SANDIA REPORT
SAND2002-2418
Unlimited Release
Printed September 2002

Some Novel Design Principles for Collective Behaviors in Mobile Robots

Gordon C. Osbourn

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

Some novel design principles for collective behaviors in mobile robots

Gordon C. Osbourn
Sandia National Laboratories
Albuquerque, NM 87185-1423

Abstract

We present a set of novel design principles to aid in the development of complex collective behaviors in fleets of mobile robots. The key elements are: the use of a graph algorithm that we have created, with certain proven properties, that guarantee scalable local communications for fleets of arbitrary size; the use of artificial forces to simplify the design of motion control; the use of certain proximity values in the graph algorithm to simplify the sharing of robust navigation and sensor information among the robots. We describe these design elements and present a computer simulation that illustrates the behaviors readily achievable with these design tools.

I. Introduction

The use of simple agents to accomplish complex collective tasks (swarm intelligence or collective intelligence) has become a popular approach in areas such as robotics control, optimization searching, artificial life and the modeling of social insect behavior. Swarms of mobile robots have been proposed for a variety of tasks that are hazardous for humans to perform, such as hazardous chemical plume tracing and operation in radiation environments. The redundancy and distributed nature of the agent resources inherent in swarm approaches offer performance advantages over complex individual agents for certain types of tasks (e.g. real-time monitoring of wide areas) and also provide general robustness against hardware failures. There has been much research effort in the area of mobile robot swarms, both externally [1,2] and within Sandia National Laboratories [3,4].

In this work we consider a class of mobile physical agents that model simple, autonomous robots. The model robotic agents in this report are assumed to be identical and to have the following set of properties: finite state machines that select behaviors; limited (embedded) processing power; rf communication with a finite range; knowledge of self position (e.g. through GPS capability); mobility and steering; a small set of environmental sensors (e.g. chemical sensors). We do not assume that these agents have actuators that can modify the environment. Thus, the agent communication that we discuss below does not involve any indirect pathways that require environmental alteration (so-called stigmergy).

There are a variety of design approaches for controlling the motion of collections of mobile robots. In this paper we consider a class of control algorithms in which the mobile robot motions are driven by artificial pair-wise artificial "forces", called Aforces, that are functions of distance between the robot pairs. The Aforces can also be functions of the agent state, which in turn can depend on environmental conditions. The Aforces acting on each agent are computed by that agent, and multiple pair-wise Aforces are combined through vector summation. Each robot then generates a motion that corresponds to the resulting artificial force. Combinations of distance-dependent attractive and repulsive Aforces can in general generate complex swarm behaviors.

Each pair-wise interaction requires that the two agents exchange certain information (e.g. location and state). The details of the agent communication protocol have a significant impact on the performance and scalability of the swarm system. Each agent must decide who else to "listen to", what information must be broadcast to other agents and if it is the agent's turn to broadcast at any given time. The use of pair-wise agent interactions means that functional agents must not become isolated from rf contact with the rest of the agent population. The amount of information to be transferred between agents and the subsequent processing requirements for that information should not violate processing and bandwidth limitations of the hardware. Agents that require *direct* information and communication from all other agents create scalability issues as agent population size N increases, since the number of communication links scales as $O(N^2)$. Clearly, large-scale applications motivate communication schemes that require only a constant number of local neighbor agent interactions for each agent, so that the number of communication links scales as $O(N)$. Such approaches can still enable certain key local information to become globally distributed when needed. For example, the detection of areas of interest and areas to avoid by a few agents can be propagated to affect the states and behaviors of the entire population. Such global information, distributed through only local communications, can greatly simplify the design of complex collective behaviors.

