[[1]](#footnote-1)

Swarm Robots Approach to Loose Fruit Picking

Si Hao Goh, Danny Wee Kiat Ng  
Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman (UTAR)  
Kajang, Malaysia  
[shawngoh87@hotmail.com](mailto:shawngoh87@hotmail.com), ngwk@utar.edu.my

*Abstract*—Mobile robot based loose fruit collection is a solution to improve efficiency of recovering oil palm fruitlets. This paper proposes a method to coordinate a swarm of mobile robots to explore an unknown terrain, and collect the loose fruits around explored oil palm plants. Each robot detects obstacle and sends the data to a server, while collecting the current state of exploration as a response. The delegation of tasks is then calculated based on the information.

*Index Terms*—mobile robot, differential drive, swarm robot, loose fruit collection

# INTRODUCTION

T

he oil palm tree is a tropical plant originated from West African tropical rainforest region, nowadays planted commercially in Malaysia. Oil palm’s fruit contains large amount of edible plant oil. Loose fruits are the ripest fruitlets fallen from the main cluster, and contain the highest amount of oil [1].

A conventional method for loose fruit collection includes manual collection and often cause back pain among workers. One solution deployed by other researchers is to use a remote-controlled mobile robot with a roller-type collecting cage. There are also proven solutions using autonomous vacuum robot [2].

To improve the solution, swarm robotics can be utilized to split the burden. Swarm robotics is an approach to realize a collective goal based on interaction between simple robots and their environment. This system can coordinate multiple robots based on each robot’s local information, and achieve equilibrium in the delegation of several tasks. Compared to centralized systems, swarm robots have relatively simple components, thus they can be modularized and mass-produced [3]. However, decentralized swarm robotics system is unable to access global data [4]. The lack of central control mechanism may cause swarm to be unpredictable and poses a difficulty in optimization [4].

The challenges to utilize swarm robotics are to identify the differentiation of roles and the algorithm for the said differentiation. In addition, certain centralized control can be added to enhance the efficiency and to enable the collection of data.

# Literature Review

In order to apply swarm robotics to the mobile robot based problem, the measurement errors from sensors need to be taken into account.

## Mobile Robot Localization

In order to track the current state of the mobile robot, the kinematics of a differential driving robot can be used for robot pose estimation [5]. The pose of a mobile robot, consist of Cartesian coordinate (*x, y)* and bearing , can be presented as a vector *p*:

(1)

The subsequent global pose *p’* at each sampling interval can be estimated with:

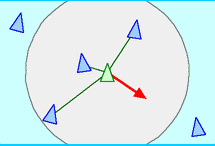
(2)

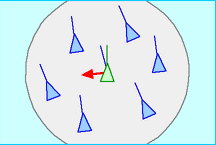
Where and is the displacement of the right and left wheel respectively in the last sampling interval. is the average of the two displacements. is the distance between the two wheels[5].

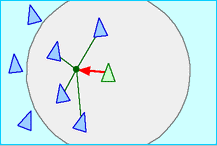
## Swarm Flocking Algorithm

Swarm flocking algorithm can be used for robot clustering and direct them to unexplored areas. The key concept in flocking includes separation, cohesion, and alignment [6].

Separation prevents collisions between the robots by adding a negative velocity proportional to the current displacements of local flockmates with respect to itself. Alignment steers the robots to the average heading of local flockmates. Cohesion steers the robots to the average position of local flockmates.

  
Fig. 1. Separation. The green robot will head away from the average position of the blue robots.

  
Fig. 2. Alignment. The green robot will adjust its heading according to the average heading of the blue robots.

  
Fig. 3. Cohesion. The green robot will head towards the average position of the blue robots.

Separation and cohesion are essentially two opposing forces that keep the swarm from colliding while keeping the general direction same [6].

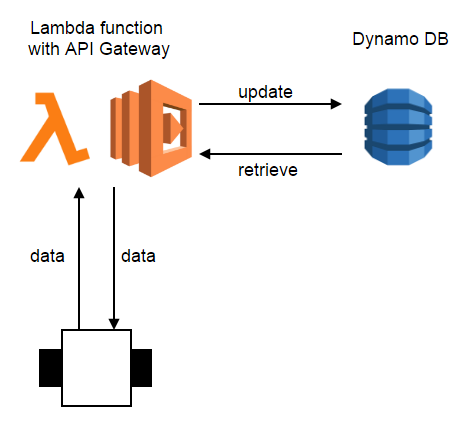
# Methodology

## Mobile Robot Construction

The embedded programming platform used is Eclipse IDE and C programming language is used to configure the robots. Each robot uses 1pcs Intel Edison microprocessor, 2pcs 30SPG- 20E DC motors with encoders, 1pcs HC SR-04 ultrasonic sensor, 2pcs motor drivers, and a 12V sealed lead acid battery.

