Improvement of programmable self-assembly in a thousand-robot swarm

Fernando Cortés Sancho

Universidad Politécnica de Madrid, Spain

Abstract. One of the multiples applications of Multi-agent Systems is assembly. Here in this paper an improvement for the (programmable self-assembly in a thousand-robot swarm) is proposed, with a robot expert that will provide the real localization to the robots, to avoid imperfections in the shape. This will be shown in a 2D simulation.

Keywords: Self-Assembly · Multi-agent systems · Kilobots.

1 Introduction

1.1 Previous work

The work proposed is based on a previous work [2], in which a thousand Kilobot[1] swarm works collectively in order to form a desired shape.

This is achieved by the self-assembly algorithm proposed in that paper, which can be divided in three main algorithms that work together inside every robot's mind.

First one is the edge-following algorithm, which is controls the movement of the robots throughout the space. Every robot has an area of a fixed radius than can be used to communicate with robots that are close, meaning, neighbors. With that area, robots will be able to move along the swarm, surrounding it and arriving to the its final position inside the shape.

Second one is the gradient formation algorithm, which is used to maintain an organization inside the swarm. There are 4 robots, that will be called 'seed' robots, that mark the figure that needs to be formed by surrounding one of its corners. By knowing which gradient does the robot has in relation with the seeds, thus, how far is the robot from them, it will be able to know its role and will know whether it has to wait, to move or to join the shape.

Third and last algorithm is the localization algorithm, used by the robots to determine its position and conclude if they are inside the desired figure or not. This is done by trileration and the area mentioned before. Robots, using at least three localized robots (robots that already know their location), will be able to set their location.

1.2 New proposal

A series of simulations were run to determine how good was the previous implementation. In those simulations, I could notice some errors due to a mismatch between the real location and the believed location of the robots. That could mean not only imperfections in the shape but also errors in the gradients.

For a robot to determine its location, a minimum of 3 localized robots are needed. It is intuitive that the more robots helping to calculate the location, the more accurate it is going to be. The same can be applied to the opposite. Sometimes robots do not have enough localized robots near to establish the position correctly.

Besides, this error scales with the shape of the figure as a result of little errors in every robot that are added every time a new robot is calculating its position.

This is why I proposed the use of an expert robot, for example, AR.Drone [4] to provide the real location to the robots in real time, in a similar way as other projects [4].

In the experiments sections, a 2-d simulation with a 60 pixelsx 60 pixels square shape will be run in order to prove the enhancement.

2 New method proposed

The new method, which has already been mentioned above, consist of adding a new robot to the swarm that has the ability to know and provide the location of every robot of the swarm. This could be done with an AR.Drone modified to perform this task.

The robots, instead of calculating their location every time they move, would have to communicate with the expert robot. This will also implies less calculations per robot, because it will not have to do the triletarion algorithm, though it will not mean any big improvement in terms of computational efficiency.

This modifications of the original paper prevents the robot from being lost or from thinking they are where they are not. And, in consequence, robots will perfectly know whenever they are inside the shape and whenever they are not, so they will be able to decide to stop when they are about to leave the desired figure.

Experiments in the next section will show how the addition of a robot master is reflected in a soften of the shape, reducing the errors and imperfections.

However, this improvement may change the aim of the original paper, because swarm robots will not be self-autonomous anymore and the concept of self-assembly gets diluted. Nevertheless, results show a sufficient improvement to consider this alternative.

[**?**,**?**,**?**], [**?**,**?**,**?**,1].

3 Experiments

In this section, a series of experiments are running in a 2-d simulation. In the simulations robots have 10 pixels as radius.

In the simulation, the robot expert works as the program itself, providing the robots with their real location.

In those experiments green circles represent seed robots, rose circles represent robots that already joined the shape and the gray constitute the desired figure, so that it can be better observed.

Beside, every robot has their own gradient in the middle.

The first one, that can be seen in Fig. 1 consists of a 60px60p square shape and 9 robots trying to form that shape. This enviroment is chosen based on its simplicity.

In the real world, the swarm is formed by a thousand robots, but in order to observe the failures, not that many robots are needed.

As it is shown in Fig. 1a, some robots, like the ones with gradient 1 and 2 below, are far out of the shape, due to the error in their believed location. Nonetheless, the rest of the robots seem to have filled the gray region correctly. That mean that 2 out of 9 robots have misled.

On the other hand, in Fig. 1b, no robot out of the shape can be seen. The simulation was run with only 9 robots but it is intuitive that with 2 more robots, the shape would be completely filled.

A clear improvement can be seen in these two figures, and as it was mentioned before, the bigger the shape, the bigger the number of robots needed and then, the probability of failures increments.

(a) Original method

(b) Method proposed

Fig. 1: Self-Assembly in a 60px60p square with 9 robots

The efficiency of this improvement can also be shown in non continuous shapes. For the next experiment,shown in Fig. 2. a shape of 60px60 square with a hole of 30px30p is run. The gray square represents the shape and the white one, in the center of the bigger one, is a hole, so it is not consider to be in the shape.

The original method, in Fig. 2a does not perform bad at all, just a robot, the one in the middle of the hole, can be consider that has failed.

However, the method proposed fits better the shape, as it does not have any robot of the swarm that is not in the shape. Furthermore, the hole has been avoided correctly, as there is no robot that has consider to stop inside it.

(a) Original method

(b) Method proposed

Fig. 2: Self-Assembly in a 60px60p square with a hole in the middle with 9 robots

4 Conclusions

In this paper a new method is proposed for an improvement of programmable self-assembly in a thousand-robot swarm. The above-mentioned results in continuous and non continuous shapes show how the figure is better filled and imperfections are softened. This is why, despite changing a bit the paradigm of the original paper,this can be a good alternative to consider when building systems of this type.

References

1. Rubenstein, M., Ahler, C., Nagpal, R. (2012, May). Kilobot: A low cost scalable robot system for collective behaviors. In 2012 IEEE International Conference on Robotics and Automation (pp. 3293-3298). IEEE.
2. Rubenstein, M., Cornejo, A., Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. Science, 345(6198), 795-799.
3. Mathews, N., Valentini, G., Christensen, A. L., O'Grady, R., Brutschy, A., Dorigo, M. (2015). Spatially targeted communication in decentralized multirobot systems. Autonomous Robots, 38(4), 439-457.
4. Dijkshoorn, N. (2012). Simultaneous localization and mapping with the ar. drone. PhD diss., Masters thesis, Universiteit van Amsterdam.