Social Network Analysis for Automatic Target Recognition in Swarm Robotics

Mahsa Maghami
Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL, USA
mmaghami@cs.ucf.edu

Michael C. Koval
Electrical Engineering and Computer Science
Rutgers University
New Brunswick, NJ, USA
mkoval@eden.rutgers.edu

Michael Georgiopoulos Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL, USA
michaelg@mail.ucf.edu

Adina E. Rubinoff
Computer Science and Linguistics
University of Rochester
Rochester, NY, USA
a.e.rubinoff@rochester.edu

***Abstract*— Over the past few years, swarm-based systems have emerged as an attractive and promising approach for implementing distributed autonomous systems. This is useful in different applications, such as automatic target recognition (ATR). In ATR, the most pressing concern is the accuracy of the system in detecting and recognizing the targets. To accurately identify targets, prior works require more than one agent to classify a target, which decreases the number of recognition errors. Increasing the number of agents required to recognize a task makes the system more accurate at the cost of delaying, which is a drawback in many applications.**

The NetBots algorithm is a novel approach that improves accuracy in ATR by introducing a social network to the swarm system. This social network keeps track of the robots' target detection histories and is used to estimate the robots' accuracy. Any time that two robots deciding on a target agree with the majority decision, a link is formed between them. The PageRank algorithm is then used to rank the robots based on the number and ranks of the robots linking to them. Robots that are more accurate will agree with the majority more often, so they will have more links and therefore attain a higher rank. With NetBots,when deciding on a task, higher ranked robots have more influence in the decision made and consequently inaccurate robots' votes will not negatively impact the majority decision. As our experiments demonstrate, this approach increases the overall accuracy of the system, compared to approaches that do not take advantage of the information provided by the NetBots' social network.

I. INTRODUCTION

Swarm robotics, inspired by swarming animals such as ants, birds, and fish, is the study of how large numbers of simple agents can be designed such that a desired collective behavior emerges from their interactions [1]. The main characteristics of the swarm approach to control a group of robots are: 1) robustness: system will continue to operate even if one or more robots are damaged or destroyed; 2) scalability:system can operate under a wide range of robot group sizes and task sizes without substantial decrease in performance; 3) flexibility:robots are able to adapt, if the task changes, while they are working on it, and will attempt to complete the modified task[1].

Because of the aforementioned desirable characteristics, swarm-based systems have emerged as an attractive and promising approach for implementing distributed autonomous systems, such as automatic target recognition (ATR).

In the ATR problem, a collection of targets is scattered through an environment and robots must first locate the targets in the environment, and then classify these targets into different categories based on certain characteristics. In previous work, this goal was achieved by unmanned aerial vehicles (UAVs) in direct communication with a central control station via wireless communication. With the improvement of technology, these UAVs can be replaced by simpler mini-UAVs each of which is capable of performing limited computations to assist in recognizing the targets but are unable to perform the recognition task all by themselves; as a result they constitute a swarm based system which through the coordinated cooperation of some of a ceratin number of swarm units they can now accomplish the target recognition task, alleviating the need for a central control station that is powerful enough to accomplish the recognition task on its own[4]. Mini-UAVs have many characteristics that are desirable in a swarm system: decreased inter-agent communication, removal of a central station, increased processing efficiency, and superior robustness[5]. Despite these advantages, mini-UAV's limited computing power increases the probability of UAVs incorrectly identifying targets [5]. To overcome this problem, often, multiple robots' opinions are required to definitively classify a task. Therefore, it is not surprising that in prior work a number of examples of multi-agent ATR systems, where cooperation of multiple robots is used to confirm the identity of a target, can be found ([4],[5],and [6]). In a situation where the robots are homogeneous, have identical physical and algorithmic capabilities, the aforementioned approaches work well. When heterogeneous robots are considered, however, a new approach is needed.

Heterogeneous robots are commonly found in systems where robots learn from prior experiences, or the robots' sensors have significantly varying manufacturer error tolerances. In these cases, inaccurate robots negatively impact the overall accuracy of the system, unless the system is designed to take the heterogeneity into account.