In this paper we introduce a graph theoretical approach to the design of swarm communication and mobility architectures that we call cluster networks. Cluster networks are graphs that each agent computes independently in such a way that: the physical positions of the agents are the graph vertices; pairs of agents agree on the presence or absence of a graph edge directly joining them. All state decisions and pair-wise "force" interactions are determined using only information originally present within an agent and information acquired from other agents that are connected by a graph edge in the computed cluster network. Cluster networks require only local communication of an agent with a small, bounded number of neighbor agents. Through an additional construct called proximity indices, cluster networks enable local proximity information about the detection of desirable and undesirable targets to be propagated through the swarm. The cluster networks require exchange of a few bytes of information in each communication cycle, yet provide sufficient local navigation information to enable a variety of complex, collective swarm behaviors. Graph theory proofs have been carried out for certain key properties of the cluster networks. In particular, the cluster network generates a connected graph of all agents that remain within rf range limits, so that all such agents are guaranteed to have pair-wise interactions to guide their behaviors. These graphs exhibit useful properties that are not generally exhibited by simple lists of nearest neighbor agents. We discuss these points below.

We also describe a set of three types of pair-wise "forces" between agents. Two of these mimic the attractive and repulsive potentials well known in physical systems (e.g. Lennard Jones potentials). The third is a novel collision-dependent Aforce inspired by magnetic forces that we propose for general mobile agent control applications. This is a tangential Aforce that acts orthogonally to the pair-wise separation. The Aforce switches between "right-hand-rule" and "left-hand-rule" directions when the agent approaches another agent too closely (i.e., an effective collision occurs). As we illustrate with simulations, this additional Aforce provides a simple approach to achieving a variety of useful swarming behaviors.

There are several goals for this work. Our main goal is to present a set of general-purpose design tools that facilitate the design of complex collective behaviors that satisfy specific performance goals. The intent is that these tools be relatively simple to use and understand, despite the complexity of the behaviors that they enable. Another goal is to produce designed behaviors that can scale up, without modification, to very large numbers of agents in a population. Further, the algorithms so designed should allow implementation in relatively low-power microcontroller processing hardware and using limited-range, low-power communication hardware. Finally, the tools should provide inherent robustness against various errors to be encountered in actual physical robotic embodiments. Our experience with these tools over a several year period suggests that these goals were not unrealistic.

II. Clustering networks

Clustering network graphs are computed using a graph theory construct [5] called a region of influence (ROI). ROIs are two-dimensional (2-D) shapes that are defined with respect to pairs of vertices. In the present case, each agent position is a vertex. The size of the ROI for each pair of vertices is scaled with the separation of the vertices, and the orientation of the ROI is determined by alignment with the line segment that would connect the two vertices. ROIs determine graph edges (of the agents in the present case) through a seemingly simple exclusionary rule -- any pair of agents in a swarm have a connecting graph edge iff no third agent is inside the ROI defined by the pair. The ROI is applied to all pairs of agents in the swarm. Each agent can carry out the ROI calculations itself to determine which other agents have graph edges to itself (called cluster neighbors), and all pairs of (correctly behaving) agents will arrive at the same conclusion about the presence of a joining graph edge between them. Many simple mathematical ROI shapes have been explored in the literature, and the mathematical properties of the resulting graphs have been examined. Here we present a new ROI shape that we have proposed, called the bow tie [6], that has been designed to produce useful graph structures for purposes of swarm agent interactions and communication. We first present the shape, then describe and illustrate the graph properties that provably follow from this shape. For the remainder of the paper we consider only clustering network graphs produced by the bow tie ROI.

The bow tie shape is given in Fig. 1. This ROI is made up of the union of two wedge-shaped regions. The radius of each wedge is equal in length to the line segment that would join the pair of agents (this line segment, not shown in Fig. 1, will be called the pair line for the rest of this paper). Each wedge area is obtained by sweeping each radius in an arc, fixed at one of the pairs of agents, with equal sweep areas on either side of the pair line. We define this ROI to not include the line that forms the boundary of the shape. The bow tie ROI can have a variable total angle of sweep (centered about the pair line) up to 120 degrees. Such bow tie ROIs only sweep an area that can contain additional agents that are no farther from both of the pair agents than the separation of the pair itself. The version used for clustering networks in this work has a sweep angle of +/- 20 degrees about the pair line.

