A Fully Decentralized Approach for Incremental Perception

I.Mir and B.P.Amavasai, *Member, IEEE*

***Abstract*—The area of Swarm Robotics is still in its infancy. Key concepts at the basic level have to be invented and developed in order to achieve the future goal of building large scale physical and controllable autonomous robotic swarms. In this paper we extend the concept of Incremental Perception in swarm robotics into the domain of complete decentralization. Our work is aimed at micro-robotic swarms where the hardware resources available for the robots will be limited. Hence a decentralized system becomes inevitable because it does not require intra-dependence of robot agents, their monitoring system or a communication mechanism for the agents; absence of all these factors results in reduced hardware requirements for the agents. We focus on the co-operative behavior of robots rather than relying on their individual capabilities. We also propose the parameters and functions that are required for a completely decentralized system and show that such a system can be successfully modeled and analyzed.**

***Index Terms*—Artificial Intelligence, Mobile Robots, Intelligent Robots, Mobile Robot Motion Planning.**

I. Introduction

THIS paper builds upon the research and findings of a previous paper [1] in which we developed and introduced the term and concept of *Incremental Perception*; it was defined as the ability of individual members of a swarm to perceive part of a complex problem, and use these pieces of information to reach a goal which is unachievable by an individual agent. We postulate that the idea would be of key importance in the development of large scale physical robotic swarms.

Our previous model was based upon a hybrid system while the current work takes the concept another step forward by presenting a completely decentralized system for heterogeneous robots. We have modeled an autonomous swarm which is able to make decentralized decisions and demonstrate stigmergy (indirect communication of agents through modification of their local environment). As a proof of the above claim, we assign the swarm a task of ring formation around an object and extraction of its 2-D shape.

The decentralized heterogeneous model that we present here is scalable and new behaviors and robot breeds can be added to it. At present the swarm has two types of robot models which differ in their architecture and behaviors. These autonomous agents can move around in a controlled world and pursue the predefined complex tasks of ring formation and shape extraction which are beyond the capability of an individual agent. We show that our model is scalable, robust, exhibits behavior based cooperation in the absence of any explicit communication and although the individual agents have limited capabilities, the swarm as a whole is able to perform increasingly complex tasks. Like a biological swarm (e.g. in ants) the success of these agents lies in their co-operative behavior and not the intelligence of individual agents.

The architecture introduced in [1] presents a hybrid model in which the swarm initially behaved in a decentralized way. This was followed by the initialization of a central controller for complex decision making which converted the architecture into centralized; this required the overhead of introducing a communication mechanism for robots. These mechanisms make the swarm dependent upon beacon agents which are in effect master nodes or decision makers [8], hence considerably reducing the important factors towards achieving true swarm architectures [3].

It has been suggested in [4] that communication or the lack of it is a key design consideration which would eventually influence the complexity of a system. In the present work, our focus has been on micro-robotic swarms; hence system complexity is highly undesirable. [5] discusses the limitations of available resources in terms of batteries, sensor systems and communication mechanisms available for micro-robotic systems. We have therefore avoided any factors that might result in increased system complexity; the intelligence in our model is because of algorithms and the behaviors which emerge as a result of agents following these algorithms. Although the absence of explicit communication makes the development of behaviors very difficult, it adds to the robustness, flexibility and scalability of the system. These are the factors classified as the motivators behind a true swarm approach in [6].

[9] and [10] mention three key factors for formation i.e. avoidance, aggregation and dispersion. We use a decentralized approach to these behaviors and show that combining them with a simple set of principals for robots results in complex swarm behaviors.

Manuscript received June 10, 2007. This work was supported by the European Union Framework 6 Project, I-SWARM (project no. 507006).

I.Mir and B.P.Amavasai are with the Microsystems and Machine Vision Laboratory at Sheffield Hallam University, UK. (phone: +44 114 2254520; fax: +44 114 2253433; e-mail: mir_immad@yahoo.com, b.p.amavasai@shu.ac.uk; www : http://www.shu.ac.uk/mmvl).

ROBOCOMM 2007, 14th – 16th Oct 2007, Athens, Greece.

ISBN 978-963-9799-08-0
DOI 10.4108/ICST.ROBOCOMM2007.2216

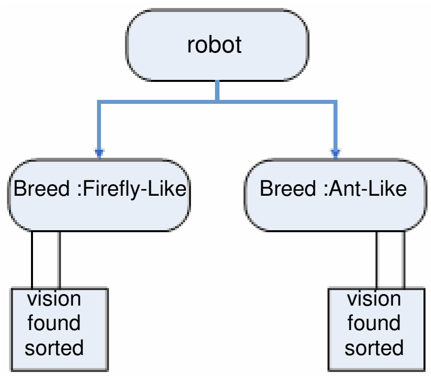

Fig. 1. Robots exist as two discrete breeds which have the same basic structure but completely different set of behaviors associated with them.