This paper deals with the issue of heterogeneous robotic in a swarm robotic system to alleviate the negative effect of ineffective robots. One of the most important contributions of the paper is the utilization of a social network that the swarm forms, which takes advantage of the history of the robots' interaction within the swarm to improve the effectiveness and efficiency of the swarm in accomplishing the required tasks. What is worth pointing out is that in this social network a robot relies on historical information obtained through local communication with robots in its vicinity only.

To keep the history of robot's behavior and address the reputation of agents in the system, we introduce a novel algorithm named NetBots. When any two robots work together on a task and are both part of the majority decision, a link is formed between them. Gradually the connectivity graph of robots grows as they cooperate to identify targets. A distributed ranking algorithm [12] is used to calculate a local rank for each robot based on the links in its local connectivity graph. Robots with higher ranks are the ones with higher reputation in the system and they exert more influence when decisions are made about the recognition of targets.

Netbots has been simulated in a Player/Stage simulator environment with a team of Khepera III robots and it can be applied to any group of mini-UAVs. Simulation results show that the NetBots algorithm successfully ranks robots based by their ability to correctly identify targets and improves the recognition accuracy of the system.

The organization of the paper is as follows: In Section II some related work is discussed. Section III describes in detail the general problem and a previously devised algorithm ([7]), called HelpBots that is used as a measure of comparison with the newly introduced NetBots algorithm. In Sections IVand V the NetBots is presented and associated implementation issues are discussed. Section VI compares experimentally the Helpbots and Netbots results and provides appropriate observations. Finally, Section VII is a summary of the paper, accompanied with a few pointed remarks.

II. RELATED WORK

In past years there were lots of attention to automatic target recognition (ATR) as it can be helpful in many military applications. Target recognition systems could be used in bad weather to help guide pilots who are unable to see the ground [8], could be helpful to soldiers that need to see and identify objects in the dark [9], or could be applied in unmanned ground vehicles to detect their surroundings and avoid obstacles [10].

As the idea of swarm robotics grew recently, lots of works tried to implement an ATR system in a distributed way by using simple agents in large number. In [4] and [5] a swarm based system of UAVs are used to efficiently detect targets in the environment. They focused on using the limited resources of UAVs to reach to an efficient decision. Miller et al in [6], used the similar system to address the task allocation problem in between UAVs. They presented some heuristic methods to achieve this goal by using UAVs equipped with Global Positioning system (GPS). Although in this work, we used the concept of confirmation of several robots to identify a target, all robots have local communication and they have no global information about the system. This reduction in amount of communication can help the UAVs to preserve the resources more and it reduces the complication of robots.

TABLE I
THE PARAMETERS IN GENERAL ATR PROBLEM

Parameter	Description
N	Number of Robots
Q	Number of robots required to detect a target successfuly
W	Waiting time

In addition, in this paper we used the concept of ranking webpages in internet by search engines, into the social network formed among robots. To rank the robots in their society, NetBots uses the PageRank algorithm locally. PageRank, first discussed in [11], treats the internet as an unweighed, directed graph in which webpages are vertices and links between pages are edges. PageRank assigns each vertex a scalar rank that can be viewed as a measure of that vertex's "importance" relative to other vertices in the graph. More concretely, a vertex's PageRank is defined as the probability of a web-surfer reaching a vertex after performing a *random walk* of the graph from a randomized starting vertex. There are lots of variation of this algorithm in the literature. For a review of some of these algorithms, see [13]. As in NetBots, robots have no global information about the system, a distributed PageRank algorithm is applied [12].

III. HELPBOTS ALGORITHM

This section describes the base algorithm, called HelpBots which is used as a benchmark algorithm against which the NetBots performance is assessed. This algorithm will be referred to as HelpBots algorithm, since its most important aspect is the robots' call for help when waiting at a task, similar to [7].

A. Problem Formulation

The generic target recognition experiment is as follows: N robots placed in an arena and their goal is to detect T targets. The robots wander around the arena looking for targets and avoiding obstacles. Once they arrive at the target they attempt to recognize it. A certain number of robots (Q) is required to confirm the identity of targets. Every robot which arrives at the target waits for a waiting time period (W) until Q robots are present at the target. Once Q robots are present at a target, the target will be labelled by the majority voting of robots;opinion. If the robots are not at a target, waiting for other robots to recognize it, they wander around the arena to find and recognize other targets, until the experiment ends. The parameters of this algorithm are listed in Table I.

