Simulation of Multirobot Movement Algorithms for Entity Detection

Dayal C. Sati
Department of ECE
B.R.C.M. CET Bahal, Bhiwani
(Haryana),India

Pardeep Kumar
Department of Electronics
Banasthali University,Tonk
(Rajasthan), India

Monika
Department of CSE
Banasthali University,Tonk
(Rajasthan),India

ABSTRACT
The problem which is addressed in this paper is to find entities by multi-robot in an unknown environment. Here it is examined that how the choice of movement algorithm can affect the success of finding the entities in an unknown environment. Assumptions are that there is no central control, and robots have simple processing power and simple sensors and no active communication. Three different movement algorithms are evaluated which can gain good performance in the different unknown environment.

Keywords
Multi-robot, Cellular space, Lego robot.

1. INTRODUCTION
There has been a growing interest in multi-robot coordination research in recent years. Compared to single robot, coordinating multiple robots can lead to faster task completion, higher quality solutions, as well as increased reliability and robustness in case of robot failure. Behaviour-based architectures are commonly used to control individual robots and robot teams. They offer the advantage of being flexible, robust, and reactive. The problem is to find entities by a group of robots in an unknown environment so as to cover the environment as much as possible while staying within communications range. The assumptions are that there is no central control, the environment is unknown, the robots operate independently, with limited communications with the other team members, and they have limited sensing capabilities. This paper present different algorithms and validate them experimentally using a simulation environment.

The primary motivation for this work comes from the need to develop robust and reliable methods that are applicable to very small robots which operate in unknown complex human-made environments. One of the major challenges that need to be addressed when using very small robots is their extremely limited ability to estimate their own location that is further complicated by the robots limitations in communications range, computing power, and suite of sensors. The approach proposed here uses basic behaviours to control the motions of each robot so that the robots will move in the environment without the need for any centralized control. Here the three different algorithms that helps in robot movement and finding entity in the environment. Algorithms are validated experimentally in simulation.

In Section 2 relevant background literature is discussed. Algorithms are presented in Section 3, followed in the Section 4 by experimental results, where the performances of the algorithms are measured in simulated environment. Finally Section 5 has been wrapped up with conclusions and discussion of future work.

2. RELATED WORK
In 1992 Gage was the first to consider the problem of area coverage by a team of robots [1]. He differentiates the problem into three types: blanket coverage, barrier coverage, and sweep coverage. Blanket coverage, the most similar to problem of this paper, has the objective of maximizing the total area covered by a static arrangement. An experiment by Howard, Mataric, and Sukhatme considers how to deploy a mobile sensor network in an unknown environment [2]. They use robots equipped with a 360 degree laser range finder. No wireless communication is present. The robot behaviour is based on potential fields.

Basically robots are repelled by other robots and walls. Their results are impressive, but this approach is not possible for very small robots due to the large sensors required. Hsiang et. al [3] use a leader-follower approach based on local rules where the robots makes chains emanating from a single source of robots. The robots follow walls by keeping the walls on their left. This simulation experiment was run in a discrete grid world and assumes “local sensors.” If this algorithm could operate well in a more realistic simulation environment, such as provided by Player/Stage [4], while using only small proximity sensors for following robots. Batalin and Sukhatme [10] rely on the deployment of beacons into the environment to help coordinate a decentralized algorithm that uses only local interactions between the robots and beacons to cover an unknown area. For this approach robots must be large enough and capable of carrying the static beacons.A small robot can accomplish the same task as a static beacon by simply remaining stationary.

In 1999, Spears and Gordon provided decentralized control of large collections of agents by having agents react to artificial forces motivated by natural laws of physics, observing that in the real physical world surprisingly complex behaviours arise from simple interactions between entities. However, their applications were self-assembly and self-repair rather than dispersion for the purpose of surveillance [12]. In another virtual physics approach, Howard et al. used a potential-field-based approach" to the deployment of a mobile sensor network by treating their robots as virtual particles subjected to virtual forces [1]. These forces cause each given robot to be repelled from the other robots as well as from other obstacles. In the environment with a potential that is proportional to the sum of the reciprocals of the distances from the first given robot. Howard et al. continued to run their algorithm until the whole network reached a static equilibrium, while in this paper after the initial dispersion, other robot behaviours such as locating a specific goal are allowed to operate.

3. MOVEMENT ALGORITHMS

The purpose of this paper is to show that how the selection of movement algorithms for a multi-robot system affects the coverage of robot observation in a given environment. Three distinct movement algorithms are considered here, all of them reactive in nature. Each movement algorithm controls two types of movements: forward/backward and turning left/turning right. Turning can occur in place or while the robot is moving. The sensors available to the movement algorithms are ultrasonic sensors, each of which returns the distance of the nearest object detected in the direction in which the ultrasonic sensor is pointing. Robots have only local knowledge, they are not under the global control and do not have any knowledge of the environment other than what they can detect with theirs sensors.

