Using Cooperative Robots for Explosive Ordnance Disposal

James McLurkin
Massachusetts Institute of Technology
Artificial Intelligence laboratory
545 Technology Square, Room 921
Cambridge, MA USA 02139
jdmac@ai.mit.edu

Abstract

Some of the most promising uses for multiple robot systems involve searching for items or resources in unconstrained and unknown environments. The use of robots to dispose of unexploded ordnance is an excellent example of one such application. This work explores the possibility of using a community of many autonomous robots to clear mine fields. These ideas were developed using small microrobots in a laboratory environment to simulate larger robots working outdoors. The robots were designed to work together, with specialized communications and sensory hardware. The software uses a behavior-based approach to form a structured community from the local interactions of simple individuals.

1. Introduction

Communities of cooperative robotic agents working towards a common goal have the potential to perform a task faster and more efficiently then the same number of agents acting independently. One practical application for these robotic communities is the safe location and removal of land mines and other unexploded ordnance (UXO).

This work explores the minimum robotic system required for implementing such a community. In nature, ants are an excellent example of such a system, so we have used them as an inspiration and have taken some loose design ideas from observations and literature. The community structure for the robots is based on local interactions, with no central controller and no global communication. This type of system has many advantages, including scaleability and robustness [1]. We present a set of simple behaviors and discuss how they could be combined to facilitate searching tasks.

Searching for items is more efficient when the agents looking for the resource are able to match their distribution in the environment to that of the resource they are looking for. This keeps them from wasting time in less dense areas and spending more time where the number of items is high. We will demonstrate simple behaviors for spatial organization and dynamic density adjustments.

This work only address fully autonomous solutions. The only human action required is the deployment of the robots and the definition of a boundary defining the search area. Solutions that require operator intervention for remote control, location, navigation, or disarmament of the UXOs were not considered. System scaleability, operator survivability, and stealthy performance are all enhanced with an autonomous solution.

2. The Robots

Due to the extreme number of variables that affect the sensory and actuation systems of physically embodied robots, it is not yet possible to completely characterize them in simulation. This problem is accentuated when behavior based software techniques are used to control the robot, as the resulting actions are very closely coupled to the sensory inputs. In this work, microrobots approximately one cubic-inch in volume were used to simulate communities of full-sized robots large enough to operate outdoors and disable UXOs. Figure 1 is a picture of one of the microrobots. These robots are less expensive and more practical to use indoors than their larger counterparts [2]. Although this method is more accurate than a simulation, there are many differences between robots that are roughly a cubic inch in size and those that are on the order of a cubic foot. Some of these differences will present problems when attempting to scale these systems to larger robots, but the same fundamental approach should still be valid.

Each robot has three motors - two for locomotion and one to actuate the mandibles. The robots run on two treads, like a tank, which allows them to rotate about their vertical axis. The third motor engages the grippers which can pick up and carry the brass foil balls which represent UXOs.

2.1 Sensors

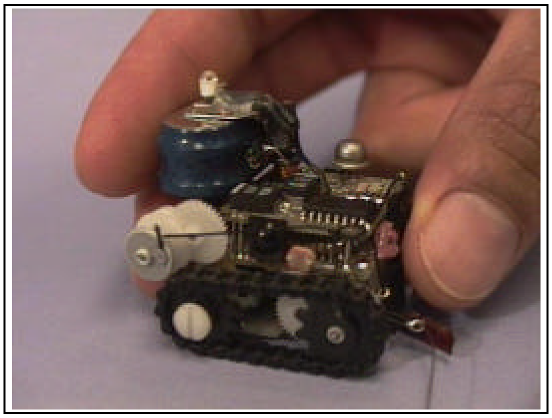


Figure 1. One of the microrobots. The light oval on the side is a light sensor, the black square right next to it is a IR receiver. The IR beacon emitter is at the top of the robot. The touch sensors are under the thumb, and the bump sensors extend beyond the bottom of the photo.

The robots used to perform these experiments are equipped with a suite of 17 sensors of different types to provide them with information about the environment and the surrounding robots [3]. For the work described in this paper, only four types of sensors were used; the infra-red (IR) receivers, the light sensors, the bump sensors, and the touch sensors.

On each side of the robot there is an IR receiver and a light sensor. The light sensors detect ambient light levels and can be used as a solar compass for navigation. The IR receivers detect four-bit communications signals from other robots and base stations. Each sensor has a field of view of about ninety degrees, so the robot is able to determine the direction of the source relative to itself. However, the sensors do not detect the range of the transmission source.

