# A Swarm Intelligence Based Algorithm for Distribute Search and Collective Cleanup


Daoyong Liu [1]
School of software
Shanghai Jiao Tong University
Shanghai China
blueskydyliu@sjtu.edu.cn

Xin Zhou [2]
School of software
Shanghai Jiao Tong University
Shanghai China
zhouxin@sjtu.edu.cn

Alei Liang [3]
School of software
Shanghai Jiao Tong University
Shanghai China
liangalei@sjtu.edu.cn

Haibing Guan [4*]
Dept Computer Science
Shanghai Jiao Tong University
Shanghai China
hbguan@sjtu.edu.cn



***Abstract*—A collective cleanup task requires a multi-robot system to search for randomly distributed targets and remove them under a dynamic environment. In traditional methods, robots wandered in subareas (which caused too much repeat search) and interchanged all detected information with their neighbors, so global searching time and communication traffic increased. In this paper, we propose a swarm intelligence based algorithm that minimizes the expected time for searching targets by dividing the environment into two levels subareas then using a dynamic computing subareas' probability algorithm for search strategy, and it can also reduce communication traffic by robots' selective information interactions with their neighbors. A modified Particle Swarm Optimization (PSO) method is used to balance searching and selecting, which helps to allocate reasonable robots to different targets. The simulation results demonstrate the higher efficiency of the proposed method when compared to another method [20].**



***Keywords-multi-robot system; collective cleanup; swarm intelligence; PSO.***


## I. INTRODUCTION

Coordination of multi-robot has received much attention in recent years due to its real-world applications [1-5]. The distributed coordination is desirable for multi-robot systems under a dynamic and unknown environment due to its robustness, flexibility, and reliability. Recently, more and more researchers have been conducted in the bio-inspired coordination method under dynamic and unknown environment [6-7]. One of the key issues for multi-robot system is how to coordinate the behaviors of individual robots so that global performance can be optimized. So the main challenge for multi-robot system is to design an effective algorithm that can adapt their behaviors via local interactions with environment and other robots.

Typical problem domains for the study of swarm intelligence include: box-pushing [1], aggregation [2], pattern formation [3], cooperative mapping [4], sorting[5], collective construction[6], foraging [7], chain-formation[8], and hove-avoidance[9] . All of these problems consist of multi-robots making decision autonomously based on their local interactions with other robots and environment. Each robot's function should be simple.

Yet, one of the major issues for multi-robot systems, especially for large scale multi-robot systems, is its efficiency. It is not a very efficient way to let a suite of robots looking randomly around the room hoping to "get lucky" to find the targets. There needs to be some effective ways to coordinate the multi-robot system, and the multi-robot algorithm should have following characteristics: be distributed among the many robots, be simple computation and have low communication traffic.

In a collective cleanup task, there are a lot of targets randomly distributed in a construction area. The robots should find the targets first, and then remove them. But some targets are too heavy to be removed by one robot, so multiple robots need to work together. So the problem is a dynamic task searching and allocation task. Robot should make decision whether to search new targets or help other robots to remove targets.

In this paper, we propose a swarm intelligence based algorithm to minimize searching time and reduce communication traffic to make the algorithm more effective. In this algorithm, the whole map is divided into a set of distinct sub-areas and each sub-area is divided into some grids. Robot will dynamically compute sub-areas' probability and the cost of traveling to them, then choose the best one to go by it local interactions and information; all robots will not necessary to store whole map's grids' state, and it only need to store subarea's state and robot's current subarea's grids state. Each robot has a buffer, the buffer only store its current subarea's girds' state. Robot will interchange there subarea's state with its neighbors and current subarea's grids' state when two robots are in the same subarea. Each robot has two statuses, searching and selecting. When there is no detected target available, robot's status should be in searching phase. Once a target or multiple targets are detected by its camera sensor, the robot should switch its status to selecting phase. A modified Particle Swarm Optimization (PSO) is used to balance searching and selecting. Since this algorithm is a distributed algorithm, so no robot will get global information and there has no global control unit; each robot will make decision individually by its local information and communication with its neighbors. A WIFI model is used for communication between robots and their neighbors.

The presentation of the paper is organized as follows: Section II introduces related works. Section III describes


*Corresponding Author: hbguan@sjtu.edu.cn

the problem statement. Section IV describes our Swarm Intelligence based algorithm for collective cleanup. Section V presents the simulation results using an embodied robot simulator Player/Stage [22] [23]. Section VI outlines the research conclusion and future work.