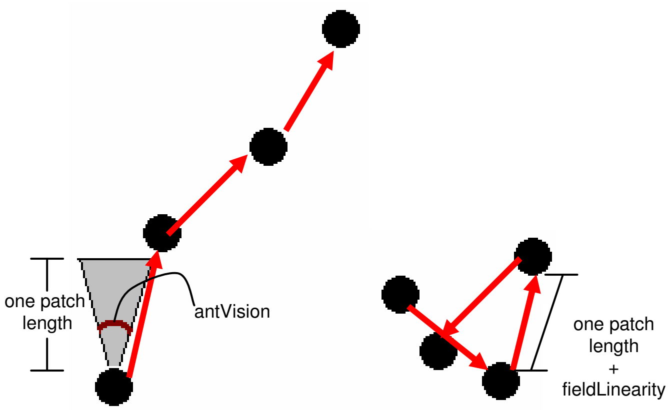

Fig. 2. An Ant-Like robot (left) has a limited field of vision determined by `antVision`. It can step forward in an area bound by the angle `antVision` and unit length of a patch. Firefly-Like robots (right) move about randomly and can take a step forward in any direction. Both types of robots have only one touch sensor and can only sense an object when the sensor touches the object.

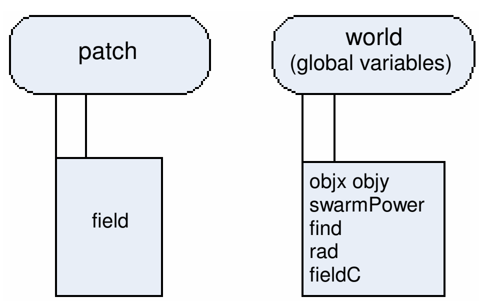

Fig. 3. Patches have only one variable each, the field. The world holds a set of variables that are accessible both by patches and robots.

II. The Design

A. NetLogo and Design Parameters

We have used the NetLogo modeling tool which is a *cross platform multi-agent programmable modeling environment* [7]. NetLogo is suitable for simulating swarm-robotic systems because of its capabilities of handling large numbers of agents (thousands). It has the ability to define the rules for agent interaction in an efficient way and allows these agents to be simulated in a concurrent environment.

The system consists of three major players *i.e.* robots, patches and the world. The term robot is self explanatory; we have designed two breeds of robots, the Ant-Like and Firefly-Like (Fig. 1) which have different architectures. The first model for Incremental Perception in [1] is rather primitive. Although it used NetLogo breeds, it did not utilize the full potential of the facilities associated with it, this feature has however been exploited in our present work and has resulted in improvement of the basic algorithms and behaviors, this has also introduced better facilities to conduct experiments in which both types of robots can co-exist.

Associated with robot breeds are the variables `vision`, `found` and `sorted`. Ant-Like robots check for the presence of an object at `heading + (antVision/2)` and `heading – (antVision/2)`, and then move forward at one of these headings. Firefly-Like robots only check for an object in their current direction, and move forward in a random direction. Both breeds of robots have only one touch sensor and the heading of a robot is also the direction of the touch sensor.

The variables `found` and `sorted` are the states of robots. A robot which has found an object will have the variable `found` set to `true`. A robot is `sorted` only when it has arranged itself around the object and is currently looking at it. The robots move around in an environment which is composed of `patches`, a NetLogo primitive that is a small division of the environment and allows defining rules under which robots and world may interact with each other. We have associated a variable `field` with each patch which is the strength of a *potential field*. The concept is similar to the real world in which a large number of cues *e.g.* temperature, humidity, strength of microwaves at different frequencies, are present and have different values at different spatial locations and time. Each robot has the ability to create a potential field (that attracts other robots) when it comes in contact with an object. The field strength in our model can have different values depending upon the number, state and motion of robots that generate it. Field strength at the point of origination is a constant represented by `fieldC`. Strength of field at any point is given by

$$\Phi = C - r \tag{1}$$

where Φ is the field strength, C is the constant `fieldC` and r is the Euclidian distance between the point where the field is being determined to the point of origination of field. The constant `fieldC` corresponds to the gravitational potential energy possessed by an object at the surface of earth, and increases with the distance from the point of origination. This potential field is shown in Fig. 5 as red colored patches whose intensity is low at the center, the point of origination, and increases gradually with the distance.