The front of the robot has two bump sensors and five touch sensors. The bump sensors detect collisions with stationary objects and other robots. Two of the touch sensors are built into the bump sensors. The remaining three are built into the mandibles. These sensors use conductivity to detect the presence of small balls of brass foil, which was used to simulate UXOs in all the experiments.

2.2 Communication

There are two IR emitters on each robot. The tag emitter is mounted on the front of the robot and has a range of about one inch. The signal is directional, so the transmitting robot has to be facing the receiver. The beacon emitter is mounted on the top of the robot and has a range of about one foot. The range of this signal is important and is defined as the communications distance. Its signal is omnidirectional, so any robot within range can detect it. There are also stationary IR beacons with an adjustable range from one to five feet. The base stations and beacon emitters transmit their signal twice a second.

All of the communications are local and untargeted. There are several advantages to this approach. The communication system can be a simple, low power design. This is accentuated with microrobots because the range requirements are on the order of inches, not yards or miles. Perhaps the greatest benefit is the reduction of communications bandwidth that each agent has to process. The number of robots that can be within communications range at any given time is limited by the amount of physical space around the receiver robot. If the communications distance is increased, more robots will be in range, and more transmissions will be received. Hence, bandwidth is limited by sensor range, not community size. This is important as it allows the population to be scaleable to any size while the processing requirements of individual agents remain constant.

2.3 Environment

The robots were run on a 10’ x 13’ environment, as shown in Figure 2. This environment was built to keep the robots off of the ground (and away from dangerous feet!) and to provide the conductive surface required for the robot’s touch sensors to operate.

Figure 2: The microrobotic environment. The white pyramids are moveable obstacles. The little black dots are the robots.

3. The Software

3.1 Behaviors

The software is a simplified version of Subsumption Architecture [4]. The interactions of many simple programs, or behaviors, control each individual robot. The interactions of these robots then lead to the desired community structure. A program consists of many behaviors running concurrently arranged in a hierarchy. The outputs from more important behaviors override, or subsume, the outputs from less important ones. In this work, a behavior reads a sensor, physical or virtual, and outputs a command to the motors or communication system. These simple programs work best when there is a good mapping between sensor space and behavior space. For example, a **move-to-light** behavior works well because there are light sensors on the robots, but a **move-to-position-one** behavior does not as there is no position sensor. Some sensory information can be inferred from other types of sensors, but it is often better to add the proper hardware.

There were no strict restrictions about what a behavior could or could not do, how it was written, or whether or not it could incorporate state. In order to keep the code modular, behaviors were made as simple as possible. An example of a simple behavior would be **move-forward** or **move-to-light**. An example of a more complicated behavior would be **sit-there-for-a-while** or **back-away-from-dropped-item**. Their functionality can be inferred from their names.

3.2 Behavior Sets

Often it is useful to change the set of behaviors at different stages in the completion of a task. For example, one set of behaviors might be useful for getting the robot to a resource-rich location, another for actually searching for the items, and then another to guide the robot home. These groups of behaviors are called behavior sets. This organization of behaviors and behavior sets is similar to the Alliance architecture developed by Lynn Parker. [5] A good example of a complete program is the game of Tag, which is shown in Figure 3.

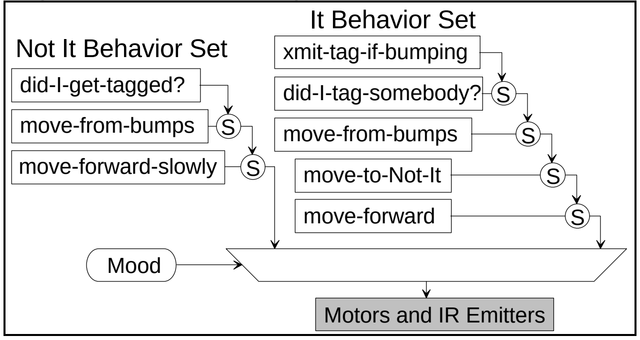


Figure 3: The behaviors needed to play tag.

3.3 The Game of Tag - An Example Program

The robots start up in the **Not-It** behavior set if they are on level ground or the **It** behavior set if they are tilted. The default behavior in the **Not-It** behavior set is **move-forward-slowly**. If the robot runs into an obstacle, the **move-from-bumps** behavior will become active, and back the robot away from whatever is in front of it. If the robot receives the **"tag"** signal on any of its IR receivers, it will stop moving and transmit **"I-got-tagged"** from its beacon emitter for three seconds. Then it will switch its behavior set from **Not-it** to **It**.