3.1 Random Walk

The Random Walk Dispersion movement algorithm is the most basic algorithm. A robot using this algorithm can be in one of two states: random forward movement or entity detection. In random forward movement robot moves forward with a random turn factor of -90 or 90 degrees which is changed at random intervals and whenever robot detects the entities. Whenever the robot detects that it has encounters a wall, it enters the wall avoidance state. In this state, the robot will stop and turn and transition back to the forward movement state.

3.2 Wall Following

The idea behind Wall Following Dispersion algorithm comes from the fact that in many indoor environments, if a robot could find an inner wall of the building and follow it, the robot would be led through the much of the structure. The robot using the Wall Following algorithm will search for the entities and proceed to follow that wall indefinitely. In this algorithm a robot has four states: find wall, align to wall, follow wall and navigate the corner. If the robot believes that it had lost the wall in any of the three non-wall-finding states, it will reset back to the initial find wall state and search for wall to follow. The major problem with this algorithm is that it assumes every entity encountered is a wall. Because of this when many robots using this algorithm are together they will tend to perceive each other as wall or entity and try to align themselves to each other.

3.3 Cellular Space

In this mode, we assume the entire environment to be a cellular space divided into equally spaced grids. The extreme most cells of the grid are considered to be the boundary cells, out of which the robot will not move. This algorithm considered each cells as a cellular automaton which would be in one among the following states:

S_f : Free state i.e. no entity in the cell

S_e : Entity state i.e. There is an entity in the cell

S_b : Boundary state ie the cell is a boundary cell

Every time the robot visits a cell if it is in a free state. Then the neighbourhood function is called. The neighbourhood considered here is the selective neighbourhood in which we consider only the front left and then the right cells. We calculate the distance between the robot and the entity for each of the neighbours and take the movement decision based on whether we consider moving the robot towards the maximum distance or towards the minimum distance.

The cellular function can be written as follows:

$$S(t+1) = \max\{d(S_f(t)), d(S_l(t)), d(S_r(t))\}$$

or

$$S(t+1) = \min\{d(S_f(t)), d(S_l(t)), d(S_r(t))\}$$

Where S_f (t) is the front neighbourhood distance of the robot at time t, S_l(t) is the left neighbourhood distance of the robot at time t, and S_r (t) is the right neighbourhood distance at time t.

4. SIMULATION

The objective of the simulation is to compare the no of entities find by the three algorithms in different time in an unknown environment. To compare the algorithms, we perform a large number of experiments within SPL simulator. The virtual robot used for the experiments is a three wheeled, Lego Tribot.

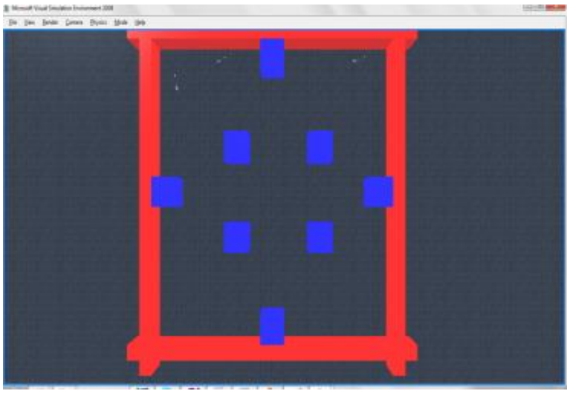

Figure 1. Simulation Environment with three robots

The robots used in the simulations have 1 laser range finder sensor. Each result is averaged over 10 runs. For the purpose of observing the working process more clearly, only1, 3, 5, 7 and 9 robots are used. Robots only have local knowledge i.e., they are not under global control. The only environment information is from the sensors. The environment is a space with the size of 10*10. The entities are multiple of 8 depends on the number of robots in the environment. Each algorithm is repeatedly executed with different no of robots and collect the number of entities detected in
60, 120, 180, 240, and 300 seconds. The simulation environments with three robots are shown in the figure1.

Table 1 summarizes the results from single robot on three algorithms. The percents value in the table indicates the percentage of the entities found in the environment by the robot in different time stamps in seconds.

Table 1. Result for single robot

No. of entities detected in %						
	Random Walk		**Wall Following**		**Cellular Space**	
Time (sec)	**Trial1**	**Trial2**	**Trial1**	**Trial2**	**Trial1**	**Trial2**
60	12.50%	12.70%	12.20%	12.90%	25.20%	25.90%
120	25.20%	25.20%	37.80%	37.50%	50.60%	50.30%
180	37.30%	25.20%	50.60%	50.40%	62.60%	62.50%
240	50.20%	50.50%	50.20%	50.50%	87.20%	87.80%
300	62.50%	62.80%	62.50%	62.80%	92.30%	92.80%

Table 2 summarizes the results from multi-robots on the Random Walk algorithm.