## II. RELATED WORK

The area of multi-robot coordination for large groups of robots has been very active in the last few years. One of the first works to deal with the motion control of a large number of agents was proposed by Reynolds in 1987 [10]. He modeled the motion of a flock of birds (called birds) by computer simulation. He believed that the whole of the birds' action is the result of each individual member's following some simple rules. Payton [11] proposed techniques for coordinating the actions of large numbers of small-scale robots to achieve useful large-scale results in surveillance, reconnaissance, hazard detection, and path finding. Virtual pheromones named symbolic message used for robots' communication and coordination. Luiz Chaimowicz [12] proposed interpolated implicit functions to control large group of robots and sensors. Nathan Michael [13] proposed a market-based coordination protocols algorithm which dynamically assigns tasks to multiple agents for multi-robot coordination. In this algorithm, all agents' action depends upon other agents by their local communication. Jelle R. Kok [14] offered solutions to the problem of multi-robot decision making by applying coordination graphs to the continuous domain by assigning roles to the agents and then coordinating the different roles. The author demonstrated this method in the Robot Cup soccer simulation domain to show its efficiency for multi-robot coordination.

PSO-inspired methods have received much attention in recent years. Particle Swarm Optimization algorithm first proposed by James Kennedy and Russell Eberhart [15] in 1995, the algorithm was discovered through simulation of a simplified social model.

Pugh and Martinoli [16] applied an adapted version of Particle Swarm Optimization learning algorithm to distribute unsupervised robotic learning in groups of robots with only local information. They presented a multi-search algorithm inspired by Particle Swarm Optimization. Each robot was represented by a particle agent in a 2D search space, but the algorithm only focus on single target searching, so it is hard to apply this algorithm to multi-target scenarios.

Cecilia and Ricardo [17] presented an application of the Particle Swarm Optimization algorithm to the foraging problem, which consists of an animal (or a group of animals) looking for a resource, typically food. They modified the Particle Swarm Optimization for this scenario called Food Swarm Optimization.

Sheetal Doctor and Ganesh K [18] presented the application of Particle Swarm Optimization for collective search. They divided all robots into some subgroups, where each group was dedicated to only one target searching. Once when all targets were found, a global processing node will identify the most important one.

Y. Meng and J. Gan [19][20] presented a PSO based method to balance the tradeoff of exploration and exploitation for a collective cleanup task using multiple robots. They used a grid based map for targets searching, but all robots should store global map's grids, and it means each robot should have enough large memory. And Y. Meng also presented a Dynamic Programming Equation (DPE) Based method for Multi-Robot searching [21]. To coordinate the behavior of individual robots and consider the interaction between the robots, a DPE based algorithm was proposed to estimate the utility function of each robot so that global performance can be achieved. But the algorithm needs complex computation.

In our works, we use a modified Particle Swarm Optimization (PSO) method to balance searching and selecting. A WIFI model is used for communication. Robot will switches between two statuses by its' local interactions with other robots and environment.

## III. PROBLEM STATEMENT

In a collective cleanup task, there are a lot of targets randomly distributed in a construction area. The robots should find the targets first, and then remove them. But some targets are too heavy to be removed by one robot, so multiple robots need to work together. In this paper, a swarm intelligence based algorithm was proposed to search the targets and remove them. The finish time of the collective cleanup task is all targets has been found and removed. Though this is a NP hard problem, the object of this study is to develop a distribute algorithm for multi-robot system so that multi-robot system can finish the task as quickly as possible.

There are some assumptions for this algorithm. (1) The outer bounder of the grid-based map is given. (2) The map size and total number of targets is given. (3) Each robot's localization is accurate. (4) Each robot can only communicate with its neighbors in a predefined range.

## IV. SWARM INTELLIGENCE BASED DISTRIBUTED COORDINATION ALGORITHM

Initially, all robots are dispersed randomly in the map. All robots' statuses are set to searching phase first. Since their randomly dispersed, it will be an effective way to let all robots search their subarea first. When a subarea has been detected, robot will make decision individually to move to another subarea or not based its local information and interactions with its neighbors and environment. When robot is in searching phase, it is an effective way to let robot detect undetected areas as much as possible. Each robot will update its local information by its detected area and its neighbor's information. Once a robot finds a target by itself or receives one or more targets message from its neighbors, robot should make decision to balance tradeoff of searching and selecting. If there is only one target has been found, robot should make decision to search new area or go to remove the target. If there has more than one target, robot not only makes decision to search new area or go to remove a target but also makes decision to select which target to be removed.

