International Journal of Computer Science & Applications
Vol. 4 Issue 3, pp 84-94

Purposeful Deployment via Self-Organizing Flocking Coalition in Sensor Networks

Yee Ming Chen [1] **Shu-Hao Chang**[2]
chenyeeming@saturn.yzu.edu.tw eric.syam@msa.hinet.net

1 Department of Industrial Engineering and Management, Yuan Ze University,
135 Yuan-Tung Rd, Chung-Li, Tao-Yuan, Taiwan, Republic of China
2 Air Force Institute of Technology,
198 Jieshou W.Rd., Gangshan, Kaohsiung, Taiwan, Republic of China

Abstract

This paper presents a deployment strategy via self organizing flocking coalitions for mobile sensor network coverage. The concepts of our approach are inspired by the flocking phenomenon which locally controls a group of mobile agents performed as follows: (i) global alignment of their velocity vectors, (ii) convergence of their velocities to a common one, and (iii) collision avoidance. Adding mobility of local control to sensor network significantly increase the capability of the sensor network by making it resilient to failures and responding to events. The controllable mobility in sensor network hence is used to help achieve the network's missions. That is, mobility may be "purposeful" instead of being treated as an uncontrollable external stimulus to which the ad-hoc networks must respond. To make use of the purposeful deployment, we implemented and evaluated a fire tracking application to determine how well sensor network deployment achieves its goal. Fire is modeled by agents that gradually spread throughout the specific environment. We then introduce a simulating model for the movement of autonomous fire tracker agents that exhibits flocking behavior and forms a mobile sensor network collectively. Fire tracker agents are used to form a perimeter around the fire agents to extinguish fire agents. Finally, we demonstrate that a group of fire tracker agents executing the leaderless flocking models with emulated sensor network can successfully flock (even in the presence of individual fire tracker agents failure) and that systematic characterization of sensor network flocking parameters is achievable.

Keywords: Sensor Network, Deployment, Flocking, Agent.

1 Introduction

Sensor network has recently received significant attention in the areas of networking, embedded and multi-agent systems [13] due to its wide range of application domains such as the following : sensors are rapidly deployed in a remote inhospitable area for a surveillance application; sensors are used to analyze the motion of a tornado; sensors are deployed in a forest for fire detection; sensors are attached to taxi cabs in a large metropolitan area to study the traffic conditions and plan routes effectively. In these applications, the distributed sensing task is achieved by the collaboration of a large number of sensors, each of which has limited sensing, computational and communication capabilities. Recent technological advances are likely to increase sensor network development and deployment efforts, because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment. One of the fundamental issues that arise in a sensor network is deployment. Traditionally, network deployment is maximized by determining the optimal placement of static sensors in a centralized manner, which can be related to the class of art gallery problems [7]. However, recent investigations in sensor network mobility reveal that mobile sensors can self-organize to provide better coverage than static placement [11]. Therefore, the expected growth in sensor network mobile deployment requires efficient sensor network modeling techniques to facilitate initial programming of the sensor nodes and their eventual reprogramming once the network is deployed. Our sensor networks deployment problem is motivated by the flocking modeling. Flocking is a group behavior of large number of agents with a common objective. In English it is described as "moving together in large numbers". Reynolds[9] introduced three flocking models. These models describe the behavior of each agent in interaction with other neighboring agents (or flockmates). The problem is that expressions like “attempt to stay close” and “attempt to match velocity” have broad meanings and are open to various applications. Applying flocking models in sensor networks deployment have led to: a) no prior information about the exact target locations, population densities or motion pattern, b) limited sensory range, and c) very large area to be observed.

In this paper, we developed an agent-based simulation environment to facilitate the mobile agents' deployment and enable varied testing of coverage and deployment issues in a wide variety of sensor networks. The simulation environment allows users to deploy applications by injecting mobile agents into a sensor network that coordinate and migrate in flock to perform application-specific tasks. Mobile agents can move or clone themselves to desired locations in response to changing conditions in the environment. In order to aid the development and simulation of flocking behavior for mobile agents, we have created a simulation of fire tracker agents swarm searching for situational awareness, target identification, and to extinguish fire within a patrolled region. Fire is modeled by mobile agents that gradually or epidemically spread throughout the network. Fire tracker agents are then deployed to autonomously form and maintain a perimeter around a fire. We have implemented and evaluated a fire tracking application to determine how well the fire tracker agents achieve their extinguishing goals. This paper makes three primary contributions. First, it demonstrates how the mobile agents can be used to facilitate the development and deployment of a sensor network. Second, we employ flocking behavior that demonstrates the reliability and efficiency of mobile agents in a highly dynamic application. Finally, in order to look deeper for applying flocking models in sensor networks deployments. The animation of fire tracking application is created.