Not It Behavior Set

Behavior	IR Beacon Signal
did-I-get-tagged?	got-tagged-signal
move-from-bumps	<no effect>
move-forward-real-slowly	not-it-signal

It Behavior Set

Behavior	IR Beacon Signal
xmit-tag-if-bumping	<no effect>
move-from-bumps	<no effect>
did-I-tag-somebody?	<no effect>
move-to-not-it	<no effect>
move-forward	no-signal

Figure 4: Compact Pseudo Code for the Tag Example.

The responses of the lower three behaviors in the **It** behavior set can be inferred from their names. **Xmit-tag-if-bumping** transmits the **"tag"** signal from the tag emitter anytime the bump sensors are activated. The next behavior, **did-I-tag-somebody?** checks the IR receivers for the **"I-got-tagged"** signal. If this signal is received, the behavior set is changed from **It** to **Not-It**. This collection of eight behaviors is all that is needed to play Tag.

The complete set of behaviors that the robots use can divided into two groups based on the sensory inputs used. One group reacts only to other robots, the other reacts to everything else. In the tag example, the behaviors **move-from-bumps**, **move-forward**, and **move-forward-slowly** interact with the world and not with other robots. Other behaviors in this category but not used in the example are **move-to-light**, **keep-light-right**, and **pick-up-item**. These behaviors give the robot the ability to navigate around the environment and interact with items.

Behaviors that sense IR transmissions from the other robots allow for community structure. The tag program used **did-I-get-tagged?**, **xmit-tag-if-bumping**, **did-I-tag-somebody?**, and **move-to-Not-It**. Some other examples are **avoid-other-robots**, and **change-behavior-set-if-near-other-robots**.

The graphic representation presented for tag can be condensed in to the more efficient representation shown in Figure 4. The behaviors are listed in descending order of importance in the left hand column. The right hand column shows what signal the IR beacon emitter will transmit if the behavior is active. A <no effect> in the right hand column means that even if the behavior is active, the beacon signal will not be modified.

4. Simple Community Structures

Multiple robots with carefully designed behavior sets and behaviors can be combined to produce community structures. Since searching for UXOs was the ultimate goal, most of the structures that will be discussed here are used for spatial organization. However, this same approach has been applied to other applications, including programming the robots to play games such as Tag, Manhunt, Freeze Tag, and Hide Under Rocks.

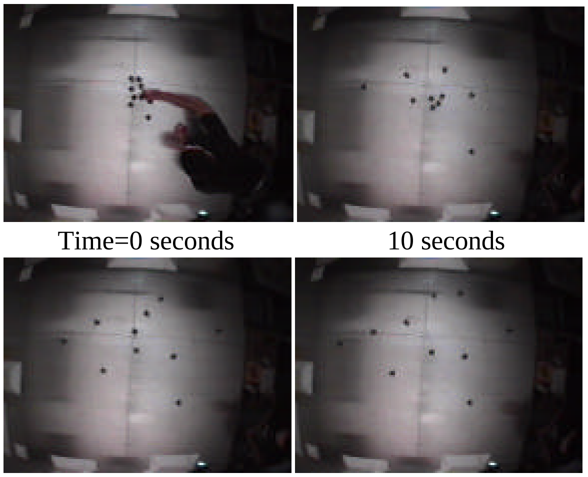
Behavior	IR Beacon Signal
move-from-bumps	<no effect>
move-to-other-ants	<no effect>
sit-there	ant-signal

Figure 5: Clustering.

Clustering is one of the simplest types of spatial organization. All of the robots transmit an identification signal twice a second. All of the signals are the same, as each robot has no sense of individuality. Each robot heads towards the first robot it sees. Eventually, all the robots are clustered into groups. If all the robots were within communications range of another robot initially, then there will be one final group. Otherwise, there will be several groups separated by at least one communications distance. These separate groups will be referred to as communications groups. This behavior worked well, but the robots tend to physically interfere with each other at such close ranges.