Ijspeert et al [7] performed a similar experiment to solve the stick-pulling problem with swarm robotics. In his work, in addition to simply wandering, the robots would respond to calls for help from other robots working on tasks. When a robot was working on a task it would send out a call for help. Any wandering robot that received that call would turn and head towards the task in order to help out. This additional communication between robots increased the speed with which the robots completed all the tasks. We used this type of communication in our HelpBots algorithm as well.

B. Helpbots Algorithm Overview

In the HelpBots algorithm, once the robot has successfully reached a task, it determines if it must adopt the *host* or *helper* role using Algorithm. If there are no robots waiting at a task, the first robot to arrive automatically becomes that task's host. Hosts are responsible for broadcasting help calls and tallying the helpers' opinions into a majority decision. The host broadcasts the call for help to all robots within a fixed radius, regardless of the relative orientation of the host and recipient. The robot being called computes the relative angle between their current heading and the broadcasting host using the information encoded in the help message. This desired heading is then passed into the navigation system, which guides the called robot towards the help signal while continuing to avoid collisions with other tasks and obstacles.

Furthermore, robots that arrive at an already occupied task, by receiving a help signal,called helpers, promptly send the host a message containing their classification of the task and wait for the appropriate number of robots (Q) to arrive. The detailed Helpbots algorithm is designated as Algorithm 1 and presented below.

```
input: Q, the number of robots required to complete a
       task
while ∃ tasks do
    Wander();
    if task is found then
        DecideRole();
        if host then
            while numRobots ≤ Q do
                CallForHelp();
            end
            CompleteTask();
        else
            // Must be a helper
            while ¬task.completed do
                Wait();
            end
        end
    end
end
```

Algorithm 1: HelpBots Algorithm

IV. NETBOTS ALGORITHM

An overview of the NetBots algorithm is presented as a finite state machine in Figure 1. Most of the *Search* and *Task Completion* states remain unchanged from HelpBots, described in detail in section III. NetBots primarily distinguishes itself from the HelpBots algorithm in the *Ranking* state, as depicted in the simplified finite state machine in Figure 1.

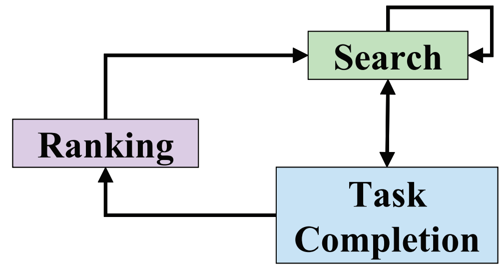

Fig. 1. Overview of the NetBots algorithm as a FSM

A. Network Forming

Each robot (R_i) using NetBots algorithm stores an undirected graph (G_i) that represents all of the interactions that the robot is aware of. Vertices in this graph represent robots and edges represent successful interactions between the joined robots. These *local graphs* may differ from robot to robot as each robot has a different record of past interactions. Generating and updating these graphs occurs only when a task is completed and is the responsibility of the host to do so.

When a helper first arrives at a task, it transmits its opinion of the target and its local graph to the host. After the host receives enough opinions ($Q-1$) to calculate the group decision about the task, it merges all of the robots' graphs into a single graph that represents the combined knowledge of the group so far. To simplify this merge operation, NetBots considers the adjacency matrix representations of each graph.

Consider robots $R_1, R_2, \ldots, R_Q$ immediately after completing a task. Let $G^{(R_k)}$ be the adjacency matrix representation of the local graph of robot R_k such that $G_{ij}^{(R_k)}$ is the number of links between robots R_i and R_j, known to robot R_k. The merged adjacency matrix, G' is defined as

$$G'_{ij} = \max(G_{ij}^{(R_1)}, G_{ij}^{(R_2)}, \ldots, G_{ij}^{(R_Q)}) \tag{1}$$

Unlike a simple matrix addition, performing an element-wise maximum eliminates the possibility of double-counting earlier edges, a crippling problem which would eventually lead to earlier interactions being more influential than recent interactions.