## Swarm Communication Platform

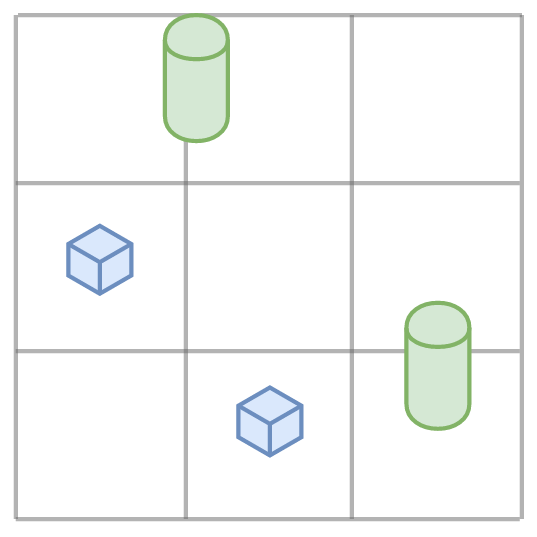
This system uses Amazon Web Services (AWS) as an information relay between robots.

  
Fig. 4. Swarm communication system based on various AWS services

An AWS Lambda function will store incoming data from the robots through AWS API Gateway and store them to Dynamo DB. After storing, the function returns the current state of all other robots as well as the environmental data to the requesting robot. This method is used to simplify the data transfer mechanism.

## Localization Approximation

This system uses a grid-based localization approach to approximate the environment and reduce computational load.

  
Fig. 5. Grid-based localization approach.

The above figure shows an example of grid-based approximation. The blue cubes are the robots and the green cylinders are the oil palm plants. The robots will always attempt to move to the center of a grid cell, unless it is obstructed by objects. The plants can appear at any location as it is the nature of oil palm plantations.

## Role Delegation

There are two roles presented in this problem, namely exploration and fruit collection. An explorer robot is a robot tasked for exploration and a worker robot is one tasked with fruit collection. Each robot may change roles at the start of each program iteration.

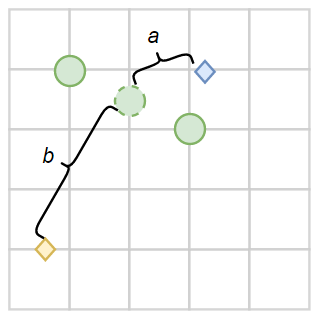
At any given time, in a field of *C* number of grid cells, *Cx* number of explored grid cells, and *n* number of swarm robots, the number of worker robots *nw* is:

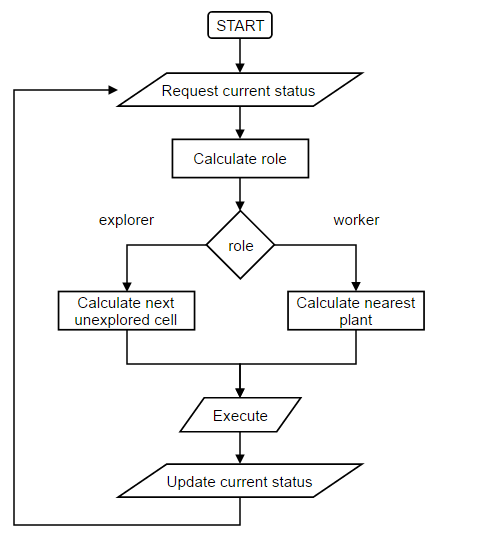
(3)

The probability of each robot to turn into a worker robot, *Pw* depends on their displacement from the mean coordinate of the plants *dplant* and the maximum possible displacement *dmax*. Maximum possible displacement is the diagonal length of the entire map.

(4)

(5)

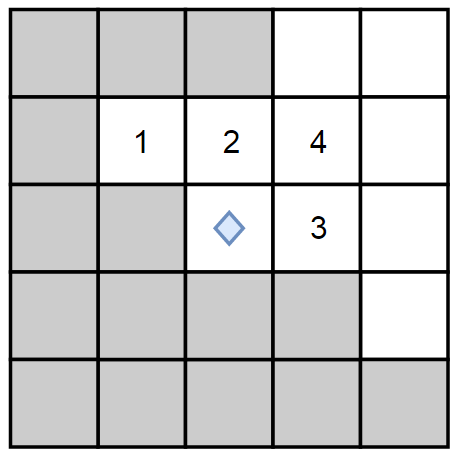
  
Fig. 6. Probability of worker robot selection based on displacement from the mean plant location. In this figure, the blue robot has a higher chance of turning into a worker robot, due to the visibly shorter displacement *a* from the mean plant location than *b*.

  
Fig. 7. Flowchart for the program.

## Exploration Strategy

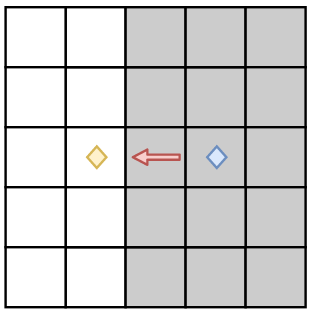
Explorer robot will ideally always move to the next unexplored grid cell while a worker robot will travel the shortest displacement to the next collectable plant and proceeds to collect the loose fruits. The next location can be calculated with a deterministic method.