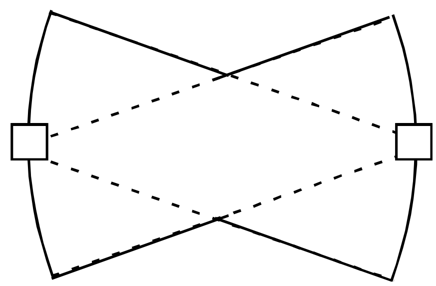

Fig. 1 Bow tie region of influence. The squares indicate the positions of the pair of vectors that are tested for clustering. The dashed lines indicate the wedge-shaped regions discussed in the text.

Figure 2a shows an example set of agents in a plane and Fig. 2b shows the associated cluster network. Several key properties of clustering network graphs are illustrated in this example. First, all of the agents are part of a connected graph. This property has been proven to be true for any number and configuration of agents and any dimensionality of the agent spatial positions [6]. Second, the graph edges for each agent have a minimum relative angle between them. This behavior can also be proved from the shape of the bow tie ROI. The clustering network graph tends to provide local communication that propagates from each agent in a wide range of directions while maintaining overall connectivity.

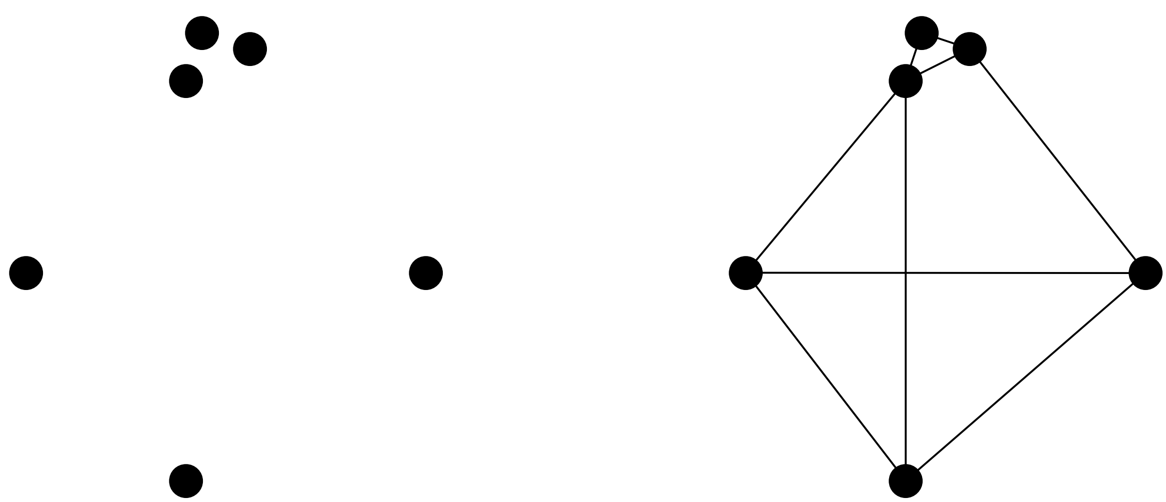

Fig. 2 Schematic of a set of robots in a plane (left) and the corresponding computed graph edges (right). Information transfer and motion control are determined using these graph edges.

These properties produce graphs that can be quite different from those that would result from interacting with the j nearest neighbors of each agent. In particular, nearest neighbor graphs may have all of the j graph edges oriented along a similar set of directions, so that communication flow may be greatly constrained in certain parts of the agent population. Further, each agent produces *directed* graph edges, i.e. these edges in general don't match with those computed by other agents. Thus, an agent may become isolated in that it is not among the closest nearest neighbors of any other agents, and so all other agents ignore it. This can clearly occur even when all agents remain within communication range of some other non-zero set of agents.