B. Ranking

Once the host has an up-to-date graph, it executes the PageRank algorithm to compute a collective PageRank vector (γ') for the merged graph. The original algorithm of PageRank is presented in algorithm 7. Each robot's rank is added to a tally associated with its decision, effectively weighting the influence of each robot's decision by its PageRank. Let E be the set of robots that decided that the target is blue and P be the set of robots that decided it as green, such that

$$w_p = \sum_{i \in P} \gamma_i \tag{2}$$

$$w_e = \sum_{i \in E} \gamma_i \tag{3}$$

where the task is identified as blue if $w_e \geq w_p$ and green if $w_p < w_e$.

Once this decision is complete, the host forms links between all robots that agree with the group decision and it send back the updated matrix to other helpers. This ensures that robots with the most accurate sensors will rapidly gain links and will become most influential in the calculation of subsequent PageRank vectors.

Therefore, the more correct decisions a robot makes, the higher its rank will be and a higher reputation it will have in its society.

input : Connectivity graph G, initial PageRank vector $x^{(0)}$, and uniform vector v
output: A matrix P s.t. $P_{ji} = \frac{1}{deg(j)}$

```
while δ ≥ ε do
    x^(k+1) = cP^T x^(k);
    w = ||x^(k)||_1 − ||x^(k+1)||_1;
    x^(k+1) = x^(k+1) + wv;
    δ = ||x^(k+1) − x^(k)||;
    return x^(k+1);
end
```

Algorithm 2: The PageRank algorithm

C. Evaluation Metrics

In order to ensure NetBots' social network is working as desired, two criteria must be met. First, each robot's local PageRank vector must converge to the global PageRank vector. In addition, the order of the robots when sorted by the ranks in the global PageRank vector must converge to the actual order of robots when arranged by sensor error. As in [12] the Kendall's τ-Distance metric [14] is used to compare two ordered vectors.

Consider two ordered vectors of robots as p_1 and p_2 with length N. Define the function $K_{\{i,j\}}(p_1, p_2)$ such that:

$$K_{\{i,j\}}(p_1, p_2) = \begin{cases} 1 & : R_i \text{ and } R_j \text{ in different orders} \\ 0 & : R_i \text{ and } R_j \text{ in same order} \end{cases} \tag{4}$$

Using this mathematical definition of the pairwise order of two vector of robots, the number of pairwise differences in order is intuitively defined as:

$$K(p_1, p_2) = \sum_{(R_i, R_j) \in P_D} K_{\{R_i, R_j\}}(p_1, p_2) \tag{5}$$

where $P_D = \{(R_i, R_j) : R_i \in p_1 \text{ and } R_j \in p_2\}$ is the set of all pair-wise combinations of the robots of p_1 and p_2. Finally, Kendall's τ-Distance will be:

$$\text{KDist}(p_1, p_2) = \frac{K(p_1, p_2)}{N(N-1)/2} \tag{6}$$

The flexibility of this metric also allows for comparison between the ranks computed by the NetBots (p_g) algorithm and the actual ranks of the robots (p_{act}, based upon their true sensor error). Low values of KDist(p_g, p_{act}) demonstrate a close correlation between NetBots rankings and actual robot accuracy.

TABLE II
TECHNICAL SPECIFICATIONS OF K-TEAM'S KHEPERA III ROBOT [17].

Parameter	**Value**
Radius	130 mm
Height	70 mm
Weight	690 g
IR Transducers	9
Communication	Bluetooth, WiFi *add-on*

V. IMPLEMENTATION

In order to demonstrate the effectiveness and efficiency of NetBots algorithm, Khepera III robots are simulated in Stage simulator [16]. Technical specifications of Khepera III is listed in Table II [17]. To navigate in the environment, avoid collision, and locate the target, robots use their IR sensors . In this simulation 2 types of targets are scattered in an obstacle free environment and they are colored differently as blue and green and robots use their built-in cameras to recognize the color of targets. Despite the abundance of sensors on the Khepera , including unused ultrasonic rangefinders [17], none of the sensors are well-suited for high-fidelity target identification. We have decided to use the *KoreUSBCam* extension turret designed by K-TEAM Corporation, the creator of the Khepera III robot. Unlike the alternative *KheperaIII Wireless Camera*, the KoreUSBCam is low-resolution (from 160×120 to 640×480 pixels) that transmits data directly to the Khepera III's controller without passing through an intermediate host computer [17].