The remainder of this paper is organized as follows. Section 2 provides an overview of sensor networks deployment and flocking models. Section 3 describes the development of simulation environment for sensor network deployment. The self-organizing flocking coalition simulation results and fire tracking applications on analysis of the mobile sensor network coverage are stated in Section 4, respectively. Finally, conclusions are made in Section 5.

2 Flocking Models for Sensor Networks

In future smart environments, wireless sensor networks will play a key role in sensing, collecting, and disseminating information about environmental phenomena. Sensor networks hold the promise of revolutionizing sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost-effectiveness, and ease of deployment. In this section, we first give a brief review of the related work on sensor network deployment, and then present the challenges and the deployment solution on mobile sensors.

2.1 Sensor Network Deployment

Traditional sensors are described as being larger than sensors in a sensor network, and requiring careful, manual positioning and communications topology engineering. The distinction between sensor networks and traditional sensors has more to do with the technical limitations than choice of design factors, including fault tolerance, scalability, production costs, operating environment, sensor networks deployment, hardware constraints, transmission media, and power consumption[4]. While these factors are important considerations in this work, only sensor networks deployment are addressed in here. The sensor networks deployment and their approaches must account for sensor failure and rapidly changing communication topologies, and they must be based on minimizing power consumption. Minimizing power consumption means minimizing communications; for mobile sensors, it also means minimizing movement. There are several approaches to construct and maintain a sensor network forms a topology between observer and phenomenon. These will differ depending on the network dynamics, which we classify as: static sensor networks and mobile sensor networks. In static sensor networks, there is no motion among communicating sensors, the observer and the phenomenon. An example is a group of sensors spread for temperature sensing. For these types of static sensor networks, previous studies have shown that localized algorithms can be used in an effective way [3], [5]. In mobile sensor networks, either the sensors themselves, or the phenomenon are mobile. Mobile sensor networks can be further classified by considering the motion of the components. This motion is important from the communications perspective since the degree and type of communication is dependent on network dynamics. This dynamic requirement on sensor locations cannot be easily met by deploying a large number of sensors, since provisioning for all possible combinations of mission requirements may not be economically feasible. More importantly, precise sensor deployment may not be possible, especially in a hostile environment, where sensors are subject to power depletion, failures, malicious attacks, and may change their physical locations due to external force. Therefore, the mobility of sensors may significantly increase the capability of the sensor network by making it resilient to failures, reactive to events, and able to support disparate missions with a common set of sensors. Since mobility assisted sensing is still a new area, there is not much work in the literature. The closet work is sensor deployment. Previous work on sensor placement [10], [11], [12] largely addressed random and sequential deployment. In a pure random scheme, many more sensors are required than the optimal number to achieve high coverage of a target area. There have been some research efforts on deploying mobile sensors, but most of them are based on centralized approaches. The work in [13] assumes that a powerful cluster head is able to know the current location and determine the target location of the mobile sensors. However, such a central server may not be available in most cases, and this approach suffers from single point failure problem. The motion capability of sensor nodes can also be used for purposes other than sensor randomly deployment. For example, in case of a sensor failure or malfunction, other sensors can move to replace the role of the failed sensor. As an event (i.e., fire, chemical spill, and incoming target) occurs, more sensors should relocate to the area of the event to achieve a better coverage. Compared with sensor randomly deployment, mobile sensors may be "purposeful" which relocates mobile sensors from one place to another place, has many challenges. Although mobile sensors relocation can be used to improve the sensing coverage, their costs may be high. To achieve a good balance between sensor cost and sensor coverage, we proposed the flocking task to assist the deployment of mobile sensor networks. In this research, we envision a population of sensors. The generic entity is "sensor," which may be termed a "mobile agent" to emphasize its autonomous computational abilities rather than its specific sensory function.

2.2 The Flocking Models

The flocking models examined in this paper are similar in form to the cooperative movement task studied in Reynolds [9]. Each mobile agent (figure 1) is equipped with a limited distance of sensing range that allows for the detection of objects through a range of angle centered on the front of the agent.