### A. Grid based Map

We assume that the multi-robot system moves in an $N\times M$ grid-based map. The map is divided into $m\times n$ sub-areas; we use $S_{ij}$ to record the subarea's state and each sub-area consists of equal number of grids; we use $G_{kt}$ to record a subarea's grids' state, and assume grids number in a subarea is s×s.

$$S_{ij} = \begin{cases} 1, \textit{subarea ( i, j) has been detected} \\ 0, \textit{subarea ( i, j) has not been detected} \end{cases} \quad (4\text{-}1)$$

$$G_{kt} = \begin{cases} 1, \textit{grid ( u ,z) has been detected} \\ 0, \textit{grid ( u, z) has not been detected} \end{cases} \quad (4\text{-}2)$$

Where $i\in(0,m), j\in(0,n), u\in(0,s), u\in(0,s)$.

$S_{ij}$ represents subarea value at (i,j),where i and j represent the row and column number. $G_{kt}$ represents a grid value at (k,t) in a subarea, where u and t represent the row and column number of a subarea. Each robot has a buffer to store current subarea's grids' state, when a subarea's detected grids arrive at some percentage, this subarea's state will be set to detected, and then robot will select another subarea to search. The buffer will be used to store a new subarea's grids' state when robot is in another subarea.

### B. Cooperative Method

**Search Algorithm:** During the searching phase, no target has been detected by any robot yet. It is an effective way to let robot explore unexplored areas as much as possible. When a subarea's detected grids' percentage arrives at a predefined value (less than 1), this subarea's state should be set to be detected in current robot memory. Robot selects another undetected subarea to detect based following factors of those undetected subareas: (1) How many potential robots which are closer to the subarea than its local subarea. (2) The travel cost to the subarea.

To reduce the computational complexity, the map is divided into m×n sub-areas, and total number of subarea is r, each subarea has H (H=s× s) grids. Robot can count how many undetected grids in current subarea based on its local information, which can be represented by $G_\alpha^k(t)$. Where α is the subarea index. The utility function of a robot for subarea α can be defined as:

$$P_\alpha^k(t) = B_\alpha^k(t) \times \frac{G_\alpha^k(t)}{D_\alpha^k(t)} \quad (4\text{-}3)$$

Where $G_\alpha^k(t)$ is the number of unexplored grids within subarea α. $D_\alpha^k(t)$ is the distance between robot k and center of subarea α, and $B_\alpha^k(t)$ is the probability of having a target in subarea α. $B_\alpha^k(t)$ is defined as:

$$B_\alpha^k(t) = \frac{D_\alpha}{\sum_{i=1}^{r} D_i} \quad (4\text{-}4)$$

Where $D_\alpha$ is the distance between subarea α and map center.

When robot is in a subarea, it will detect its current subarea until detected percentage arrives at a predefined value. Some other algorithms let robot wander in subarea [19][20]. In this case, robot detected may be repeated over the girds which have been detected. In this paper, we give each grid a utility function $U_{st}^{ki}(t)$, and robot will detect grid first which has high utility. Base on our observation in the simulation results, we find that it will be more effective to let robot detect a subarea corner first, so we defined the utility function as:

$$U_{uz}^{ki}(t) = D_{uz}^{ki}(t) \quad (4\text{-}5)$$

Where $D_{st}^{ki}(t)$ represents the distance of grid (u,z) in subarea i and subarea i's center . Robot will remember which girds have been detected in its buffer and not select that kind of grids to move to. This will minimize the expected time for searching time, so global performance will be improved.

**Local Communication:** Each robot uses a WIFI model to communicate with other robots. Each robot makes decision by its local information. When two robots are in a predefined communication range, they will interchange local information. Two robots will interchange all global map information in traditional methods [20]. To reduce communication traffic, each robot has a grid buffer and the buffer only save robot current subarea's grids' state. When two robots are in a predefined communication range, if they are in the same subarea, the will interchange their global subarea's state and current subarea's grids' state. If they are not in the same subarea, they just only need to interchange their subarea's state. To accelerate information interchange, all subareas and their grids' state are stored in a bit data structure.