A. Robots' Communication

The NetBots task identification algorithm, similar to Dasgupta's UAV algorithm [5] requires local communication between robots to transmit the result of an identification vote. Unlike Dasgupta's voting algorithm, NetBots' helpers must also transmit a graph to the task's host, and the host must transmit back a vector of computed ranks and a graph. This type of information is targeted at a single robot, the host, and must be reliably delivered.

This demand for targeted and reliable communication is best achieved with a message passing algorithm [18]. Khepera III robots support two of the most popular physical incarnations of wireless message passing: Bluetooth and 802.11 WiFi (via an optional add-on WiFi card) [17]. While both technologies have been extensively used in prior works [18],[19], we have decided to use WiFi based upon its superior range and K-TEAM's online recommendation:

> The Khepera III is also able to include Wireless Ethernet network communication. This configuration is a perfect solution for applications requiring communication between two robots. [17]

TABLE III

VALUES USED FOR PARAMETERS IN ALL EXPERIMENTS.

Parameter	Value
Q	3
N	6,8,10,12,14
Waiting Time	20,30,40,50,75,100,125,150,175,200(s)
Number of Tasks	9,30
Task distribution in environment	Uniform

VI. RESULTS

A. Experimental Setup

To test the NetBots algorithm, tests were run on both HelpBots and NetBots. In these experiments, the robot models we used in Stage are 1.35m in diameter, a factor of ten larger than real-life Khepera III robots, the targets were represented as colored circles 1.7m in diameter, and the arena is 32m by 20m. Within the arena, the tasks and robots were evenly distributed using a uniform distribution. Table III shows the selected values of all parameters in experiments.

B. Waiting Time Effect

One the important factors in this system is the waiting time parameter. To show the effect of this parameter on the task accomplishment of the system, we ran only the HelpBots algorithm with 10 tasks in the environment. In navigation algorithm both NetBots and HelpBots are the same, therefore successfully completion of task is not dependant on the social network concept. Figure 2 shows the rate of success plotted against the waiting time. Success Rate (SR)is calculated as:

$$SR = \frac{SuccessfulRuns}{TotalRuns} \tag{7}$$

where an unsuccessful run is one that has to be stopped before all the tasks are completed. The lower the bar, the fewer times the robots succeeded in completing all the tasks. As is shown in figure, for the higher number of robots, the success rate is mostly 100%. When there are many robots, they will be able to complete all the tasks no matter how long they have to wait around for helpers, because there will always be robots available to help them. However, the lower numbers of robots are not always successful, because it takes them longer to complete all the tasks. Ten minutes may not be enough time. Also, for the lower numbers of robots that do have below 100% success rates, it is interesting to note that the success rate peaks at a different waiting time for each of them. For example, with eight robots, the best success rate (of 100%) is somewhere between the 50 and 100 second waiting times (probably around 75 seconds), while for 6 robots the peak is at 50 seconds. This shows that lower waiting times are better for smaller amounts of robots. The more robots there are, the longer they can afford to wait for a helper, because there's a higher chance that a robot will be free and come along to help. This also shows that higher waiting times aren't always better - after a certain point, increasing the waiting time reduces the success rate.

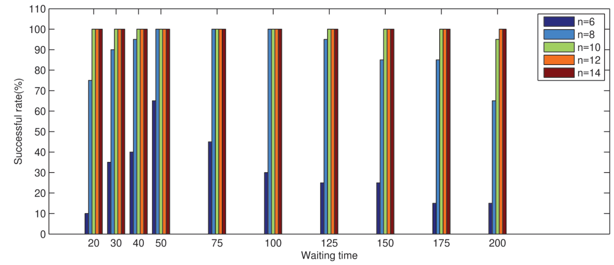

Fig. 2. HelpBots results: Success rate (SR) plotted against waiting time. Data is shown for the six different numbers of robots (n) tested.