III. Proximity indices

The clustering network graph can be used by the agents to provide navigation information to each other using integer values that we call proximity indices. These indices are computed by each agent and broadcast to the other cluster neighbors. If a particular agent acts as the origin (i.e. has a proximity index of unity), then the values computed by the other agents correspond to the minimum number of graph edges that must be traversed from the origin to that agent. Agents can take on the origin value in a variety of useful

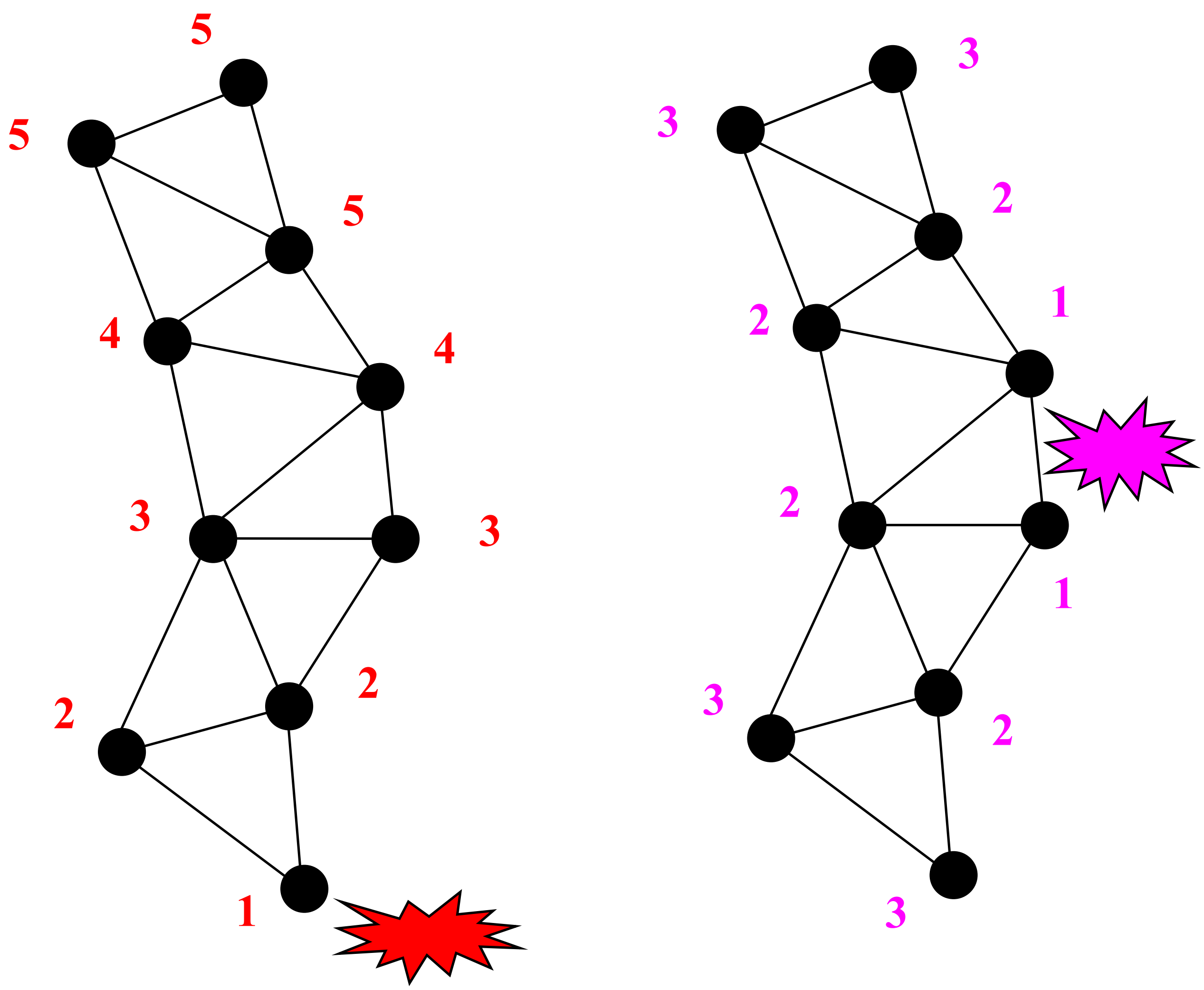

Fig. 3 Two sets of (schematic) proximity grid values with (pmax = 5) for the same graph structure, where the environmental stimuli that trigger the creation of origin graph nodes are schematically indicated by the red and violet regions.