The world is a set of global variables with which robots and patches may interact (Fig. 3). In fact it is the world that enables the simulation of behavior based coordination. In Fig. 4, note the directional arrows showing behavior based interaction between components. The robots and patches may interact with each other by changing their variables

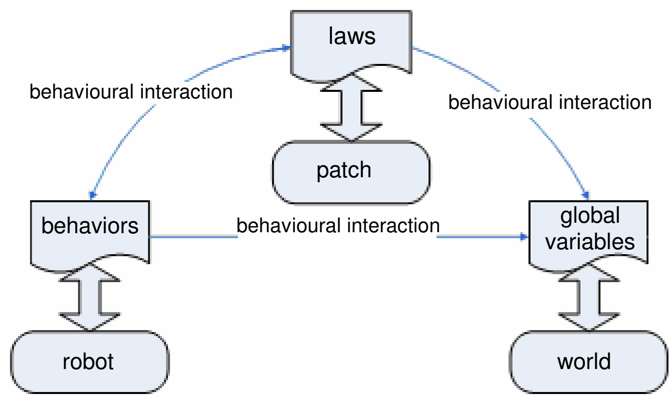

Fig. 4. Robots and patches may interact by changing each other's variables, but world cannot do so. It only preserves states.

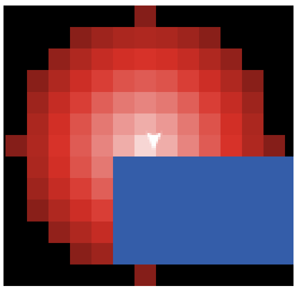

Fig. 5. Potential Field as a factor of Euclidian distance from the center of field. The blue patches constitute an object.

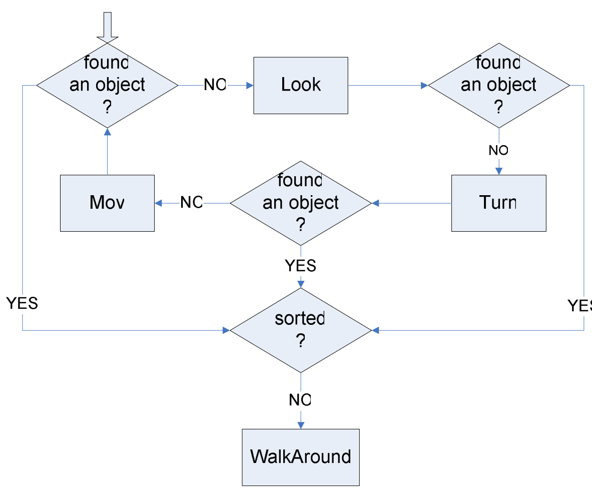

Fig. 6. The basic algorithm that controls the behaviors of robots.

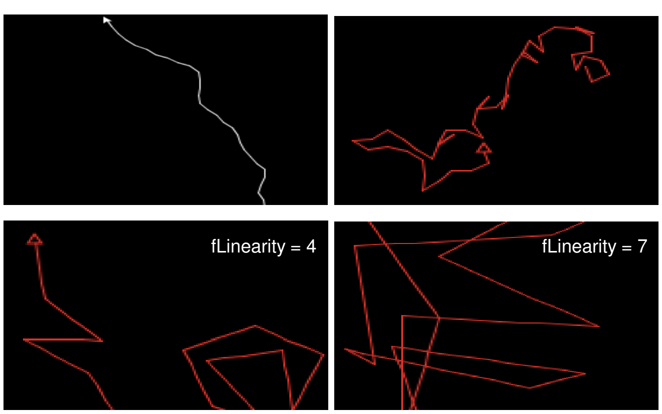

Fig. 7. (Left) Path followed by an Ant-Like robot. Firefly-Like motion (right) with different values of `fieldLinearity`.

e.g. when a robot touches an object, it creates a potential field around itself. Potential fields exist as a patch variable `field`; hence a robot is essentially manipulating a patch variable while generating the field. However, interaction with the world is limited in the way that the world may not directly affect any robot or patch, rather the world preserves certain states such as `robotEnergy` (the collective ability of swarm surrounding an object to perform a task) which may indirectly affect the decision making process.

Since the system is decentralized, the agents in the swarm behave autonomously without having to depend upon other agents in order to accomplish their tasks. The swarm does not rely upon any central controller, not even for complex decision making which is accomplished merely depending upon the behaviors of robots mentioned in the next section.

B. *Swarm Behavior*

The basic algorithm is explained in Fig. 6 and is common for both breeds of robots; however the implementation of internal modules is different. A robot starts off by first checking if there are any objects in its vicinity. If not, it first looks around to check for any object by calling the function `Look()`. The function `Look()` has two implementations one for each breed of robots. The difference is that Ant-Like robots can only look around for an object in a limited area which is governed by the variable `antVision` while Firefly-like robots have a 360^{o} viewing angle.

If the robot has not found any objects until now, it prepares to move, first by calling the function `Turn()` which sets the heading of a robot for its next move. This function gives rise to the movement models. As in [1] Ant-Like motion is rectilinear followed by random sharp turns (Fig. 7 top left), while Firefly-Like motion is Brownian motion (Fig. 7 top right). At this stage, a robot again checks the possibility of finding an object, failing which it calls the function `Mov()`.

