred by using all three marking methods in combination, slowing down the amount of simulation time that is performed in the real time units we measure.

As with the previous experiment, the improved performance using the three marker types in combination is due to the stigmergic trail being constructed and extended faster than would otherwise occur. The addition of the local maxima markers promotes even wider coverage of the environment than bottleneck markers by themselves. Local maxima markers have the additional effect of drawing agents into the open initially where they can see a wider portion of their surroundings, and increasing the likelihood that one or more agents will perceive a trail end sooner.

These effects in combination contribute to the stigmergic trail being established earlier in the trial, allowing the team to benefit from its use for a longer period of time. This is also evident in that this combination of marker types resulted in a best performance in a single trial: reaching the goal 247 times. This is significantly higher than any previous marker combination. Finally, we added local minima markers, resulting in the complete combination we have termed Stigmergic Navigation. As can be seen in Figure 5, the increase in performance achieved by adding local minima markers was again a small one relative to other marker types. Using all marker types resulted in 3856 goals being reached in 40 trials for an average of 96.4 goals per trial (with a standard deviation of 60.62 goals). The average time to locate the goal for the first time in a particular trial was 490.11 (with a standard deviation of 262.24) in 24814.38 decision-making invocations. The former is a 17.26 percent decrease (compared to agents using no markers) in the average time to initially locate the goal, and a 57.2 percent decrease in the average number of decision-making invocations required to initially locate the goal, compared to agents using no markers.

There are several likely reasons that local minima markers did not result in a substantial improvement. First, the situation that local minima markers are intended to handle appears in only a small portion of the environment. Consequently, the opportunity for them to have a positive effect on the agents' performance is limited. Moreover, the agents themselves are fairly well-equipped to deal with local minima via introduction of noise into their movements as part of their behaviour-based design. Finally, the other marker types can themselves reduce the frequency of local minima in this experiment.

Observations of the agents in a number of experiments showed that the stigmergic trail leading to the goal was typically built along a path that bypassed the upper balcony where local minima are known to appear. Instead, the trail typically takes the agents across Area 5 (see Figure 3) to either Ramp 3 or 4, down the channel near the center of the map, down the tunnel connecting it to Area 2 before proceeding down the last hallway before the goal. As a result, the occasional positive impact of the local minima markers did not have a substantial effect on overall performance.

5.4 Summary of Results

Table 1 illustrates the total number of goals achieved across all 40 trials, illustrating the efficacy of employing no markers (*None*), Bottleneck Markers (*BTl*), Local Maxima Markers (*LocMax*), Stigmergic Trail Markers (*StigTrail*), all of these together (Stigmergic Navigation - *StigNav*), and the various combinations described in the previous subsections. This table shows the average real-world time (in seconds) and the average number of agent decision-making invocations required to find the goal the first time in a trial for the same combinations of marker types.

Table 1. Summary of Results: Total number of goals, average initial goal discovery time, and average number of agent decision-making invocations before initial goal discovery, by combination of marker types.

Marking Method	Total Goals	Average Time	Average AgentInv
None	161	592.35	57982.88
BTl	278	463.67	43917.58
LocMax	224	533.48	52135.00
BTl/LocMax	305	567.02	51697.60
BTl/LocMax/LocMin	323	467.67	24068.98
StgTrl	1828	462.19	45043.30
StigTrl/BTl	2253	508.15	46896.00
StigTrl/BTl/LocMax	3817	525.3	30869.03
StigNav	3856	490.11	24814.38

None=No Markers, BTl=Bottleneck Markers, LocMax=Local Maxima Markers, LocMin=Local Minima Markers, StigTrl=Stigmergic Trails, StigNav=Stigmergic Navigation

The standard deviations for the time taken to reach a goal the first time and the number of goals reached were both large. The former is due to the nature of reactive navigation itself - while areas that are covered can benefit from the effects of markers, the areas that have not yet been covered are still slow to navigate. While making a stigmergic trail should not intuitively increase the speed at which a goal is first found, there are some specific phenomena in this domain to account for improvement in cases where this does occur (Section 5.2). The standard deviation in the number of goals reached is directly connected to the variation in finding the goal the first time. In cases where stigmergic trails are built, the team is essentially following this trail repeatedly until time expires. If that trail is not built early, performance suffers correspondingly.