contexts. For example, a sensor may indicate the presence of a material or situation of interest at the agent's location. Each agent calculates a proximity index for itself as follows:

```
if  (an agent is at the origin)
        proximity index = 1
else
        proximity index =  min ( pmin + 1, pmax )
```

where pmin is the smallest proximity index being broadcast by agent's cluster neighbors and pmax is a positive integer value that can be chosen to prevent overflow of the indices. We note that this algorithm is analogous to one used to compute Euclidean distance maps for image processing applications [7]. In the absence of an origin, all agents reach the maximum proximity index value of pmax. Thus, all agents within pmax graph edges of an origin become aware of it, and this information is globally available if pmax is chosen greater than N (the number of agents). The properties of the proximity indices are illustrated for an example set of agents in Fig. 3. The figure shows two example index sets that arise from sensor responses at two different locations.

The figure shows an area that triggers the sensor along with the stable states that the proximity values reach after a few communication cycles. These proximity values clearly provide navigational information for any agent that has a cluster neighbor with proximity index less than pmax. By following the proximity index "gradient" provided by the other agents, an agent can move towards the graph origin and thus to any environmental property that triggers such origins. Similarly, undesirable environmental conditions can be avoided by moving towards agents with larger proximity indices. The useful information provided by the proximity indices can be directly incorporated into the agent behaviors by making the pair-wise agents Aforces dependent on them, as discussed below.

Multiple proximity indices can be maintained simultaneously if agents need information about more then one type of material or situation. These multiple indices can be used in a simple way to provide different types of navigation for agents in different states, or to guide navigation subject to certain constraints (e.g. move toward a certain detected stimulus while also avoiding another type of detected stimulus).

IV. Artificial Forces

The clustering network graph provides key information needed to define a set of artificial forces that each agent computes for itself in order to guide its own motion. In particular, an agent will only compute pair-wise Aforces for cluster neighbors. The vector Aforces that we consider here are functions of the distance between the agent pair, and can also depend on the states and proximity indices of each agent. We use only the vector direction of the sum of Aforces acting on each agent, as the velocity of moving robots is often relatively constant.

Two straightforward Aforces that we use together are an attractive component that dominates at large separations and a repulsive component that dominates at small separations. Such Aforces are intended to mimic the nanoscale intermolecular forces that produce liquid and solid collective behaviors from many individual molecules. These interactions are convenient for achieving the sorts of agent clustering and flocking behaviors that are so familiar to this field.

We also propose a novel Aforce in this work that has proved useful both for providing robustness of the algorithm performance and for facilitating a variety of swarming, surrounding, and searching behaviors. This force is only active within a certain range of cluster neighbor separations. When active, the agent creates a force component that, by itself, would cause the agent to "orbit" the cluster neighbor. The orbit initially chooses either the clockwise or counterclockwise direction randomly. This orbit directions is maintained until the orbit path is occluded by another agent. In this case, the orbit direction is reversed. This force component acts stochastically to find free paths around a cluster neighbor, and produces very robust surround behaviors (e.g., even in the presence of irregularly-shaped obstacles) when activated.

A useful and flexible design tool is the specification of different subsets of these three fundamental Aforces, so that Aforces are "switched" on and off depending on both the states and information broadcast among an agent and its cluster neighbors.