C. Accuracy Rate

Figure 3 plots the average number of correctly identified targets against waiting time. Each graph shows data for a different number of robots. Each graph also shows data from the HelpBots and the NetBots algorithms side by side.

In the graph with n equal to six, the NetBots algorithm consistently identifies more targets correctly than the HelpBots algorithm. This is the expected result; after the robots' PageRank vectors converge to the actual error vector, the robots successfully distinguish between accurate and inaccurate robots. When n is greater than six, however, there are too few interactions for the PageRank vectors to converge before all the tasks are completed. If we ran these trials with more tasks, the NetBots algorithm would perform better than HelpBots for larger numbers of robots. With thirty tasks, however, the graphs show that NetBots is on average as accurate as HelpBots for numbers of robots greater than six.

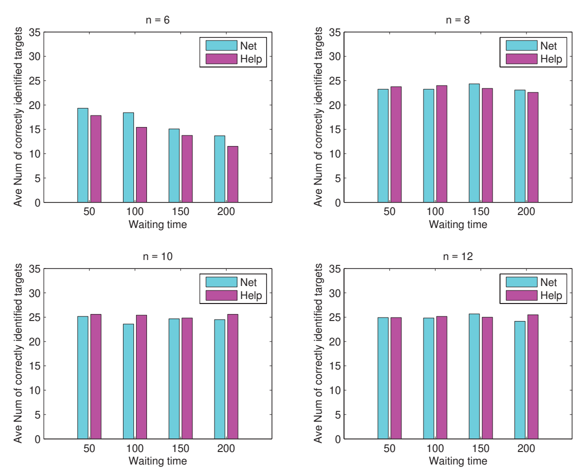

Fig. 3. HelpBots vs. NetBots: Correctly completed tasks plotted against waiting time (in seconds). Each graph shows results from experiments with a different number (n) of robots.

D. Network Results

We need to show that the local graphs in between robots which is all information they have about the system is converging to the global information we saved as a reference. Section IV-C introduced two metrics to evaluate the quality of social network and ranking system. Figure 4 plots the value of the L_1 residual against the number of correctly detected targets. This figure has graphs for four different waiting times, and each graph has data from all four numbers of robots tested. In all situations, the L_1 residual decreases steadily to zero as time progresses. This shows that the error between the local and global PageRank vectors is decreasing over time; in other words, the vectors are converging, exactly as expected.

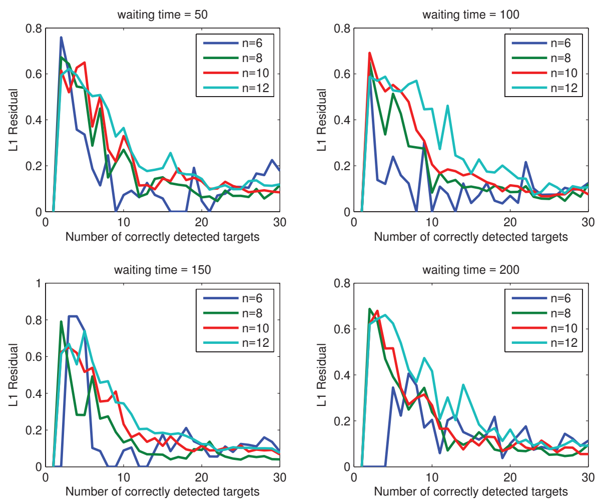

Fig. 4. Value of the L_1 residual metric over number of correctly detected targets for the NetBots algorithm. Each graph represents a different waiting time, and each shows data for four different numbers of robots.

VII. CONCLUSIONS

Automatic target recognition problem requires that a robot or robots locate and classify a collection of objects scattered throughout an arena. A swarm system is well-equipped to solve this sort of problems as in this system the robots'individual actions produces a cohesive group behavior. This is perfect for target recognition because the robots are able to act as a cohesive unit to complete tasks quickly and efficiently.