The function `Mov()` in Fig. 8 has different implementations for both breeds of robots and is designed to investigate different behaviors. Ant-Like robots only check the possibility of a collision, and then move forward in a direction that has already been adjusted by the function `Turn()`. The implementation for Firefly-Like robots is however different and is more complicated. Their motion is affected by two factors, namely `fieldLinearity` and `fieldDefiance`. The variable `fieldLinearity` introduces small rectilinear intervals between their random turns (Fig. 7). They take a number of steps forward (defined by `fieldLinearity`) in their present direction, before setting their heading to a random direction as shown in Fig.8.

The second factor governing Firefly-Like motion is the variable `fieldDefiance`. In [1] we added a field defiance factor to both kinds of motion which gave less flexibility in our previous investigations. Robots inside a potential field should ideally follow the field. This however results in an undesirable effect where the whole robot population ends up

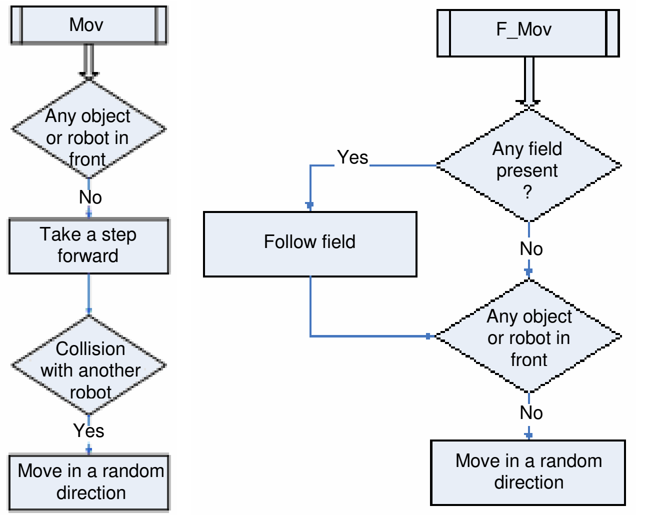

Fig. 8. Mov and F_Mov. The prefix 'F' corresponds to Firefly-Like robots.

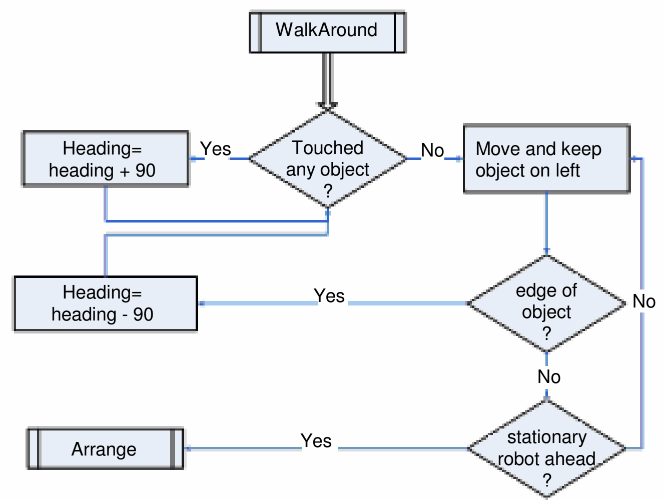

Fig. 9. The function WalkAround introduces behavior based cooperation, formation and marching capabilities.

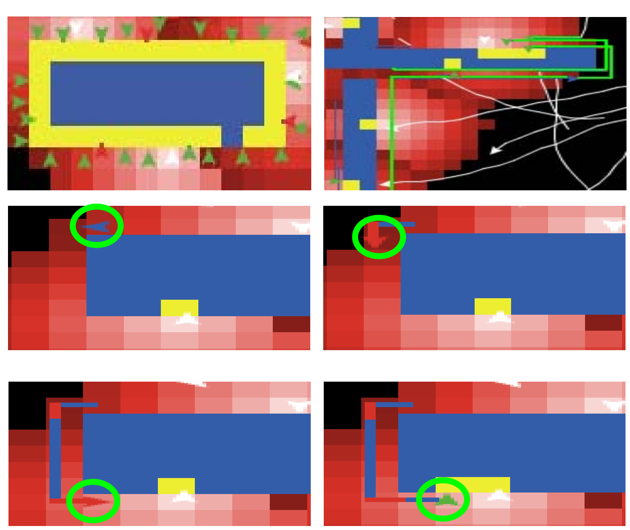

Fig. 10. Marching and Formation. The robots in a ring formation (top left). Robots marching around a '+' shaped object(top right). The rest of the images focus a robot that marches around the object after finding an object inside a field, and finally arranges itself next to the first robot that found the object. This behavior results in a ring formation (top left).

at a point of lowest potential, whereas the idea behind introducing potential fields is to allow robots to continue their search within a constricted area around the object. The variable `fieldDefiance` introduces a factor of randomness that prevents robots from gathering at one point within the field.