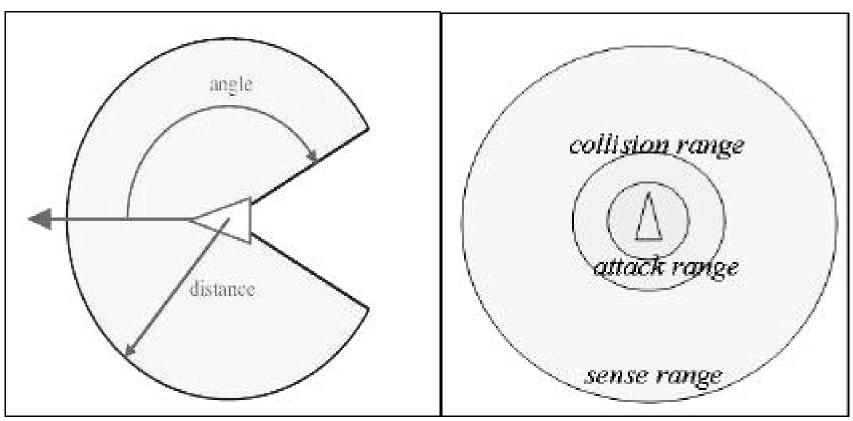


Fig.1. Mobile agent

Each mobile agent has direct access to the whole scene's geometric description, but flocking requires that it reacts only to flockmates within a certain small neighborhood around itself. The neighborhood is characterized by a distance and an angle, measured from the agent's direction of flight. Flockmates outside this local neighborhood are ignored. The neighborhood could be considered a model of limited perception, or just the region in which flockmates influence a mobile agent's steering. The basic three flocking models consist of simple steering behaviors which describe how an individual mobile agent maneuvers based on the positions and velocities its nearby flockmates:

As shown in Fig.1, a triangle representing the mobile agent in the core of three concentric circles could be clearly seen. These three specific ranges according to the center to perimeter order are attack range, collision range and sense range respectively. The three ranges above could be simply interpreted in relation to the way we name them. For instance, once upon agents entering the "attack" ranges of others, they attack each other; entering the "collision" ranges, they collide with each other; entering the "sense" ranges, they sense each other.

We Assuming:
Sense Range Radius = Rs,
Collision Range Radius = Rc,
Attack Range Radius = Ra,
Distance to a Flockmate = D;

(a) Separation model: steer to avoid crowding local flockmates (Fig. 2);

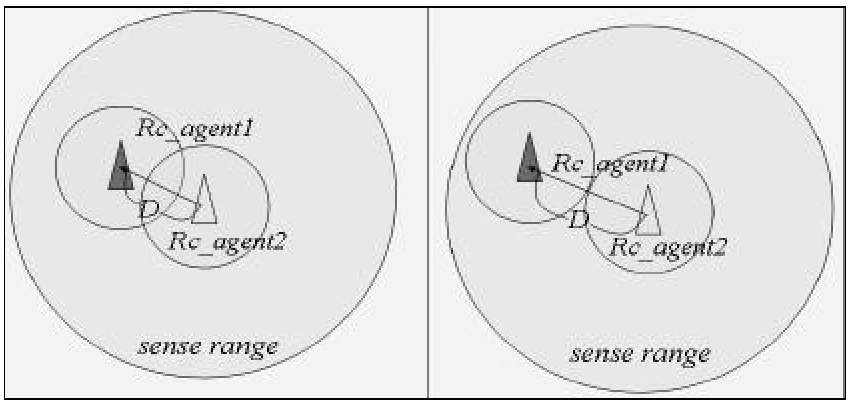


Fig.2. Separation model

The algorithm of separation model is shown as the following:
If
D< (Rc_agent1+ Rc_agent2),
too crowd;
then
Move away from flockmate
Until
D= Rc_agent1+ Rc_agent2.

(b) Alignment model: steer towards the average heading of local flockmates (Fig. 3).