In this paper to address the heterogeneity of robots in a system and distinguish accurate robots from less accurate ones, we formed a social network among robots based on their cooperation and ranked them based on their perceived past accuracy. Each robot stores a partial graph that contains information about all the links in the social network that it is aware of. Every time robots collaborate on tasks they combine the information in their graphs into a new, updated graph. In this way the robots strive to always have the most updated information about the social network. Links are formed between robots that agree with the majority decision at task. Robots are then ranked using the distributed PageRank algorithm, which ranks robots based on how many links they have. Robots that decide tasks correctly more often will have more links, so they will receive higher ranks. This way the ranks of the robots will eventually indicate their accuracy.

VIII. ACKNOWLEDGMENTS

This work was supported in part by the NSF Grants: 0341601, 0647018, 0717674, 0717680, 0647120, 0525429, 0806931, 0837332.

REFERENCES

[1] E. Sahin, Swarm Robotics: From Sources of Inspiration to Domains of Application, *Lecture notes in computer science* , vol.3342, 2005, pp 10-20.
[2] L. Bayindir, E. Sahin, A Review of Studies in Swarm Robotics, *The Turkish Journal of Electrical Engineering*, vol 15, no 2, 2007, pp 115-147.
[3] L. Iocchi, D. Nardi, M Salerno, Reactivity and Deliberation: A Survey on Multi-Robot Systems, *Reactivity and Deliberation in Multi Agent Systems*, 2001, pp 9-32.
[4] P.Dasgupta, S. O Hara, P.Petrov, A Multi-agent UAV Swarm for Automatic Target Recognition, *Lecture Notes in Computer Science*, vol. 3890, 2006, pp 80-91.
[5] P.Dasgupta, A Multiagent Swarming System for Distributed Automatic Target Recognition Using Unmanned Aerial Vehicles, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 38, 2008, pp 549-563.
[6] D. Miller, P. Dasgupta, T. Judkins, Distributed task selection in multi-agent based swarms using heuristic strategies, *Lecture Notes in Computer Science*, vol. 4433, 2007, pp 158-172.
[7] A.J. Ijspeert, A. Martinoli, A. Billard, and L.M. Gambardella, Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment, *Autonomous Robots*, vol. 11, 2001, pp 149-171
[8] B. Bhanu, Automatic Target Recognition: State of the Art Survey, *IEEE Transactions on Aerospace and Electronic Systems*, vol 22, no 4, 1986, pp 264-379.
[9] J.A. Ratches, C. P. Walters, R. G. Buser, B. D. Guenther, Aided and automatic target recognition based upon sensory inputs from image forming systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 19, no 9, 1997, pp 1004-1019.
[10] J.R. Spofford et al, Overview of the UGV / Demo I1 Program, *Reconnaissance, Surveillance, and Target Acquisition for the Unmanned Ground Vehicle: Providing Surveillance 'Eyes'for an Autonomous Vehicle*, 1997, pp 21-40.
[11] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", *Stanford Digital Libraries Working Paper*, 1998.
[12] Y. Wang, D.J. DeWitt, "Computing pagerank in a distributed internet search system", *Proceedings of the Thirtieth international conference on Very large data bases*, vol. 30, 2004, pp 420-431.
[13] P. Berkhin, A survey on PageRank computing, *Internet Mathematics*, vol 2, no 1, 2005, pp 73-120.
[14] M.G. Kendall,Rank correlation methods, *London: Charles Griffin and Co*, 1995.
[15] R.T. Vaughan, B.P. Gerkey, "Really reusable robot code and the player/stage project", *Software Engineering for Experimental Robotics*, pp 267-289.
[16] B.P. Gerkey, R.T. Vaughan, A.Howard,"The player/stage project: Tools for multi-robot and distributed sensor systems", *Proceedings of the 11th international conference on advanced robotics*, 2003, pp 317-323.
[17] K-Team Corporation, "K-TEAM Corporation: Khepera III Robot", *http://www.k-team.com/kteam/index.php?page=197*, July 2009.
[18] M. Grosseschallau, U. Witkowski,U. Ruckert,"Low-cost bluetooth communication for the autonomous mobile minirobot Khepera", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, 2005, pp 4149-4199.
[19] S. Hauert, J.C. Zufferey,D. Floreano, Evolved swarming without positioning information: an application in aerial communication relay, *Autonomous Robots* , vol. 26, no.1, 2009, pp 21-32.