**A Modified PSO Based Method for Balance:** When a robot finds a target by itself or receives one or more targets message from its neighbors, robot should switch to selecting phase, so the robot have to decide to search a new area or go to remove the target together with other robots. For each target, we define is utility function as follows:

$$U_{ij}^k(t) = \frac{W_{ij}(t)}{D_{ij}^k(t)} \quad (4\text{-}6)$$

Where $W_{ij}(t)$ is the weight of the target at location (i,j), $D_{ij}^k(t)$ is the distance between robot and the target.

In general, the higher the utility of the target, the more attractive the corresponding target is to the robot. Therefore, the benefit of moving to this target would be higher. If more robots moving to this target (the target may not needs so many robots), which lead the less available robots for searching in the future, so we should prevent this scenario. We tend our attention to Particle Swarm

Optimization (PSO) algorithm [15] to balance the searching and selecting.

In PSO, each particle moves in the map and updates its velocity according to the best previous position and global best position achieved by its neighbors. The velocity vector of a particle can be defined and updated as followings:

$$\begin{cases} V_{id}(t+1) = W \times V_{id}(t) \\ \qquad + C_1 \times rand() \times \left(P_{id}(t) - X_{id}(t)\right) \\ \qquad + C_2 \times rand() \times \left(P_{gd}(t) - X_{id}(t)\right) \\ X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \\ 1 \le i \le N,\ 1 \le d \le D \end{cases} \tag{4-7}$$

Where $v_{id}(t)$ is the velocity of robot i on the dth dimension, $p_{id}(t)$ is the local optimal position, $p_{gd}(t)$is the global optimal position, c1 and c2 are the accelerate factors, w is the inertia factor, $x_{id}(t)$ is the position of robot i on the dth dimension. Since this is a distribute algorithm, robots make decision only by local interactions, so it would be difficult to select global best. Directly using PSO-based strategy would be difficult. A modified PSO method is proposed to adapt this situation. The velocity vector of a robot can be updated like this:

$$V^k(t+1) = W \times V^k(t) + C_1 \times rand() \times (P_l^k(t) - X^k(t)) \tag{4-8}$$

Where W represents factor balance the random offset within certain grids. $P_l^k(t) = \max U_{ij}^k(t)$ represents the global best target position the neighboring robot have found. Each robot will use (4-8) to update it velocity when in selecting phase.

## V. SIMULATION

To evaluate the performance of the swarm intelligence based cooperative algorithm and take the real world robot constraints into considerations, we select Player/Stage as our embodied robot simulator. Player/Stage is a famous embodied simulator developed by University of Southern California [22] [23]. At last we choose another method to compare with our method.

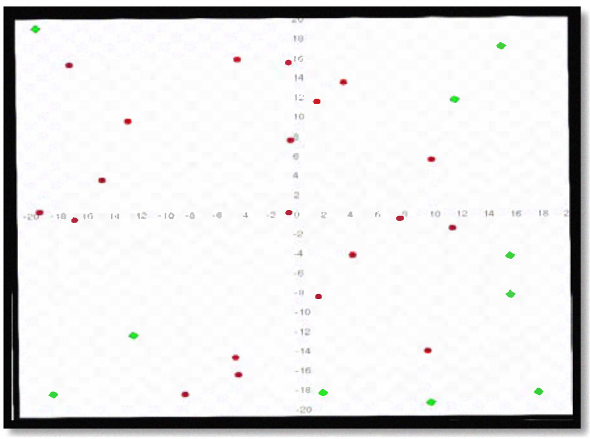

Fig.1. Initial state of the robot distribution with 20 robots and 10 targets.

The simulator interface is shown in Fig.1, map size is 40m×40 m, 1600 grids in all, and we divide the map into 16 subareas. Each subarea includes 100 grids. Several targets are distributed randomly with green color. We use some homogeneous Pioneer 3DX robots. They are distributed randomly with red color in the simulator. Where each robot is equipped with a camera system to identify the targets, a laser to avoid obstacles and a wireless communication card called WIFI for communication.