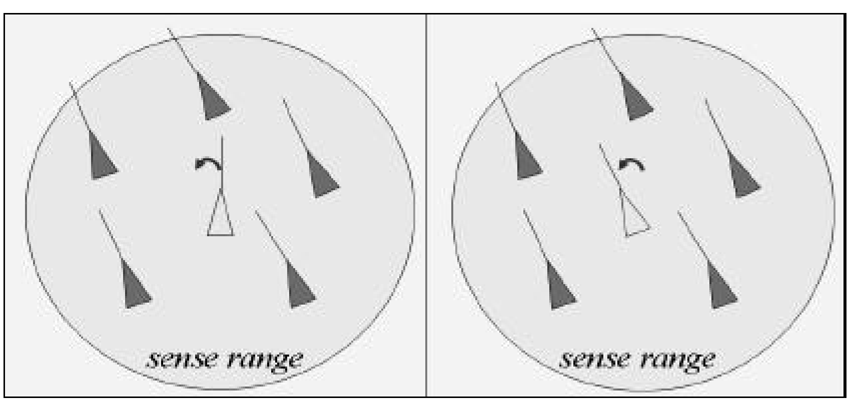


Fig.3. Alignment model

The algorithm of alignment model is shown as the following:

Assuming:
Sense Range Radius = Rs,
Sense
Average heading angle and velocity of flockmates nearby (within Sense range);
Then
Align itself with (that is, head in
the same direction and speed as) other nearby flockmates.

(c) Cohesion model: steer to move toward the average position of local flockmates (Fig. 4).

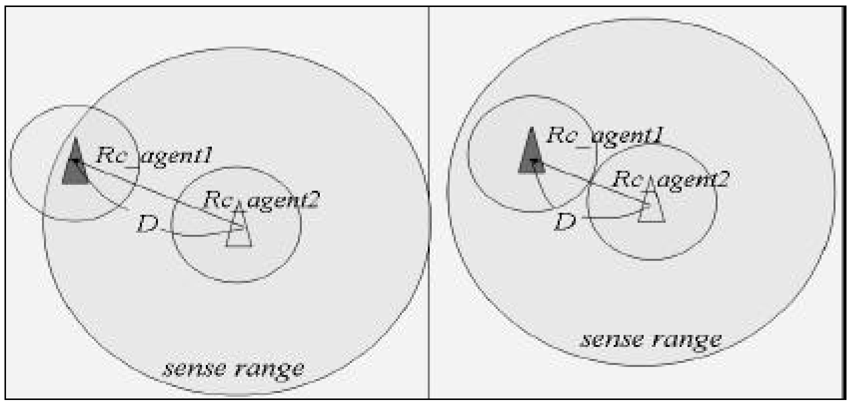


Fig.4. Cohesion model

The algorithm of cohesion model is shown as the following:

if
D> (Rc_agent1+ Rc_agent2),
too far;
then
Move closer to flockmate
Until
D= Rc_agent1+ Rc_agent2.

The superposition of these three flocking models results in all mobile agents moving as a flock while avoiding collisions. Motivation for this work comes primarily from flocking task as a mechanism for achieving *velocity synchronization* and *regulation of relative distances* between agents in the sensor network.

3 Development Simulation Environment for Sensor Network Deployment

In order to aid the development and simulation of flocking behavior for mobile sensor network deployment, the simulation environment comprised of two layers: user interface and mobile agents simulator. Figure 5 is a screenshot of the agent-based simulation environment. The agent-based simulation environment is divided into three areas. The top area is a 3D view showing the fire tracker agents performed their purposeful mobility (separately shown in Fig.9). The left area includes the statistics chart of quantitative comparison between two antagonistic agents (i.e. fire tracker agents and fire agents) and sensors' range display. The right area is the user interface layer which consists of two programs: Configuration Interface Program and Simulation Viewer. Configuration Interface Program provides the slide bars to directly manipulate mobile agents' sense and collision range radius for the simulation. Simulation Viewer is a tool to check and verify the simulation results.