Behavior	IR Beacon Signal
move-from-bumps	dont-subsume
move-from-other-ants	dont-subsume
sit-there	ant-signal

Figure 6: Dispersion



Time=0 seconds 10 seconds

20 seconds 30 seconds

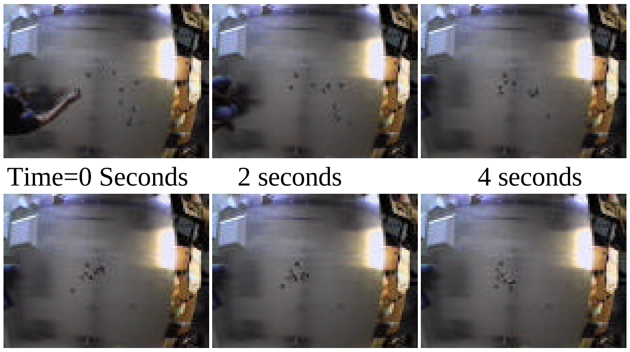
Figure 7: Dispersion Video Clips

Dispersion is the opposite of clustering. Again, the robots transmit an identification signal twice a second from their beacon emitters. Each robot heads away from any other robot it detects. Eventually, they are all separated by at least one communication distance. If an additional robot is added to the center of the group after the robots have spread out, a wave of spreading will propagate throughout the rest of the group until there is a communications distance in between each robot. The drawback to this behavior is that it forces the robots to lose contact with each other. If the search space is not bounded on all sides, robots that are left out of the communications group might become lost.

Behavior	IR Beacon Signal
sit-there-until-tilted	found-UXO-1-signal
touch-UXO	found-UXO-1-signal
move-from-bumps	<no effect>
move-to-the-ant-with-UXO-1	found-UXO-2-signal
move-to-the-ant-with-UXO-2	found-UXO-3-signal
move-to-the-ant-with-UXO-3	found-UXO-4-signal
move-to-the-ant-with-UXO-4	found-UXO-5-signal
move-to-the-ant-with-UXO-5	<no effect>
sit-there	no-signal

Figure 8: Relay Clustering

In relay clustering one central robot initiates the relay by transmitting an IR signal from the beacon emitter, usually because it has found an item of interest. Any robot within range of the first robot heads towards it while transmitting **“I-see-a-robot-with-a-UXO”**. Any robot within range of the second robot but not the first transmits **“I-see-a-robot-that-sees-a-robot-with-a-UXO”** and heads towards the second robot. Since the second robot is already heading towards the first one, eventually the third robot will find the initial robot. The layering can extend however many layers deep the situation demands. The number of robots that eventually respond to the initial



Time=0 Seconds 2 seconds 4 seconds

6 seconds 8 seconds 10 seconds

Figure 9: Relay Clustering Video Clips

transmission depend on the density of the robots, the number of layers in the algorithm, and the communications distance. Our experiments tested this technique up to five layers deep, and were able to get robots to respond from at least four communications distances away. The video clips in Figure 9 show the triggering of the central robot, the communications relay throughout the group, and the resulting clustering of most of the other robots in the environment.

Did not Receive Message Behavior set

Behavior	IR Beacon Signal
did-I-get-the-signal?	did-not-get-message-signal
sit-there	<no effect>

Received Message Behavior set

Behavior	IR Beacon Signal
sit-there	got-message-signal

Figure 10: Relay Information Transfer

Relay information transfer is the closest approximation of global communication present in the current system. One robot originates the transmission of information from its beacon emitter. If there are any other robots within range, they receive the transmission and begin relaying the same transmission. Eventually, every robot in the same communications group as the initial robot will receive the transmission. This worked a little better than relay clustering, as the robots were not moving. Almost all of robots within the communications group would eventually receive the message.

Leader Behavior set

Behavior	IR Beacon Signal
move-forward	leader-signal
stop-if-no-follower-signal	<no effect>

Follower Behavior set

Behavior	IR Beacon Signal
move-to-leader	follower-signal

Figure 11: Following

Having the ability to follow other robots can be useful. Care must be taken to make a stable chain of moving robots. The speed at which a robot travels is dependent on that particular robot's hardware and its battery life. If the leader is faster than the followers, the followers will lose sight and become lost. One type of following uses the beacon emitters on the followers to signal to the leader when it is time for it to move. The leader transmits **`“I’m-the-leader”`** as before, but only moves forward while it can detect the **`“I-see-the-leader”`** signal from the follower behind it. This communications feedback ensures that the follower will always be in range of the leader. This type of following can work for much longer distances, but eventually a communications error will leave the follower stranded.

Another type of following can be found in nature in leptothorax ants. [6]. The tag emitters on the follower is used to signal the leader when it is clear to move. The leader transmits **`“I’m-the-leader”`** as before, but only moves forward a short distance. The followers transmit **`“I-see-the-leader”`** as before, but this time for their short range tag emitters. If the followers get within an inch of the leader, this signal informs the leader that it is clear to move forward another short distance. Although much slower, this type of following ensures that the followers will not get lost.