Table 2. Result for random walk algorithm

Time (sec)	**Random Walk**			
	3	**5**	**7**	**9**
60	12.60%	14.20%	18.20%	20.20%
120	22.20%	24.10%	28.50%	29.30%
180	35.30%	37.80%	34.30%	34.30%
240	49.50%	51.20%	53.90%	55.60%
300	60.40%	61.80%	62.50%	64.10%

Table 3 summarizes the results from multi-robots on the Wall Folowing algorithm.

Table 3. Result for Wall Following algorithm

Time (sec)	**Wall Following**			
	3	**5**	**7**	**9**
60	13.40%	14.60%	19.80%	21.10%
120	37.80%	38.20%	39.50%	40.30%
180	49.40%	50.20%	51.40%	51.60%
240	48.50%	49.70%	50.80%	52.60%
300	60.20%	61.40%	62.60%	62.90%

Table 4 summarizes the results from multi-robots on the Cellular Space algorithm.

Table 4. Result for Cellular Space algorithm

Time (sec)	Cellular Space			
	3	5	7	9
60	25.20%	25.30%	27.60%	27.90%
120	50.20%	51.60%	53.60%	53.90%
180	60.30%	61.50%	62.70%	64.80%
240	85.90%	86.40%	87.90%	88.00%
300	92.60%	94%	95.20%	96.80%

The Graph that compares the performance between three algorithms is shown in given figure 2.

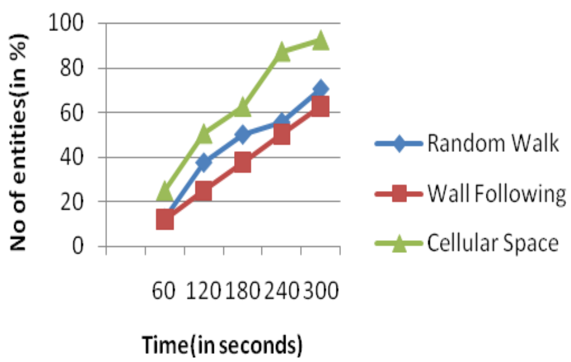

Figure 2. Performance Comparioson

5. RESULT ANALYSIS

The result shows that the Cellular Space algorithm performs the best in an unknown environment. This is not surprising in that this algorithm has to find more entities than other algorithm. It does however indicate that the knowledge of the neighbourhood can help to speed up the exploration and searching process.

The Random Walk algorithm performed second best among these algorithms tested. The main reason behind the performance of this algorithm is that the robots take random turns which increase its exploration and searching capabilities.

The Wall following algorithm has some flaws in it. The major flaw is that the robot treats the other robot as a wall i.e. the algorithm has no ability to distinguish between robots and wall.

6. CONCLUSION AND FUTURE SCOPE

This paper examined the performance of several movement algorithms in finding entities in an unknown environment. The algorithm has been tested by different number of robots in an unknown environment. The results shows that knowledge of the neighborhoods helps the robot to explore and search entities. The combination of this algorithm and a group of robots would make for an effective system for exploring and finding more entities of interest in an unknown environment.

7. REFERENCES

[1] T.Arai, E. Pagello, L.E. Parker, "Guest editorial: advances in multirobot systems." IEEE Trans. on *Robotics and Automation*, Vol.18, No. 5 pp.655-661, oct. 2002.

[2] A.Brumitt, B.Stentz. 'Dynamic mission planning for multiple mobile robots'. In IEEE ICRA'96, 1996.

[3] T.R. Hsiang, E. Arkin, M. A. Bender, S. Fekete, and J. Mitchell. Algorithms for rapidly dispersing robot swarms in unknown environments. In Proc. 5th Workshop on Algorithmic Foundations of Robotics (WAFR), 2002.

[4] Alberto Bressan and Wen Shen, "semi cooperative strategies for differential games," International Journal of Game theory, ,vol. 32 issue 4, pp561-593,2004

[5] Y.Cao, Fukuna, and A.Kahng. Cooperative Mobile Robotics: Antecedents and directions. *Autonomous Robots*, 4:7-27, 1997.

[6]Vail and Veloso, "Dynamic Multi-robot Coordination," Multi-Robot Systems: From Swarms to Intelligent Automata, vol. II, pp. 87-100,2003.

[7] Sen, S., Sekaran, M.,Hale, J., "learning to coordinate without sharing information" AAAI-94, pp.426-431.

[8] D. W. Gage. Command control for many-robot systems. In 19th Annual AUVS Technical Symposium, pages 22–24, Huntsville, Alabama, June 1992

[9] A. Howard, M. J. Mataric, and G. S. Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In Proc. Int'l Symp. on Distributed Autonomous Robotic Systems", 2002.

[10] L.E. Parker, ALLIANCE: 'An architecture for fault tolerant Multi Robot Cooperation ', IEEE Transaction on Robotics, vol.14 (2), pp.220-240, 1998.

[11] A. Zelinsky. A mobile robot exploration algorithm. *IEEE Transactions on Robotics and Automation*, 8(2): 707-717, 1992