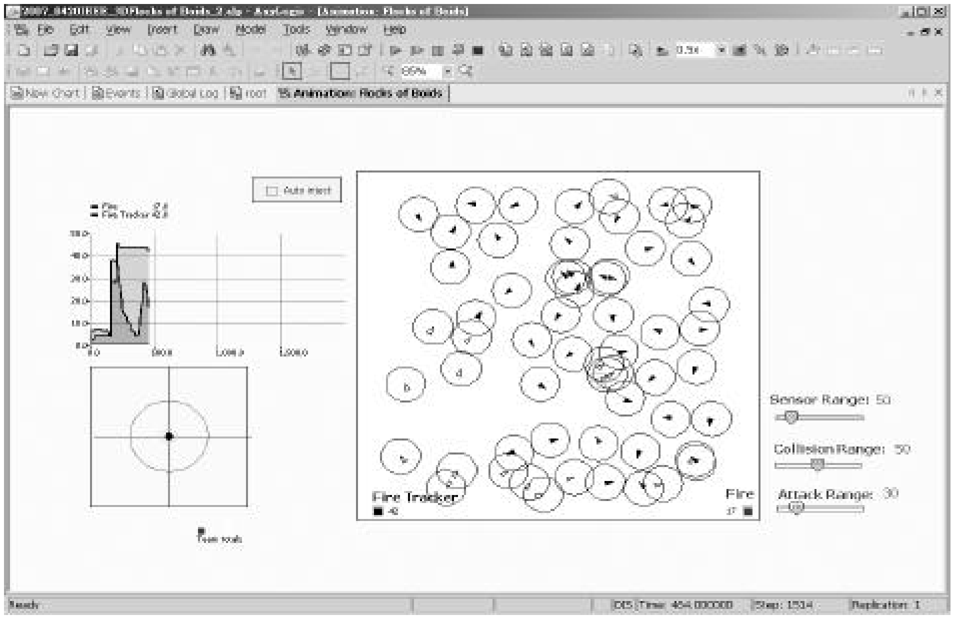


Fig.5. Snapshot of agent-based simulation environment interface

All task-oriented components, such as sensors and targets are implemented as mobile agents in the simulation environment. Each agent has its own thread to progress its state. The thread execution and interaction of agents are controlled by the agent-based simulation environment. Mobile agents' simulator can generate two categories: fire tracker, and fire agents. Fire tracker agents can be dynamically formatted the sensor networks to perform their mission autonomously. With a bottom-up orientation, three flocking models behavior abilities were specified. These abilities can be combined to form composite abilities. In our simulation environment, fire tracker agents are provided with the flocking models. Figure 6 shows a state transition diagram for the fire tracker move follow the flocking behavior. However, if fire trackers were destroyed by fire agents or themselves energy exhausted, the fire tracker agent enters the eliminated or exhausted state. We employ internal state timers to handle this situation. Alternatively, all triggered behaviors could be allowed to fire simultaneously, which is sufficient if there is adequate separation of signals and behavior in an agent's state space [2, 6].