Behavior	IR Beacon Signal
flash-lights	<no effect>
count-ants	<no effect>
sit-there	ant-signal

Figure 12: Measuring Density

Using their IR receivers, each robot is able to count the number of other robots in its immediate vicinity. Since the communications range is the same for all the robots, the number of robots counted can be used as a density measurement. (See diagram) The ability to measure the local density of robots is very useful. Dispersion and clustering can now be based on density and not on the communications distance. This allows for an intermediate form called swarming.

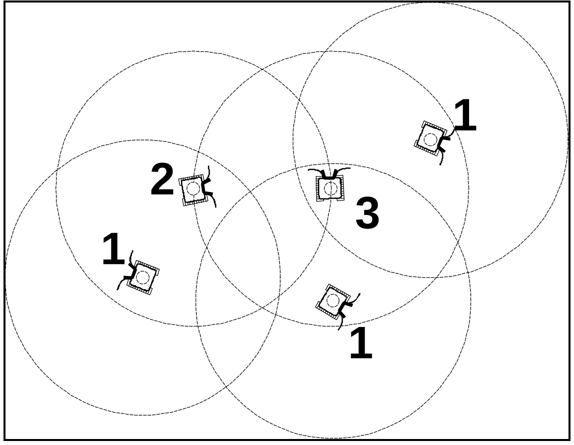


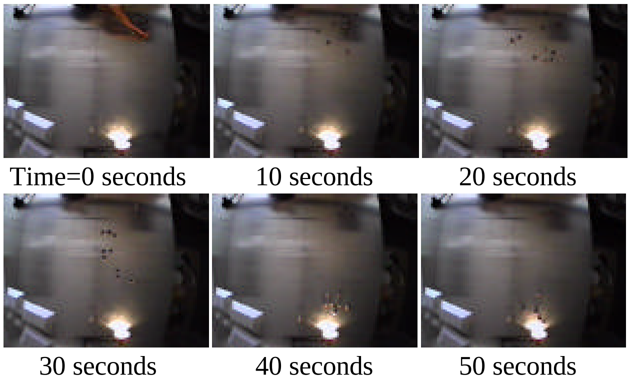
Figure 13: Density Measurement. The numbers next to the robots show how many other robots they can detect. Since the communications range is constant and uniform, the density of robots is count/communications area.

Behavior	IR Beacon Signal
move-from-bumps	<no effect>
flash-lights	<no effect>
signal-max-density	max-density-signal
signal-min-density	min-density-signal
count-ants	<no effect>
move-to-min-density	<no effect>
move-from-max-density	<no effect>
sit-there	ant-signal

Figure 14: Swarming

Swarming keeps the local density of each robot within specified bounds. If the density is too high, the robot tries to disperse. If the density is too low, the robot tries to converge. This maintains the coherence of the communications group, even as the robots move around the environment. Swarming is keeps a moving group of robots together, so it is always used in conjunction with some other kind of navigation behavior. Figure 15 shows swarming and contour following, a technique which is discussed in the next section.

The communications distance is very important to the functionality of this behavior. In the microrobots this distance was too small, so in order to maintain an adequate density to prevent robots from wandering off, they were too densely packed and got in each other's way. The working space around the robot has a radius of approximately one inch, while the communications distance is approximately twelve inches. A communications distance of four or five times that number would probably allow this behavior to be more effective.



Time=0 seconds 10 seconds 20 seconds

30 seconds 40 seconds 50 seconds

Figure 15: Swarming while contour following video clips.

5. Explosive Ordnance Disposal

There are four main issues for autonomously removing unexploded ordnance: selecting a removal strategy, bounding the search space, detecting the mines, and using cooperation to increase the efficiency of the search. The following sections discuss each topic in detail.

5.1 The Search Strategy

The simplest removal strategy uses expendable robots with built-in submunitions charges. The charges allow the robots to be detonated when they are in place over a mine or any other type of UXO. With this strategy, each robot is only responsible for finding only one UXO. When a robot has discovered its target, it stops on top of it and broadcasts the discovery to any nearby robots. This information can be used to dynamically adjust robot density and will be described further in the cooperation section. When all of the robots have each found a target, the entire group can be detonated simultaneously via remote control. The detonation signal could be a global signal or a communications relay from robot to robot

A more complex situation arises when each robot is able to disable many UXOs, either by manipulation or submunitions deployment. [2] One method of using manipulation is to collect the UXOs in a central location, then detonate or disarm them there. The process of gathering the mines and returning home with them is called central point foraging. This technique is only feasible with unexploded munitions, as autonomous manipulation of mines is beyond the abilities of current robotic systems. All of the microrobots used for these experiments did not have functioning mandibles or long-range navigational sensors. The lack of these two components makes implementing central point foraging difficult, if not impossible, so this method was not explored in this work.