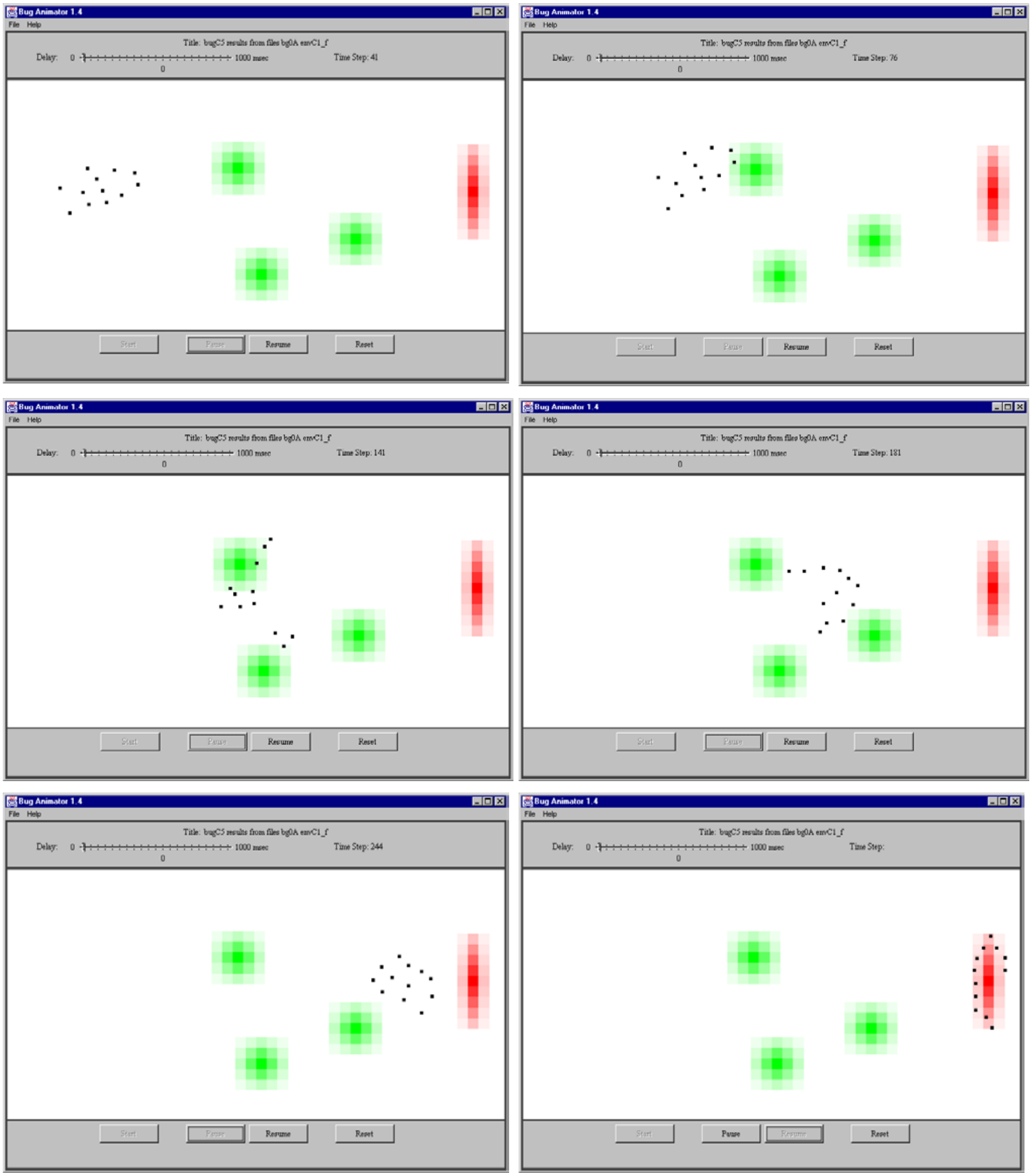

Fig. 4. Six time steps from a computer simulation using the design principles described in the text. The robots (black squares) are drifting towards the right. They are designed to share information in order to avoid the green terrain, detect the red terrain and surround the perimeter of the red terrain.

