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 presents 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 help 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 surveying [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 force, the robots repel each other. The robots repelled from the other robots as well as from other obstacles. In the environment with a potential that is proportional to the force, the robots repel each other.
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:

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

or

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

Where \( S_l(t) \) is the front neighbourhood distance of the robot at time \( t \), \( S_f(t) \) 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.

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, only 1, 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 figure 1.

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

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Random Walk</th>
<th>Wall Following</th>
<th>Cellular Space</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trial 1</td>
<td>Trial 2</td>
<td>Trial 1</td>
</tr>
<tr>
<td>60</td>
<td>12.50%</td>
<td>12.70%</td>
<td>12.20%</td>
</tr>
<tr>
<td>120</td>
<td>25.20%</td>
<td>25.20%</td>
<td>37.80%</td>
</tr>
<tr>
<td>180</td>
<td>37.30%</td>
<td>25.20%</td>
<td>50.60%</td>
</tr>
<tr>
<td>240</td>
<td>50.20%</td>
<td>50.50%</td>
<td>50.20%</td>
</tr>
<tr>
<td>300</td>
<td>62.50%</td>
<td>62.80%</td>
<td>62.50%</td>
</tr>
</tbody>
</table>

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

Table 2. Result for random walk algorithm

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Random Walk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>60</td>
<td>12.60%</td>
</tr>
<tr>
<td>120</td>
<td>22.20%</td>
</tr>
<tr>
<td>180</td>
<td>35.30%</td>
</tr>
<tr>
<td>240</td>
<td>49.50%</td>
</tr>
<tr>
<td>300</td>
<td>60.40%</td>
</tr>
</tbody>
</table>

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

Table 3. Result for Wall Following algorithm

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Wall Following</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>60</td>
<td>13.40%</td>
</tr>
<tr>
<td>120</td>
<td>37.80%</td>
</tr>
<tr>
<td>180</td>
<td>49.40%</td>
</tr>
<tr>
<td>240</td>
<td>48.50%</td>
</tr>
<tr>
<td>300</td>
<td>60.20%</td>
</tr>
</tbody>
</table>

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

Table 4. Result for Cellular Space algorithm
The Graph that compares the performance between three algorithms is shown in given figure 2.

### Table 1: Performance Comparison

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Cellular Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>25.20%</td>
</tr>
<tr>
<td>5</td>
<td>25.30%</td>
</tr>
<tr>
<td>7</td>
<td>27.60%</td>
</tr>
<tr>
<td>9</td>
<td>27.90%</td>
</tr>
<tr>
<td>120</td>
<td>50.20%</td>
</tr>
<tr>
<td>180</td>
<td>60.30%</td>
</tr>
<tr>
<td>240</td>
<td>85.90%</td>
</tr>
<tr>
<td>300</td>
<td>92.60%</td>
</tr>
</tbody>
</table>

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