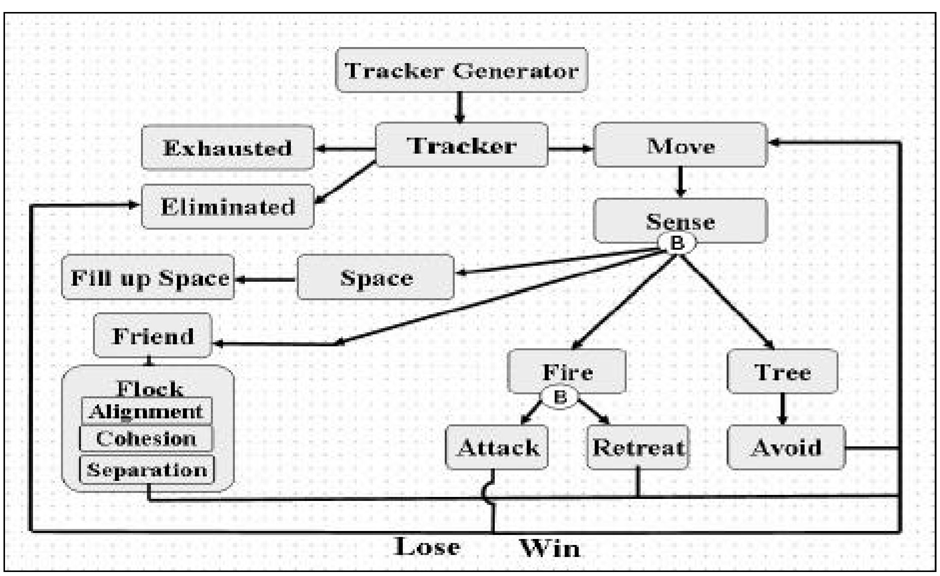


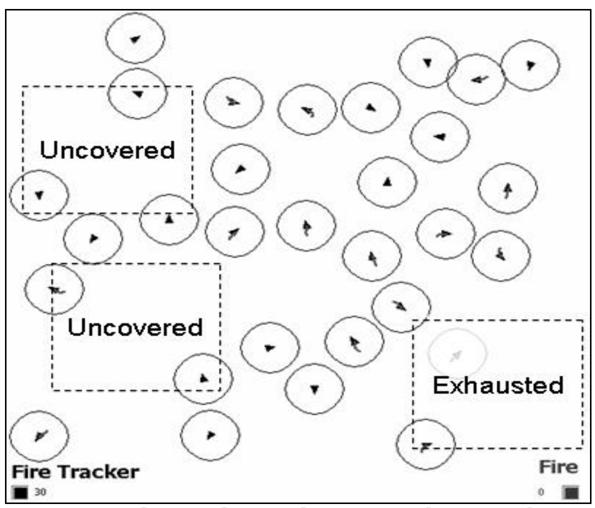
Fig.6. State transition diagram for fire tracker's behavioral modeling

4 Mobile Sensor Network Deployment And Fire Tracking Applications

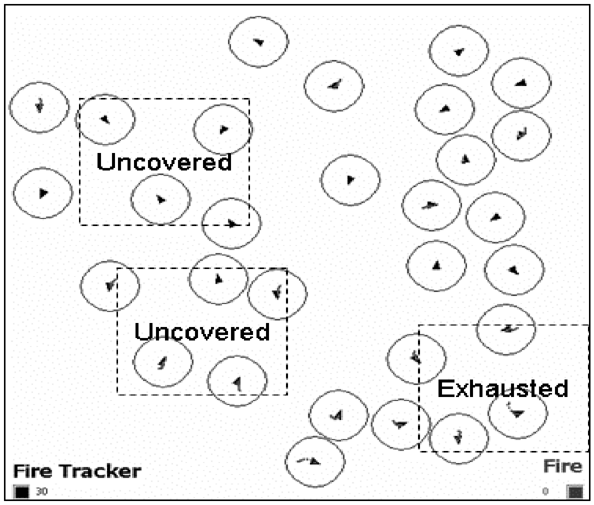
The simulations launch further research in sensor network with self-organizing flocking coalition to deployment. It is possible to evaluate configurations with an arbitrary number of autonomous agents in environments with varying characteristics and behaviors. The agent-based simulation environment proposed in this study gave rise to the real world by different means, depending upon the fundamental requirement of deploying a group of sensors in the hostile area to detect, barrier or eliminate the targets. For example, Sandia foams fight SARS virus [8] research by Sandia Technology [14], and a case study of forest fire emergent system [1]. Therefore, the fire tracking application consisting of two antagonistic agents with particular features is revealed. Followings are to demonstrate the scenarios of fire-fighting.

4.1 Dynamic Team Formation: Automated Rendezvous

Except for not hitting something unconcerned, the more likely emphasis should be put in the dynamic coordination after fire trackers successfully reached the patrol area. Specifically, fire trackers might form an imperfect coverage with certain amount of exhausted or eliminated agents which led to ineffective performance. With respect to the trend of sensors to fill up space uncovered, all of which can be seen in Fig. 7(a) which highlights three dotted line rectangles, one of which uncovered because of the failure of a fire tracker while the others are temporarily uncovered in the dynamic sensor coordination teaming process. Subsequently, as observed in Fig. 7(b), which shows the consecutive snapshots of 2-D flocking for a group of fire trackers. Based on this simulation, flocking can be used as a means of automated rendezvous (or gathering) for a medium to large number of autonomous fire trackers. It is apparent that after some finite time, the fire trackers fill up the uncovered area and maintain its connectivity in their patrol area and form the functional team cooperatively to accomplish assigned missions during their lifetime.



(a) Fire trackers exhausted/uncovered in patrol area



(b) Sensor agents fill up uncovered patrol area

Fig.7. Consecutive snapshots of fire trackers (black triangles) fill up patrol area

4.2 Fire Tracking and Extinguish

One emergent pattern that has been simulated concerns synchronized simultaneous fire extinguishing. The fire trackers (black triangles) identified fire agents (hollow triangles) and perform tracking and extinguishing. Similarly, sensor networks are gifted with sensing the existence of fire agents. Therefore, they go around the emergent mobile fire agents to terminate. This fire tracking and extinguish problem can be solved with flocking behavior. The tracking behavior performed by the fire tracker agents (see Fig. 8) is to chase fire agents detected within their sense range. Whenever a fire tracker catches up with a fire agent with the same velocity and approaches of the attack range mentioned in Fig.1, the fire tracker attacking the fire agent and extinguishing it, and vice versa.