V. Example collective behaviors

A variety of collective behaviors have been created in simulations. These include:
Surround irregularly shaped region; "Patrol" a perimeter; "Flocking" in a general direction; "Flocking" to a site of high sensor readings; Local navigation guidance inside swarm; Avoiding undesirable terrain; Recruiting more agents to a newly discovered region of interest; Creating complex geometric formations; Detecting certain types of failures of individual robots and compensating for the loss of these agents.

We provide a representative example of a complex collective behavior designed from the principles described above in Fig. 4. The six panels of the figure are sequential frames (left to right, top to bottom) from an animated simulation using the design techniques described above. The mobile agents are the black squares. The green regions are "undesirable" terrain that can be detected only be agents that are physically within these regions. The red area is a "target" area that also can be detected only be agents that are physically within the region. This red region is to be surrounded by the mobile agents. The agents are initialized to drift towards the right and slightly upwards. They proceed towards the right while avoiding undesirable terrain. The swarm reforms after passing around green terrain. When one of the agents first encounters the red target area, all of the agents become "aware" of this discovery using their proximity indices, and all switch to a "surround" set of Aforces which allow them to find and collectively wrap around (including behind) the boundary of the red area. The robots are designed to halt after surrounding such an area.

Our intent is that the tools should provide inherent robustness against various errors to be encountered in actual physical robotic embodiments. We have only limited experience driving small sets of physical robots with these techniques. We have more extensively studied the robustness of a variety of simulations against errors and noise that are introduced into the coordinate readings, the environmental sensor readings and the movement directions of the robots. We have also considered the failure or malfunctioning of a few robots within the swarm. The detailed results are not presented here, but we generally observe considerable robustness for modest error and noise levels for simulated swarms up to 400 robots.

VI. Conclusions

We have presented a set of novel design principles to aid in the development of complex collective behaviors in fleets of mobile robots. The key elements are: the use of a graph algorithm that we have created, with certain proven properties, that guarantee scalable local communications for fleets of arbitrary size; the use of artificial forces to simplify the design of motion control; the use of certain proximity values in the graph algorithm to simplify the sharing of robust navigation and sensor information among the robots. The intent is that these tools be relatively simple to use and understand, despite the complexity of the behaviors that they enable, and that the designed behaviors can scale up, without modification, to very large numbers of agents in a population. Further, the algorithms so designed are intended to allow implementation in relatively low-power microcontroller processing hardware and using limited-range, low-power communication hardware. Finally, the tools are intended to provide inherent robustness against various errors to be encountered in actual physical robotic embodiments. Our design elements have met these goals during our multiyear experience using them to design complex robotic behaviors.

Acknowledgements

This work was sponsored by the U.S. Department of Energy under Contract DE-AC04-94AL85000. Sandia is a multiprogram laboratory operated by Sandia Corp., a Lockheed Martin Co., for the U.S. Department of Energy.

References

[1] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions," *Proceedings of the 1995 IEEE/RSJ IROS Conference,* 226-234, 1995.

[2] L.E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics," *Distributed Autonomous Robotic Systems 4*, ed. L.E. Parker, G. Bekey, J. Barhen, Springer-Verlag, pp. 3-12, 2000

[3] S. Goldsmith, J. Feddema, R. Robinett, "Analysis of Decentralized Variable Structure Control for Collective Search by Mobile Robots," SPIE98, *Proc. Sensor Fusion and Decentralized Control in Robotic Systems*, Boston, November 1-6, 1998.

[4] J.T. Feddema, D.A. Schoenwald, F.J. Oppel, "Decentralized Control of a Collective of Autonomous Robotic Vehicles," *Proceedings of the American Control Conference*, Arlington, VA, June 25-27, 2001.

[5] G.T. Toussaint, "The Relative Neighborhood Graph of a Finite Planar Set," Pattern Recognition 12, pp. 261-268, 1980.

[6] G. C. Osbourn, "Some Provable Properties of VERI Clustering," SAND2000-1766, 2000.

[7] J. C. Russ, *"The Image Processing Handbook,"* CRC Press, Boca Raton, Florida (1992).