The Function `WalkAround()` (Fig. 9) makes the robots march around an object and end up in a ring formation. The arrangement of robots around the object is because of stigmergy. It has already been mentioned that a robot creates a potential field when it touches an object in the absence of a field i.e. when it is the first robot to find an object. However, if a robot touches an object within a field, it realizes that another robot near by has found the same object and is looking at it. The robot then tries to find the stationary robot by using the algorithm in Fig. 9.

The result of this behavior is that the robots arrange themselves in a form that is visible in Fig. 10 (top left). In the figure, white colored robots are the first to find an object and hence they generate a field around themselves.

A robot turns green when it has arranged itself around an object (Fig. 10 bottom right). The blue and red colored lines are the trails left behind by a robot while it marches around an object in search of its proper position. The object itself is shown in blue. If part of the object is in contact with an agent it is shown the yellow.

III. RESULTS

The effects of various system parameters on convergence time, and hence the ability to complete a task were investigated. These parameters are quite large in number as shown in Fig. 11, including `fieldDefiance`, `antVision`, `fieldLinearity and signalRadius`. The experiments were initially conducted with discrete breeds of robots followed by a study of their interoperability. The world size was set to 72x46 patches and the results were averaged over 50 runs, so that they can be statistically analyzed.

Given an overall complex task that cannot be accomplished by a single member of swarm, each agent has the capability of contributing to the collective goal. This individual capability of robots is called the Robot Energy. Fig. 12 shows an exponential decrease in convergence time while the robot energy increases. This rate of decrease is significant when the energy increases from 2 to 5, followed by the smoothening of the curve.

In Firefly-Like robots, however this effect is not very smooth as shown in Fig. 13. The graph shows a steep decrease in convergence time when the robot energy increases from 2 to 4 and from 9 to 11. This decrease is at a smaller rate between robot energy 4 and 9.

As expected, an increase in swarm population results in a decrease in convergence time. Fig. 14 shows this phenomenon in Ant-Like robots while Fig. 15 corresponds to Firefly-Like robots. The rate of decrease is a smooth curve in the case of Firefly-Like robots as well as in Ant-Like robots. However in the case of Firefly-Like robots, a clearly visible breakeven

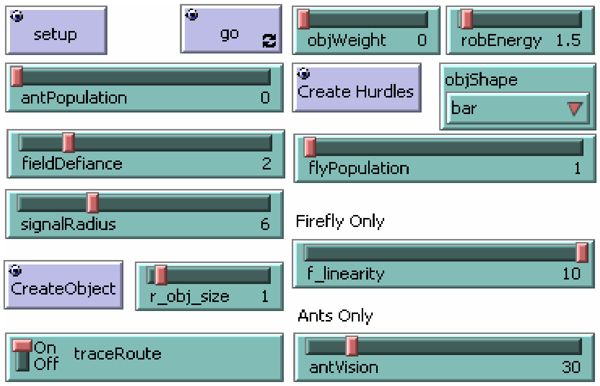

Fig. 11. The NetLogo control panel. The sliders and buttons allow changes in a number of system parameters.

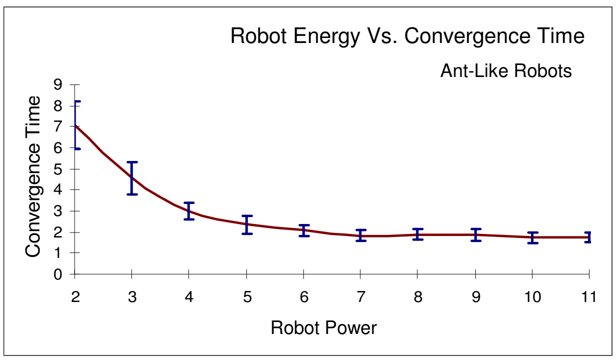

Fig. 12. Robot Energy Vs. Convergence Time in Ant-Like robots. The curve starts to smoothen at Energy of 4, and almost straightens at7.

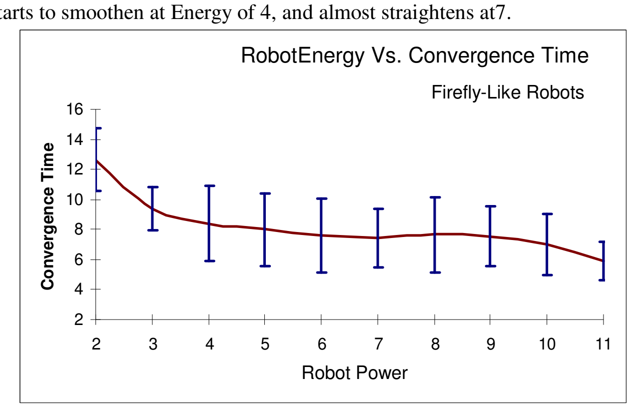

Fig. 13. Robot Energy Vs. Convergence Time in Firefly-Like robots. For robot energy of 2 to 6, the curve is similar to that of Ant-Like robots.