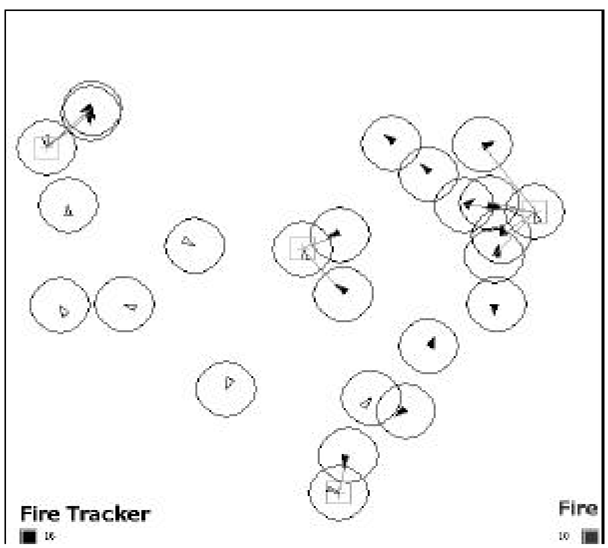


Fig.8. Fire tracker agents tracking mobile fire agents

The fire tracking procedure can be simultaneously displayed in 3-D animations showing the lower boundary of the forest being searched (fire agents with black triangle) and the fire tracker agents (white triangle). The fire tracker agents performed their purposeful mobility in the forest which zoomed as in Fig. 9.

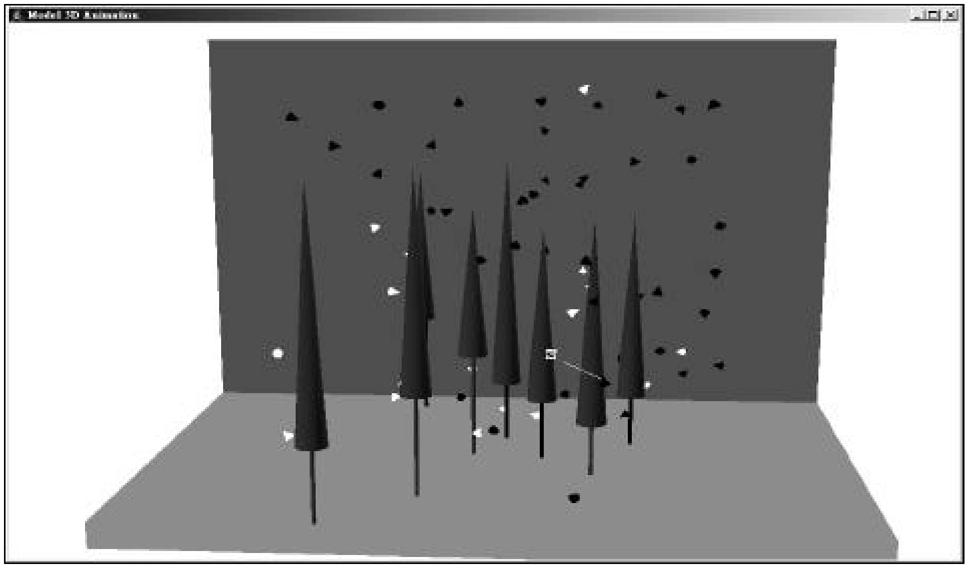
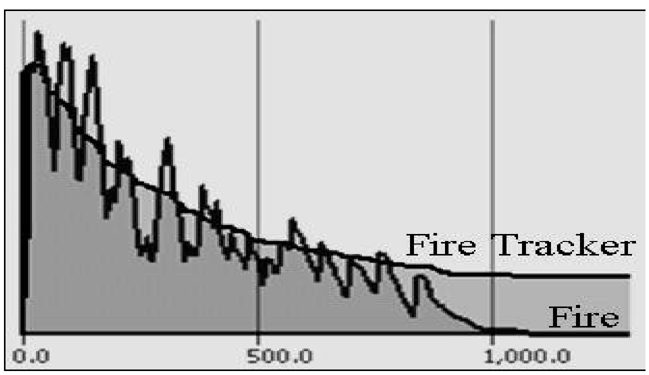
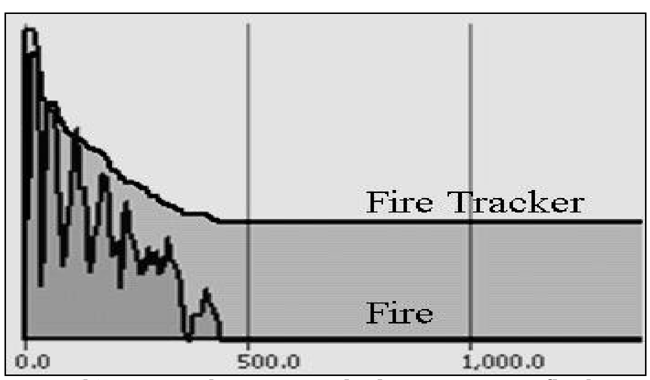