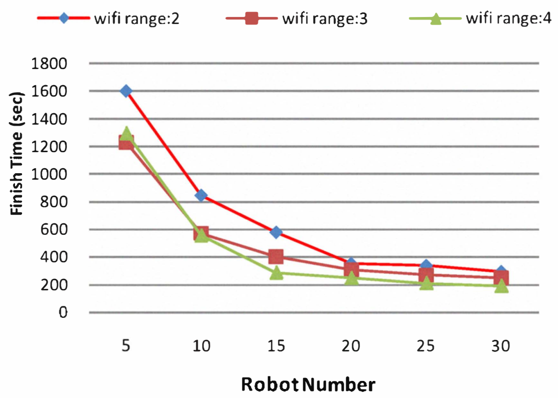


Fig.2. Different number of robots and communication range for collective cleanup. Robot number is from 5 to 30, communication range is from 2 to 4 meters.

From the Fig 2, we can find out when the number of robots increased from 5 to 30, global finish time is decreased accordingly. When we increase WIFI communication range from 2 meters to 4 meters, we can also find out that a greater communication range can improve global performance.

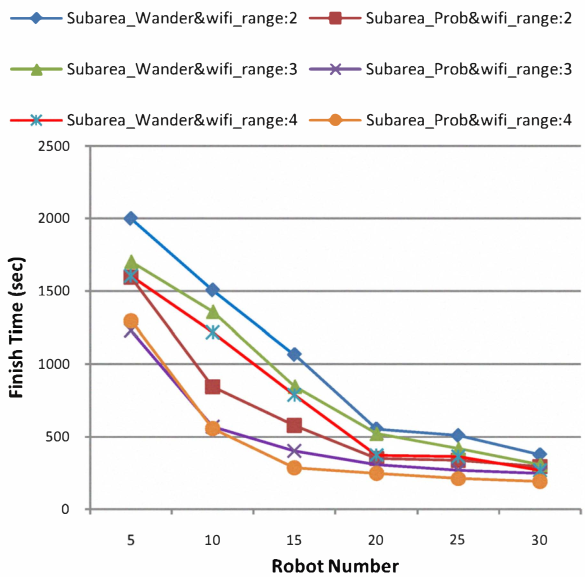


Fig.3. Compared with a Subarea_Wander algorithm with different robots and communication range. Robot number is from 5 to 30, communication range is from 2 to 4 meters.

In Fig3 we take our algorithm compared with another method [20]. Since this method lets robots wander in a subarea (which caused too much repeat search, so global finish time increased), we call it Subarea_Wander algorithm. We named our method Subarea_Prob algorithm because it uses a probability based algorithm to search when robot in subarea. Communication range is from 2 meters to 4 meters, robots number is from 5 to 30. The task finished time represents the overall performance. The less finish time, the faster robot detects and removes the target, and also means less power consumption in all. It is obvious that Subarea_Prob algorithm outperforms the Subarea_Wander algorithm in both finish time and overall power consumption.

## VI. CONCLUSION AND FUTUREWORK

A swarm intelligence based algorithm is proposed for a distribute search and collective cleanup task. Our aim is to reduce overall finish time and communication traffic to make the system high efficiency. Compared with [20], the simulation results demonstrate the higher performance of our algorithm. In our Subarea_Prob algorithm, each robot don't need to store global map grids' state, it only need to store subareas' state and robot current subarea's grids' state in the buffer. Robot only needs to interchange subareas' state and current subarea's grids' state when two robots are in the same subarea. It will cause less information interchange between robots, so global communication traffic can be reduced.

Future work will include new improvements to accelerate searching time and more consideration about real-world environmental elements, and we will apply our algorithm to our hardware platform to evaluate our algorithm.

## ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant No. 60773093、60970107、 60970108), The Science and Technology Commission of Shanghai Municipality (09510701600).

## REFERENCES

[1] J.Mataric,M.Nilsson, and K. T. Simsarian,"Cooperative Mult-robot Box-Pushing IEEE/RSJ International Conference on Intelligenct Robots and Systems,1995.

[2] A. Martinoli, A. J. Ijspeert, and F. Mondada, "Understanding Collective Aggregation Mechanisms: From Probalilistic Modeling to Experiments with Real Robots", Robotics and Autonomous Systems, Vol,29,pp. 51-63, 1999.

[3] T. Balch and R. C. Arkin, "Behavior-based Formation Control for Multi-robot Teams", IEEE Trans.on Robotics and Autonomous Systems, 1999.