6 Discussion

In this chapter we have described a series of increasingly sophisticated stigmergic techniques for navigation in a three-dimensional domain, and have examined the efficacy of these in a software environment. By marking bottlenecks and local maxima, agents were able to more frequently discover an initially unknown goal. At the same time, through stigmergic trail-making, agents were able to greatly increase the frequency and ease with which they subsequently located a previously discovered goal, and this improvement was increased substantially by combining this with simpler techniques. All these methods are noteworthy in that the team of agents cooperatively marks the environment over time and shares knowledge of their explorations with one another, resulting in significantly better performance.

Despite having been examined in a software domain, the principles behind each of the techniques described here are all translatable to the physical world. For example, the heuristic of marking bottlenecks to deal with navigational difficulties such as doorways and hallways was useful because in this environment, it is difficult (if not impossible) for an agent to detect the existence of a doorway at a distance. Here, the markers allow the agents to place a more easily detectable object at the location of interest, lightening the sensory load on other agents. In a physical environment this presents similar difficulties, since it is likely that a robot would have to actually be at the doorway already in order to detect it (without a map and accurate localization). Using markers, the agent can "see" the doorway at a distance and utilize that information more readily.

The work presented here bears some relationship to the literature cited in section 2. The main difference between this work and those previously described [27, 28, 25] is that here markers are physically dropped in an environment (albeit a simulated one for the purposes of implementation), as opposed to being maintained and communicated between agents as part of larger world models. Moreover, these other methods require explicit communication to work. Agents therefore require communication hardware and a reliable network to share information. This makes such approaches unsuitable for many domains where direct communication might be intermittent or blocked, making stigmergy an attractive alternative.

The work of Sauter, Parunak, et al. [28, 25] has the added requirement of special place agents that must be evenly distributed throughout the environment ahead of any navigating agents. While having an advantage in that the approach can easily model the natural decay of pheromone trails, this additional factor makes this approach impractical for many real world environments. Although outside the scope of this work, it is worth noting that others are working on physical stigmergic mechanisms that allow for physical fading [35].

Finally, though the methods of Vaughan et al. [27] allow robotic agents to navigate to a goal location more quickly and reliably than otherwise, they do nothing to reduce the time it takes for the goal to be initially discovered. In contrast, our method to promote exploration, described in Section 3.1, seeks to allow the team to discover an unknown goal for the first time faster than otherwise. This is especially advantageous in applications where quickly locating a particular unknown goal is critical (such as rescue scenarios).

7 Challenges and Future Work

While the results presented here are interesting, there is much future work to be done in this area. Deciding when to drop and when to follow markers is critical to the effectiveness of stigmergic navigation. If agents drop markers indiscriminately, they introduce noise and distraction that can hinder rather than help agent performance. Here, we have concentrated on showing the advantages to be gained by being discriminate in choosing when to mark the environment in order to avoid marker clutter in the first place. However, several of the marker types fade in simulation to achieve the results here, which limits the effect of clutter in these marker types.

While in the real world, such fading would have to be implemented using physical mechanisms (e.g. chemical pheromones perceivable by a robot [35]), we are interested in both emphasizing the issue of reducing marker clutter as well as the use of physical markers that require less specialized equipment to perceive and manipulate. Our future work will thus entail the use of teams of agents that can remove as well as place markers, something that is more easily adaptable to the physical world given current technology. Adding the ability to remove a placed marker when it does not match the conditions for its placement is relatively simple on its own, and deals with the issue of marker clutter without the need for fading. This becomes a more involved strategy, however, when a marker being moved or removed has a relationship to others, as it would in stigmergic trail making. We are currently exploring a number of strategies for marker removal under changing goals that are intended to limit the damage done to a trail when markers are removed simply due to misperception or misinterpretation. The ability to remove markers also adds one more crucial behaviour to the set that make up a marking approach: an agent must decide how the removal of a marker alters its reaction to other markers. For example, if an agent has removed a bottleneck marker, there are likely additional markers upcoming that bear an indirect relationship to this and could also be candidates for removal - but at the same time, other upcoming markers may be completely independent.

Related to the issue of marker fading and clutter is the supply of markers themselves - our evaluation does not deal with a finite supply of markers in an agent. Instead, we focus on attempting to limit the need for an overly-large number of markers, both from the standpoint of clutter and the standpoint of a finite marking supply. Being able to pick up markers implies being able to re-use them, and thus the work described above should also support working with a finite marker supply.

In addition to exploring issues of when to drop markers (or remove them), the issue of *when to follow* markers is an important one to explore further. Agents need to balance exploitation of markers with performance of other tasks. If an agent simply follows the markers left by others, without additional exploration, the added value of using these mechanisms is limited, since the markers are not being deployed as extensively as they could. When using bottleneck markers, for example, we employed the simple mechanism of having agents ignore markers for a period of time after reaching one. This allowed agents to wander around the area that they presumably entered upon reaching a doorway for a time, and perhaps find another different doorway. Analogous mechanisms exist for other marker types, and these can certainly be improved.