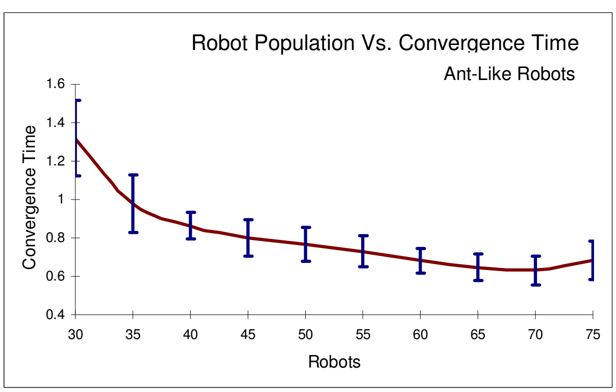

Fig. 14. Swarm Population Vs. Convergence Time for Ant-Like robots

occurs at a population size of 55, after which the convergence time starts to increase.

Although not very clear in Ant-Like robots, a similar affect occurs when the population increases over 70 robots. This, we believe is because an increase in population results in increased collisions between robots, hence is dependent on population density.

The spatial distribution of an object also effects the convergence time. The swarm takes a significantly longer time to converge around a bar shaped object as compared to a cross (+) shaped object where the surface area of both objects is the same. Fig. 16 and 17 show this phenomenon.

The variable antVision (the angle of vision of ants) has already been explained. An increase in antVision initially results in a decrease in convergence time. There is however a tradeoff to this and the convergence time actually starts increasing once the antVision exceeds 105º Fig. 18.

In Firefly-Like Robots, fieldLinearity introduces small rectilinear movements between random turns at large angles and has an effect visible in Fig. 19, similar to the effect seen for Robot Energy (Fig. 11).

Potential fields attract any Firefly-Like robots entering the field towards an object. The variable signalRadius governs the radius of these fields. Fig. 20 shows that the convergence time first increases with an increase in the radius but after reaching a breakeven, suddenly starts decreasing and hence its value should be chosen very carefully.

Fig. 21 shows how field defiance affects convergence time in Firefly-Like robots. Note that fieldDefiance does not have any effect on movement of Ant-Like robots. This effect has intentionally been introduced in order to investigate different behaviors.

The graph in Fig. 22 shows the affect of the ratio between Firefly-Like and Ant-Like robots on convergence time. Ant-Like robots converge in less time as compared to Firefly-Like robots. The curve shows how convergence time varies for different populations of Ant-Like and Firefly-Like robots.

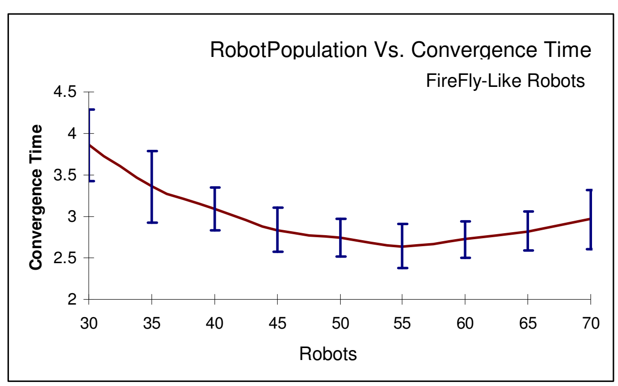

Fig. 15. SwarmPopulation Vs. Convergence Time for Firefly-Like robots. Minimum convergence time occurs at robot population 55, beyond which convergence time actually increases.

The above analysis leads to the fact that convergence time depends on a number of factors as given by the following relationship:

$$t \propto \frac{f_d}{R_s \times N_r \times E_r} \tag{2}$$

where t is the convergence time, f_d is field defiance, R_s is the signal radius (greater than the minimum threshold), N_r is the robot population (less than the maximum threshold) and E_r is the robot energy.

Fig. 23 demonstrates shape extraction and due to a varying number of robot agents in the arena. In the figures, the blue patches correspond to a bar shaped object while yellow patches are the points on the boundary of the object that has been recognized by the robots. The shape is represented by the x-y coordinates of robots that have found an object and have correctly arranged themselves around the object.