[4] B. Yamauchi, "Decentralized Coordination for Multi-robot Exploration", Robotics and Autonomous Systems, Vol.29,No.1, pp.111-118, 1999.

[5] T. Weigel,J.s.Gutmann, m.Dietl, A. Kleiner, and B. Nebel, "CS Freiburg: Coordinating Robots for Successful Soccer Playing", Special Issue on Advances in Multi-Robot Systems, T. Arai, E.Pagello, and L. E. Parker, Editors, IEEE Trans. On Robotics and Automation, Vol. 18, No.5 pp. 685-699,2002.

[6] R. L. Stewart and R. A. Russell. "A Distributed Feedback Mechanism to Regulate Wall Construction by a Robotic Swarm". Adaptive Behavior. 14(1):21-51, 2006.

[7] M.J.B. Krieger, J. B. Billeter, and L. Keller, "Ant-like Task Allocation and Recruitment in Cooperative Robots," Nature, Vol. 406, 2000, pp.992-995

[8] S. Nouyan, M. Dorigo, "Chain Formation in a Swarm of Robots', Technical Report TR/IRIDIA/2004-18, IRIDIA – University libre de Bruxelles, Belgium, Dea Thesis, September 2004.

[9] V. Trianni, S. Nolfi, M. Dorigo, "Cooperative Hole Avoidance in a Swarm-bot" , Robotics and Autonomous Systems, Volume 54, number 2, pp. 97-103, 2005.

[10] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", Computer Graphics, 21(4), July 1987, pp. 25-34.

[11] David Payton, Mike Daily, Pheromone Robotics, Autonomous Robots 11, 319–324, 2001

[12] Luiz Chaimowicz, Nathan Michael, "Controlling Swarms of Robots Using Interpolated Implicit Functions", International Conference on Robotics and Automation, Barcelona, Spain April 2005.

[13] Nathan ichael, Michael M. Zavlanos , "Distributed Multi-Robot Task Assignment and Formation Control", IEEE International Conference on Robotics and Attomation Pasadena, CA, USA , May 19-23, 2008.

[14] Kok, J.R., Spaan, M.T.J., and Vlassis, N., "Multi-robot decision making using coordination graphs" In Preceedings of the International Conference on Adavanced Robotics (ICAR), pp. 1124-1129, Coimbra, Portugal, June 2003.

[15] Kennedy J, Eberhart R.C.: Particle Swarm Optimization, IEEE International Conference on Neural Networks. 1995.

[16] J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization", IEEE Swarm Intelligence Symposium, 1-5 April 2007, pp.332-339.

[17] Di Chio, C., Poli, R., & Di Chio, P."Extending the Particle Swarm Algorithm to Model Animal Foraging Behaviour", International Workshop on Ant Colony Optimization and Swarm Intelligence, Brussels, Belgium, September 4-7, pp. 524-515.2006

[18] S. Doctor, G. Venavagamoorthy, and V. Gudise, "Optimal PSO for Collective Robotic Search Applications", IEEE Congress on Evolutionary Computation, Portland, OR, pp. 1390 – 1395, June 2004

[19] Y. Meng and J. Gan, LIVS: Local Interaction via Virtual Stigmergy Coordination in Distributed Search and Collective Cleanup, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2007), Oct. 29 – Nov. 2, San Diego, CA, USA.

[20] Yan Meng, Jing Gan: A distributed swarm intelligence based algorithm for a cooperative multi-robot construction task. Swarm Intelligence Symposium, 21-23 Sept. 2008:1 – 6

[21] Y. Meng, Dynamic Programmin Equation Based Multi-Robot Searchin, nternational Journal of Factory Automation, Robotics and Soft Computing, Issue 2, 2008. pp. 7-14

[22] Vaughan, R.T. "Stage: a multiple robot simulator." Institute for Robotics and Intelligent system, School of Engineering, University of Southern California, Tech. Rep. IRIS-00-394, 2000

[23] Gerkey B.P., Vaughan R.T., Sty K., Howard A., Sukhatme G.S., Mataric M.J., "Most valuable player: a robot device server for distributed control." In proceeding of the IEEE /RSJ International Conference on Intelligent Robots and Systems(IROS2001), PP.1226-1231. Wailea,Hawaii. October 29, 2001