Adapting these techniques to the physical world also introduces new problems in control: behavior-based robots must have a supply of marker types (or some substance used for marking), and must have the ability to drop (and remove) these items. Dropping a marker is a high-level control command in our approach, but would have to be expanded to physically implement this operation at a motor control level. Similarly, the agents would also have to be physically able to perceive these markers and distinguish between marker types. In simulation, perception has a bounded distance for accuracy, but this is still more accurate than perception in the real world.

More immediately, we intend to use the existing system to examine the effect of the number of agents on the utility of these techniques, as well as changing conditions in the environment. For example, it would be useful to study the performance of local minima markers in an environment that *forces* agents to negotiate them to get to the goal (rather than bypassing them, as sometimes occurred in these experiments). We also intend to experiment with changing conditions in the agents' behaviours (e.g. how long to ignore a bottleneck marker).

A natural extension of this work will be to adapt the markers described here to systematically explore an environment as opposed to locating an isolated goal. By giving markers unique identifiers, agents can pass marker lists that indicate the order in which they should be traversed to reach a goal. This approach is likely to be more resistant to localization errors than that of Vaughan et al. [27], since markers exist externally.

Ultimately, we would like to build agents that *learn* when to place and follow stigmergic markers, in order to minimize the placement of markers and maximize discrimination in exploiting markers. One approach to accomplishing this could include categorizing perceptual features into exemplar states to allow agents to generalize and learn the characteristics of desirable locations and update them dynamically. Agents could then use stigmergic markers to identify valuable locations more generally as they travel.

Another application of stigmergic markers that warrants exploration is using markers to assist agents in localizing cooperatively in conjunction with path planning-based navigation. This could be accomplished by having agents dynamically create landmarks in environments that lack easily identifiable signposts. In this approach, each marker dropped would encode the agent's beliefs about the coordinates of the location that the agent is marking along with a confidence factor. This would allow other navigating agents to use the marker as a potential landmark and as a means to localize.

While stigmergic techniques, in nature and otherwise, are not new, we believe that both they and the central concept they embody – storing knowledge in a distributed fashion throughout the world as opposed to inside an agent – is very under-appreciated in artificial intelligence. Modern research is only now beginning to truly recognize the power of storing knowledge in the world, and techniques embodying this concept, such as stigmergic navigation, will be seen more extensively in future.

References

1. Nehmzow U, Gelder D, Duckett T (2000) Automatic selection of landmarks for mobile robot navigation. Technical report UMCS-00-7-1, Department of Computer Science, University of Manchester
2. Baltes J, Anderson J (2003) Flexible binary space partitioning for robotic rescue. In: Proceedings of the IEEE international conference on intelligent robots and systems, Volume 3. Las Vegas, pp. 3144–3149
3. Reece, D, Kraus M (2000) Tactical movement planning for individual combatants. In: Proceedings of the 9th conference on computer generated forces and behavioral representation. Orlando, FL
4. Verth J, Brueggeman V, Owen J, McMurry P (2000) Formation-based pathfinding with real-world vehicles. In: Proceedings of the game developers conference. CMP Game Group, San Jose, CA
5. Werger B, Mataric M (1999) Exploiting embodiment in multi-robot teams. Technical report IRIS-99-378, Institute for Robotics and Intelligent Systems, University of Southern California
6. Brooks R (1991): The role of learning in autonomous robots. In: Proceedings of the 4th annual workshop on computational learning theory. Santa Cruz, CA, pp. 5–10
7. Balch T, Arkin R (1993) Avoiding the past: A simple but effective strategy for reactive navigation. In: Proceedings of the IEEE international conference on robotics and automation, Volume 1. Atlanta, GA, pp. 678–685
8. Fu D, Hammond K, Swain M (1996) Navigation for everyday life. Technical Report TR-96-03, Department of Computer Science, University of Chicago
9. Sgorbissa A, Arkin R (2001) Local navigation strategies for a team of robots. Technical report, Georgia Tech Robotics Laboratory, Georgia Institute of Technology