A strategy that does work with either mines or UXOs involves the use of submunitions charges. A robot can place a small submunitions on or near a UXO, then begin searching for its next target. At the end of the exercise, all of the robots can return to the base, leaving only the submunitions behind to be remotely detonated. This approach meshed well with the abilities of the microrobots.

5.2 Bounding the search space

The area to be cleared of UXOs needs to be bounded. Otherwise, any robot that is separated from the communications group would drive off into oblivion. There are many ways to construct a boundary, depending on the sensory information available to each robot. Each method has tradeoffs between the complexity of the sensory systems needed, the amount of supporting equipment required, the size and shape of the target area, and the redundancy of the search - how many times the same area will be covered by multiple robots. With the

Figure 16: Dispersion with Obstacle Avoidance edge effects. The effects can be seen more clearly in the composite reflection picture on the right.

microrobots used for these experiments, three methods were explored; perimeter boundaries, contour boundaries, and point to point navigation.

Perimeter Boundaries

The simplest method of defining a perimeter boundary is to erect a physical wall around the search area. Radio, laser, ultrasound, or GPS can also be used to bound the area or endow the robots with a Cartesian coordinate system. These methods would allow the robots to search an area bounded by "virtual" boundaries, although they have tradeoffs with accuracy, supporting hardware, and terrain limitations. Any size and shape search area can be constructed, limited only by the maximum complexity of the boundary method used. With this type of system, robots tend to travel the same areas multiple times, providing good search redundancy.

For the microrobot simulations, physical walls were used to approximate this kind of boundary. The microrobots tended to spend more time around the boundaries due to obstacle avoidance edge effects, so behaviors were added to force then to move straight after a collision.

Obstacle avoidance edge effects occur because it takes the robots a little bit of time after a collision to back away and head back into the environment. This post-collision time increases the probability that at any given instant, a robot will be near a wall. Figure 16 shows dispersion with edge effects. The composite picture was produced by mirroring and flipping the original image about the boundary edges. The density of the robots along the horizontal and vertical center lines in the composite image is higher than the density over the rest of the area.

Random motion commands were added to the search behavior to provide better results. The robots do not have encoders, and even it they did, their treads slip quite a bit. As a result, they rarely move straight; there is usually some curvature. In a large enough area, this causes the robots to drive in big circles, which is not the most efficient method of searching for UXOs. Random motion also increases the search redundancy [7].

Contour Boundaries

A contour the robots can sense can also guide them through the mine field. This contour could be a geographic feature, such as a coast or a road, or an artificial feature, such as a laser. The operator provides a predefined starting and ending point. If the robots follow this edge, they would only clear a line of mines through the environment. In order to sweep out a larger area, we have added a swarming behavior. Some of the robots follow the contour and the others swarm around them, clearing a larger area. Division of labor between the robots that track the contour and the ones that swarm was accomplished by density. The robots in the middle of the swarm will be in the area of maximum robot density. If the number of robots around them exceeds a certain threshold, then they follow the contour. All the other robots in less dense areas swarm around the contour followers and search for UXOs.

The microrobots used a flashlight as their contour. The **`move-to-light`** behavior easily allows the robots to follow this beacon. Swarming was added to this basic to cover a larger search area. Each robot measured it's local density in order to determine whether to follow the beacon or to swarm around neighboring robots. Robots with density measurements higher than a set threshold followed the beacon while the others swarmed around them.

Point to Point Navigation

Equipped with the proper sensors, a group of robots can head from point to point, searching for mines along the way. If the robots are unable to sense the points at all times, a compass heading can be used for inter-point navigation. When a compass is used, dead-reckoning errors must be dealt with, as robots could get lost if enough errors are integrated over a long amount of time. A swarming behavior can be added to the programming to keep the robots in one communications group, yet still allowing them to sweep out a large area. If the points are arranged in a circle, or the robots are programmed to know the order of the signals, they can traverse the path multiple times, providing good search redundancy.

In the microrobotic simulation, sunlight was used to determine a compass heading, and IR beacons were used to mark the navigation points. The robots performed well when they could sense all of the required navigational points. Spreading the points out and using compass navigation from point to point introduced errors and some of robots wandered off.

5.3 Detecting UXOs

There are several technologies that can be used to detect unexploded ordnance. A metal mine will disturb a local magnetic field. Some mines are not buried, and can be found optically. Radar and sonar technologies can also be employed to locate buried ordnance. However, all of these methods have one thing in common, the robot needs to be in very close proximity to the mine to be able to detect it. This range will always be much smaller than the communication range of an individual robot. The microrobots use conductivity to detect crumpled balls of brass foil that represent UXOs.