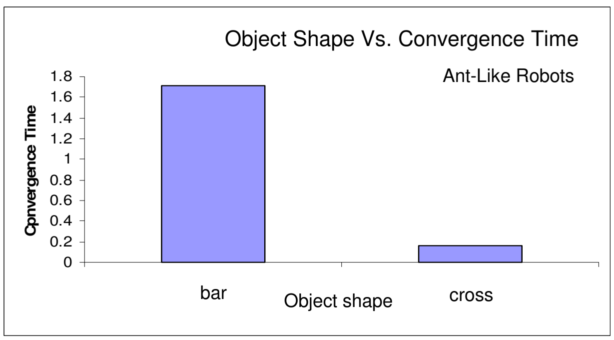

Fig. 16. Object Shape Vs. Convergence Time for Ant-Like robots

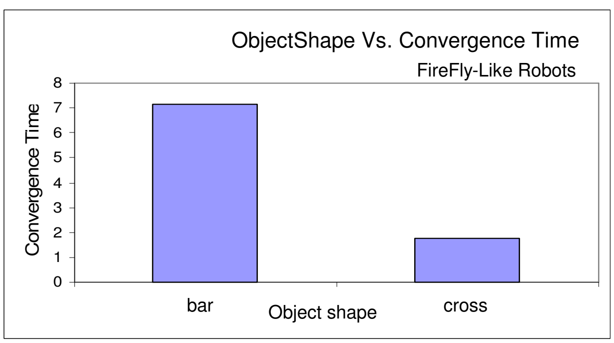

Fig. 17. Object Shape Vs. Convergence Time in Firefly-Like robots

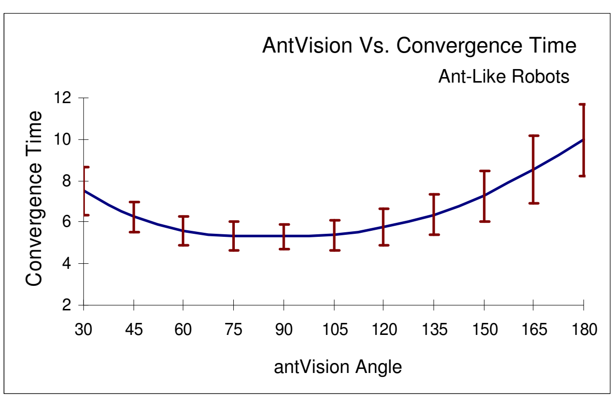

Fig. 18. AntVision Vs. Convergence Time in Ant-Like robots

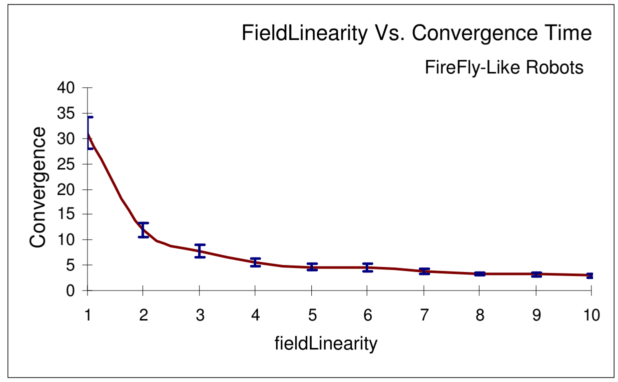

Fig. 19. Field Linearity Vs. Convergence Time in Firefly-Like robots

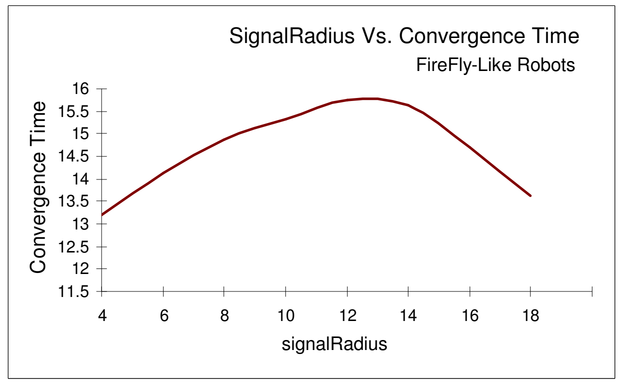

Fig. 20. Signal Radius Vs. Convergence Time in Firefly-Like robots

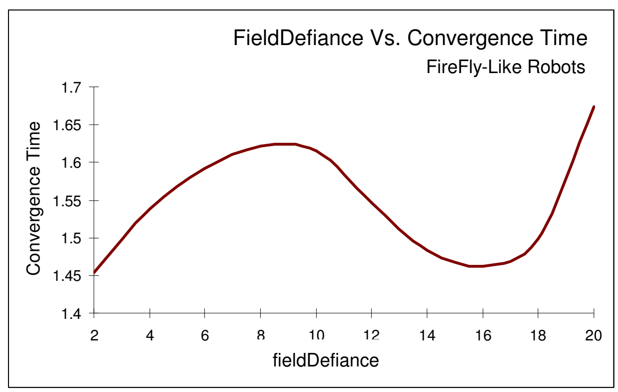

Fig. 21. fieldDefiance Vs. Convergence time in Firefly-Like robots

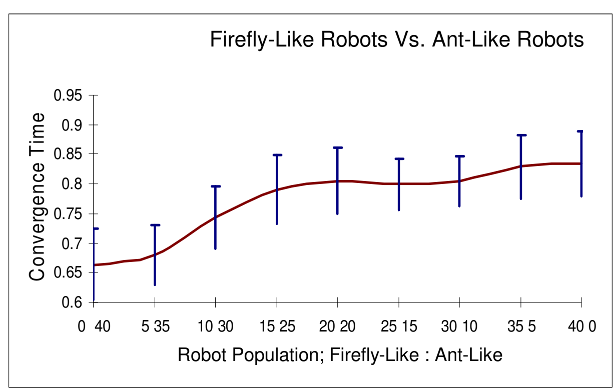

Fig. 22. Firefly-Like : Ant-Like robots Vs. convergence Time

Fig. 23. Shape extraction, over time, of a bar shaped object using a varying number of robot agents. The yellow patches are patches that are in contact with a robot.

The top left image is generated when the first robot touches the object and hence stops moving. The rest of the images show how the perception of object improves as more and more robots find the object. The bottom right image is the final shape extracted with 19 robots. Three visible disconnections in the boundary formed around the object are due to the physical locations of robots that did not allow further robots to complete the boundary. We achieved a 90% shape extraction averaged over 50 runs. The percentage of the shape extracted is determined by

$$n = \frac{\textit{number of robots surrounding the object}}{\textit{number of robots required for 100\% shape extractions}} \quad (3)$$

IV. CONCLUSION

In this paper we have extended the concept of Incremental Perception into the decentralized domain. By doing so, the requirement of a central controller is removed and thus true swarm behavior can be achieved. We have further identified the parameters and functions required for modeling a fully decentralized system.

Two kinds of swarm movements were modeled: Ant-Like and Firefly-Like. The tradeoffs between the different parameters have been experimented on and analyzed. In the future, we will extend the study to include environments with multiple simultaneous objects. In this scenario an optimal number of agents are required to perceive the object efficiently. We postulate that repulsive potential fields can be used to prevent overcrowding of agents around single objects, and to reduce the probability of collisions. Further investigation will also be carried out to determine the relationship between object shape/size to population density.