Fig.9. Zoom snapshots of fire tracker agents perform purposeful mobility in the forest

Quantitative comparisons with other deployment strategies such as random deployment, the deployment using flocking model show that this approach can provide better coverage and quickly to respond to environmental changes (see Fig 10).



(a) Fire tracker agents using random deployment



(b) Fire tracker agents' deployment using flocking models

Fig.10. The quantity comparison between two antagonistic agents using different deployment strategies

(The x-axis represents time progress in second and the y-axis represents the numbers of agents)

5 Conclusion

In this paper, we have developed an agent-based simulation environment. The two layered architecture for our sensor networks deployment simulation is presented: user interface and mobile agents' simulator. We have described flocking behavior to perform coalition in the sensor networks. To make use of the purposeful deployment, we implemented and evaluated a fire tracking application to determine how well sensor network deployment achieves their extinguish goal. We introduce a model for simulating the movement of autonomous fire tracker agents that exhibit flocking behavior and collectively form a mobile sensor network. Fire is gradually spread throughout the specific environment. Fire tracker agents are used to form a perimeter around the fire agents to extinguish fire agents. Finally, we demonstrate that a group of fire tracker agents executing the leaderless flocking models with emulated sensor network can successfully flock and that systematic characterization of sensor network flocking parameters is achievable. From quantitative comparisons with other tracking strategies such as random deployment, our approach can provide better coverage and quickly to respond to environmental changes. This approach has been supplemented by our simulation results and is useful to develop and verify the deployment strategies of sensor networks through software simulation prior to real test.

References

[1] Bonabeau, E., Theraulaz, G. "Swarm Smarts." Scientific American, pp. 72-79, March 2000.

[2] C.-L. Fok, G.-C. Roman, and C. Lu., "Mobile Agent Middleware for Sensor Networks: An Application Case Study," International Conference on Information Processing in Sensor Networks (IPSN'05), Los Angeles, CA, April 2005.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc.4th ACM International Conference on Mobile Computing and Networking (Mobicom'98)*, Aug. 2000.

[4] Internet Engineering Task Force MANET Working Group, "Mobile ad hoc networks (MANET) charter," http://www.ietf.org/html.charters/manetcharter.html.

[5] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming.," in *Proceedings of the Eighteenth ACM Symp. on Operating Systems Principles [21]*, pp. 146–159, Oct. 2001.

[6] Lesser, V.; Ortiz Jr., C. L.; and Tambe, M. 2003. Distributed Sensor Networks: A Multiagent Perspective. Kluwer Academic Publishers.

[7] O'Rourke, J. 1987. Art Gallery Theorems and Algorithms. Oxford Univ. Press.

[8] R. Oberst., "Sandia Foams Fight SARS Virus," *Sandia Technology Journal*, 6(1):14-15. 2004.

[9] Reynolds, C.W., "Flocks, Herds, and Schools: A Distributed Behavioral Model," Computer Graphics, 21(4): pp. 25-34, 1987.

[10] S. Meguerdichian, F. Koushanfar, G. Qu and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks," *Mobicom*, 2001.

[11] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Network," *IEEE INFOCOM'01*, April 2001.

[12] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K. k. Saluja, "Sensor Deployment Strategy for Target Detection," *First ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[13] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," INFOCOM, 2003.

[14] http://www.eurekalert.org/features/doe/2004-06/dnl-sfs061404.php