Every sensor is prone to errors. With microrobots, sensor errors cause the robots to occasionally push the "mine" along the surface of the table a short distance before sensing it. In a real implementation, a sensor error would either cause the UXO to detonate under the robot or to remain undetected. Although a mine that has been detonated is no longer a threat, the explosion could compromise a stealthy operation. A mine that is missed by a single robot becomes the responsibility of the other robots in the group, so strategies with good search redundancy have a better chance of eventually discovering it.

5.4 Using Cooperation to Increase Efficiency

Cooperation can help to reduce the time required to complete the operation by taking advantage of the robots that have already found mines and using that information to aid in the search.

When each robot is responsible for removing a single UXO the ideal solution is to eventually have the distribution of robots exactly match the distribution of mines in the search area. When that happens, every robot will be on top of a mine.

The ability to measure the local density of robots can help match the distribution of robots to the distribution of mines. This is useful when the mines are unevenly distributed throughout the environment. Searching robots can sense the local density of other robots that have found UXOs. This measurement is used to cause the searcher to disperse if the number of discoveries in the area is greater than the maximum UXO density. If the measurement is less than maximum UXO density, then the searcher heads toward the successful robots to look for more mines in the area. The maximum density is determined by how many robots can physically occupy the same area. A lower number can be chosen based on operator observations, but this is not necessary.

A simple version of recruitment was used to vector robots from one location to another in the communications group. This could be used to direct robots away from areas with many discovered UXOs to areas that have not been as thoroughly searched. If each robot transmits a signal for a short time after it finds a UXO, that signal could be relayed through the communications group and be followed by a robot that has not found a mine yet.

When each robot can disable many mines, cooperation between multiple robots becomes more complicated. The ideal solution is no longer defined by where each robot is at the ending time. Instead, one needs to know where each robot was during the entire exercise.

One way for a robot to communicate where it has been is to leave markers, similar to trail markers in ants[8], or flower markers in bees[9]. This could be accomplished in a number of ways, such as laying scent trails [10] or depositing "bread crumbs" that other robots can sense. One of the best places to leave a marker is on top of a mine. This does several things; it stores information in the environment on the past locations of the robots and it allows other searchers to detect a discovered mine from a longer distance. If the marker can be placed near the mine, it would be even more useful if it contained a submunitions charge that can be detonated to destroy the UXO.

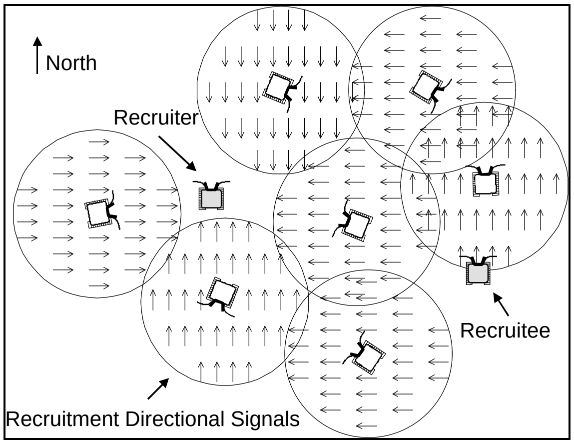


Figure 17: Recruitment in action. The gray robot on the right can follow the signals to get to the gray robot on the left.

If the robots are able to sense deployed submunitions charge markers in a similar manner and with a comparable range that they can sense other robots, then all of the cooperative strategies from the previous section (Each robot disables one UXO) can be applied to this strategy. If the range or type of detection are different, then similar techniques will work, when the range and subsequent density calculations adjusted.

The recruitment techniques from the previous example (each robot disables one UXO) can also be applied. A robot with a new discovery pauses and transmit its find for a short time before heading out to search again. This signal is then propagated throughout the group to direct other searchers towards the source of the new discovery.

6. Recommendations for Future Work

One of the biggest problems with the current system is the lack of long-range navigational sensors. This made it difficult to program the robots to search a specified area or to go to a specific location. Because of this, any robot that wandered out of range of a IR base station or the communications group would get hopelessly lost. A dead-reckoning system might help this problem somewhat, but a better solution would be to design a new sensor.