[11] presents a theoretical framework for design and analysis of distributed flocking algorithms in free-space and in the presence of multiple obstacle where as our experiments, so far, have been carried out in a 2-D environment, and in the presence of only one stationary object. Our future experiments will be carried out in the presence of multiple moving objects.

V. REFERENCES

[1] I. Mir, B. P. Amavasai and S. Meikle, “Incremental Perception in Robotic Swarms,” 10th IEEE International Multitopic Conference (INMIC 2006), Pages 427-432.

[2] Y.U. Cao, A.S. Fukunaga, A.B. Kahng, “Cooperative Mobile Robotics: Antecedents and Directions,” Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 95. ‘Human Robot Interaction and Cooperative Robots’, 1997.

[3] G. Beni, "From Swarm Intelligence to Swarm Robotics", Proc. of Workshop on Swarm Robotics, 8th Int'l Conf. on Simulation of Adaptive Behavior (SAB '04), Los Angeles, USA., published in Lecture Notes in Computer Science, Springer-Verlag, July 2004.

[4] T. Balch, “Communication, Diversity and Learning: Cornerstones of Swarm Behavior,” in *Swarm Robotics,* Sahin E., and Spears, W. (eds). Springer Verlag. 2005.

[5] J. Seyfried, M. Szymanski, N. Bender, R. Estana, M. Thiel, H. Wörn, “The I-SWARM project: Intelligent Small World Autonomous Robots for Micro-manipulation,” From Animals to Animats 8: The Eighth International Conference on the Simulation of Adaptive Behavior (SAB'04), 2004.

[6] E. Sahin, “Swarm Robotics: From Sources of Inspiration to Domains of Application,” In Swarm Robotics: State-of-the-art Survey, Lecture Notes in Computer Science, pages 10-20, Berlin Heidelberg, Springer-Verlag. 2005.

[7] U. Wilensky, NetLogo. http://ccl.northwestern.edu/NetLogo/, Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL. 1999.

[8] S. Nouyan, M. Dorigo, “Chain Based Path Formation in Swarms of Robots,” in Proceedings of ANTS2006, pages 120-131, Springer Verlag, Berlin, Germany, 2006.

[9] M. Mataric. Designing emergent behaviours: From local interactions to collective intelligence. In Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats 2, pages 432-441, 1992.

[10] M. Mataric. Minimizing complexity in controlling a mobile robot population. In proceedings of the 1992 IEEE International Conference on Robotics and Automation, pages 830-835, Nice, France, May 1992.

[11] R. Olfati-Saber, "Flocking for Multi-Agent Dynamic System: Algorithms and Theory," IEEE Trans. on Automatic Control, vol. 51, no. 3, pp. 401-420, March 2006.