10. Mataric M (1997) Behavior-based control: Examples from navigation, learning and group behavior. Journal of Exp and Theor AI 9:323–336
11. Brooks R (1991) Intelligence without reason. In: Mylopoulous J, Reiter R (eds) Proceedings of the 12th international joint conference on artificial intelligence. Sydney, Australia, pp. 569–595
12. Balch T, Arkin R (1994) Communication in reactive multiagent robotic systems. Autonomous Robots 1:27–52
13. Baltes J, Anderson J (2002) A pragmatic approach to robot rescue: The keystone fire brigade. In: Smart W, Balch T, Yanco H (eds) Proceedings of the AAAI mobile robot competition. Edmonton, Canada, pp. 38–43
14. Wurr A (2003) Robotic team navigation in complex environments using stigmergic clues. MSc thesis, Department of Computer Science, University of Manitoba. Winnipeg, Canada
15. Groner T, Anderson J (2001) Efficient multi-robot localization and navigation through passive cooperation. In: Proceedings of the 2001 international conference on artificial intelligence. Las Vegas, pp. 84–89.
16. Arkin R (1998) Behavior-based robotics. MIT Press, Cambridge, MA
17. Pirjanian P (1999) Behavior coordination mechanisms: State-of-the-art. Technical Report IRIS-99-375, Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles
18. Balch T, Arkin R (1995) Motor schema-based formation control for multiagent robot teams. In: Lesser V, Gasser L (eds) Proceedings of the 1st international conference on multiagent systems. AAAI Press, Menlo Park, CA, pp. 10–16
19. Moorman K, Ram A (1992) A case-based approach to reactive control for autonomous robots. In: Proceedings of the AAAI fall symposium on ai for real-world autonomous mobile robots. AAAI Press, Menlo Park, CA
20. Balch T, Arkin R (1997) Clay: Integrating motor schemas and reinforcement learning. Technical report, College of Computing, Georgia Institute of Technology
21. Arkin R., Balch T (1997) Aura: Principles and practice in review. Journal of Exp and Theor AI 9:175–189
22. Brooks R (1986) A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation RA2:14–23
23. Holland O, Melhuish C (1999) Stigmergy, self-organisation, and sorting in collective robotics. Artificial Life 5:173–202
24. Perez-Uribe A, Hirsbrunner B (2000) Learning and foraging in robot-bees. In: SAB2000 proceedings supplement. International Society for Adaptive Behavior. Honolulu, HI, pp. 185–194
25. Parunak H, Brueckner S, Sauter J, Posdamer J (2001) Mechanisms and military applications for synthetic pheromones. In: Proceedings of the workshop on autonomy oriented computation. Montreal, Canada
26. Werger B, Mataric M (1996) Robotic food chains: Externalization of state and program for minimal-agent foraging. In: Proceedings of the 4th international conference on the simulation of adaptive behavior: From animals to animats 4. MIT Press, Cambridge, MA, pp. 625–634
27. Vaughan R, Stoy K, Sukhatme G, Mataric M (2002) Lost: Localization-space trails for robot teams. IEEE Trans on Robotics and Automation 18:796–812
28. Sauter J, Matthews R, Parunak H, Brueckner S (2002) Evolving adaptive pheromone path planning mechanisms. In: Proceedings of the 1st international

joint conference on autonomous agents and multi-agent systems. Bologna, Italy, pp. 434–440

29. Kube C, Bonabeau E (2000) Cooperative transport by ants and robots. Int Journal of Robotics and Autonomous Systems 30:85–101
30. Kube C, Zhang H (1994) Stagnation recovery behaviors for collective robotics. In: Proceedings of the IEEE international conference on intelligent robots and systems. Munich, pp. 1883–1890
31. Rumeliotis S, Pirjanian P, Mataric M (2000) Ant-inspired navigation in unknown environments. In: Sierra C, Gini M, Rosenschein J (eds) Proceedings of the 4th international conference on autonomous agents. Barcelona, pp. 25–26
32. Howard A, Mataric M, Sukhatme G (2002) Localization for mobile robot teams using maximum likelihood estimation. In: Proceedings of the IEEE international conference on intelligent robots and systems. Lausanne, Switzerland, pp. 434–459
33. Arkin R, Balch T (1998) Cooperative multiagent robotic systems. In: Bonasso R, Murphy R (eds) Artificial intelligence and mobile robots. MIT/AAAI Press, Cambridge, MA
34. Balch T, Arkin R (1998) Behavior-based formation control for multi-robot teams. IEEE Trans on Robotics and Automation 14:926–939
35. Kuwana Y, Nagasawa S, Shimoyama I, Kanzaki R (1999) Synthesis of silkworm moth's pheromone-oriented behavior by a mobile robot with moth's antennae as pheromone sensors. Biosensors and Bioelectronics 14:195–202