An explorer robot should always maximize the explored area with minimal distance travelled. In other words, when presented the choice, the robot should always move towards area that is surrounded by the most explored area. This will reduce the need for having to turn back to the similar location. The ideal behavior can be illustrated as follows:

  
Fig. 8. The selection of next move based on the number of explored cells based on a coverage check of 3X3. The blue diamond represents the robot and the grey and white squares represent explored and unexplored cells. Higher

priority will be given to cell number 1 and so on.

In the case where the robot has no more unexplored grid cells left, the robot will utilize the flocking algorithm to move towards the destination of other explorer robots.

  
Fig. 9. The movement (red arrow) of blue robot when no unexplored grid cell is left around itself. The movement will be based on flocking algorithm.

# Result

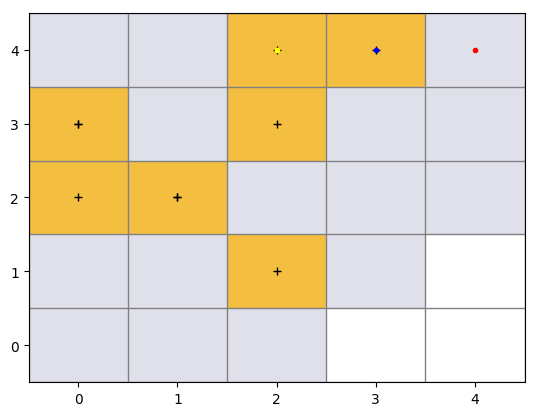
The delegation algorithm is simulated using Python 3 using *matplotlib* library. The grid size used is 5X5. Each simulation has up to 10 plants with randomized location. Completion occurs when all plant locations have been collected from.

The resultant number of iterations until completion, number of plants, and number of explored cells are recorded. The following table shows the result of 10 consecutive trials with 3 units of robots.

TABLE I

Simulation Result

|  |  |  |  |
| --- | --- | --- | --- |
| Trial No. | No. of plants | No. of Iterations | Explored cell (/25) |
| 1 | 9 | 15 | 23 |
| 2 | 7 | 15 | 24 |
| 3 | 7 | 13 | 22 |
| 4 | 9 | 17 | 24 |
| 5 | 7 | 19 | 20 |
| 6 | 8 | 18 | 22 |
| 7 | 6 | 17 | 23 |
| 8 | 7 | 11 | 22 |
| 9 | 8 | 18 | 24 |
| 10 | 9 | 14 | 21 |
| Average | **7.7** | **15.7** | **22.5** |

  
fig. 10. An example of simulated result of the 3rd trial at the 13th iteration. White and grey cells represent unexplored and explored cells. Orange cells represent cells that have been collected. The black crosses in the middle of the cell represent the initially randomized location of plants. The colored dots are the final location of the robots.

# Conclusion

In this paper, a loose fruit collection method is proposed and simulated. This method uses a deterministic algorithm for grid-based exploration, and uses flocking algorithm to reach unexplored areas further away. During the exploration, some robots may be assigned as a worker robot to collect fruits. The delegation of task is based on probability, thus only uses each robot’s local information.

The result has showed that the algorithm achieved an average of 90% explored areas and collected 7.7 plants, with 3 robots under 15.7 iterations.

Our future work will focus on deployment on finer grid as well as optimization on delegation algorithm.

Acknowledgement

We would like express our thanks to Universiti Tunku Abdul Rahman (UTAR) for nurturing and their relentless support towards us.

References

1. Sime Darby. (2014). “Palm Oil Facts and Figures”. [Online]. Available: <http://www.simedarbyplantation.com/upload/Palm-OilFacts-and-Figures.pdf>
2. Muhammad Muzakkir Mohd Nadzri & Afandi Ahmad, “Roller Picker Robot For Loose Fruit Collection System”. (2016). [Online]. Available: http://www.arpnjournals.org/jeas/research\_papers/rp\_2016/jeas\_0716\_4712.pdf
3. Beni G. (2005) From Swarm Intelligence to Swarm Robotics. In: Şahin E., Spears W.M. (eds) Swarm Robotics. SR 2004. Lecture Notes in Computer Science, vol 3342. Springer, Berlin, Heidelberg
4. Barca, J., & Sekercioglu, Y. (2013). Swarm robotics reviewed. *Robotica,* *31*(3), 345-359. doi:10.1017/S026357471200032X
5. Haoxiang Lang, Ying Wang, Clarence W. de Silva. “Mobile Robot Localization and Object Pose Estimation Using Optical Encoder, Vision and Laser Sensors”, *International Conference on Automation and Logistics*. pp. 2, 2008.
6. Craig Reynolds, “Boids”. (2001). [Online]. Available: http://www.red3d.com/cwr/boids/

1. [↑](#footnote-ref-1)