In behavior-based control systems, the robot's actions are closely related to their sensory inputs, so the absence of a proper "position sensor" was difficult to compensate for. Global positioning receivers, radio beacons or laser guidance systems might be effective long-range navigational aids. Another idea is to develop a scent marker which would allow the robots to leave trails behind them as they move around. They could then follow their own trails back to the group, or use trails laid by other robots to look for unexplored locations or to avoid heavily searched areas. The biological literature contains examples of scent trails that are used to communicate information. Many different species including ants, caterpillars, and bees all use this type of communication. Because the information is stored in the environment, an individual agent is able to communicate with many members of a group over a long period of time.

The current IR communications array can not sense the range of the signals it receives. This made it difficult for the robots to stay within communications range of each other without getting in each others way. A more sophisticated system would be able to maintain a minimum separation distance from other robots, while assuring that there is always a robot within a certain radius. If the farthest robot moves outside of the maximum radius, then it can be tracked and followed before it leaves communications range.

One of the goals for these experiments was to determine the simplest system required to accomplish these goals. Local communications were used to keep the robots and community structure simple. However, the added complexity of global information might outweigh the added cost and complexity of the hardware required [11].

Any approach to clearing ordnance will run into the problem of knowing when all the mines have been found. The only way to know for sure would be to know how many were there to start with. Simulations might be able to predict the ideal minimum time it would require, but real-life performance could take much longer. One way to determine when the operation was complete would be to continue to add robots until no new mines have been found for a given amount of time. That time would vary from system to system, depending on the environment and the robots.

There were only twelve robots available for these experiments, and on any given day, usually eight to ten were actually working. To fully characterize and understand the complex interactions that occur when many independent agents interact this population is not enough. Future experiments should incorporate many more agents, maybe ten times as many, in order to understand the interactions not only between robots, but between groups of robots.

The previous paragraph also illustrates a more pragmatic point. Reliability is an important issue when you are using one robot for an experiment, but it becomes critical when you have a dozen. A careful, simple, reliable design can save hundreds of debugging hours during testing and operation. Current systems are not yet capable of this level of robustness, but it is only a matter of time before sensor technology, control schemes, and robotic hardware allow us to meet these goals.

7. Conclusion

This paper describes simple behaviors that can be combined to enable robots to search for UXOs. Distributed control software based on local interactions allows for as many robots as needed while keeping the design and support systems simple. There was no top-down control, no central coordination, and no global communication. Individual robots were controlled using a behavior-based programming technique.

Explosive ordnance removal is one of the most promising uses of autonomous mobile robots. A group of robots cooperating has the potential to complete the task safely and efficiently.

The research was funded by a grant from the Naval EOD Technical Division.

8. References

1 Mataric, M.J. (1994), 'Interaction and Intelligent Behavior', *Technical Report AIM-1495*, MIT Artificial Intelligence Laboratory

2 Greiner, H., Angle, C.M., Jones, J.L., Shectman, A., Myers, R., 'Enabling Techniques for Swarm Coverage Approaches', *Second Symposium on Technology and the Mine Problem, Monterey,* CA 1996

3 McLurkin, J.D. (1995), 'The Ants: A Community of Microrobots', *MIT Undergraduate Bachelor's Thesis,* unpublished.

4 Brooks, R.A. (1986), ‘A Robust Layered Control System for a Mobile Robot’, *IEEE Journal of Robotics and Automation* **RA-2**, 14-23

5 Parker, L.E. (1994) ‘Heterogeneous Multi-Robot Cooperation’, *Technical Report AIM-1465*, MIT Artificial Intelligence Laboratory

6 Möglich, M. U. Maschwitz, and B. Hölldobler. (1974) ‘Tandem Calling: a New Kind of Signal in Ant Communication’. *Science*, 186: 1046-47

7 Adler F.R. and Gordon D.M. (1991), ‘Information Collection and Spread by Networks of Patrolling Ants’. *The American Naturalist.* Vol. 140, No. 3: 373-400

8 Hölldobler, B. and Wilson, E.O. (1990), ‘The Ants’, *The Harvard University Press, Cambridge, MA*

9 Giurfa, M. and Nùñez, J.A. (1992), ‘Honeybees Mark with Scent and Reject Recently Visited Flowers’. *Oecologia*. Vol. 89: 113-117

10 Russell, Andrew. (1995), ‘Laying and Sensing Odor markings as a Strategy for Assisting Mobile Robot Navigation Tasks’. *IEEE Robotics and Automation Magazine*, September 1995: 3-9

11 Parker, L.E. (1992), ‘Local versus Global Control Laws for Cooperative Agent Teams’. *AI Memo AIM-1357*, MIT Artificial Intelligence Laboratory