Target Searching using Cooperative Heterogeneous LEGO Robots

Yan Meng, Richard Lemence, Louis Simons, Edward Schneider, and Imrul Sumit
Embedded Systems and Robotics Lab
Stevens Institute of Technology, Hoboken, NJ 07030
{ymeng1, rlemence, lsimons, eschneid, isumit@stevens.edu}

Abstract - **In this undergraduate summer research project, a team of semi-autonomous mobile LEGO robots is developed from scratch for target searching in an indoor office, where the targets are dispersed randomly on the searching area. A hierarchical heterogeneous architecture is proposed for this multi-robot team: a host computer, master robots, and slave robots. The host computer is in charge of global localization and path planning for robots. An algorithm for visual position tracking of cooperative robots within the operational environment is proposed using a static surveillance camera system. The master robots are responsible for target and obstacle detection, and the slave robots are to carry the load of target. The host computer communicates with master robots through wireless communication, while the master robots send commands to the slave robots through IR sensors. The experiments have been conducted in an indoor office, and the experimental results demonstrate the proposed cooperative multi-robot system is feasible and efficient for a searching task.**

Index Terms – Multi-robot system, target searching, cooperative robots, vision-based localization.

INTRODUCTION

Multi-robot teams are desirable for many reasons in some tasks, such as planetary explorers, urbane search and rescue, and surveillance. A team of robots can simultaneously collect information from multiple locations. Team members can exchange sensor information, collaborate to track and identify targets, or even assist each other to overcome obstacles. By coordinating with its team members, a team can exploit information derived from multiple disparate viewpoints. A single robot, though potentially equipped with a large array of different sensing modalities, is limited at any one time to a single viewpoint. Furthermore, under some hazardous environments, the failure of one or several robot(s) should not lead to failure of the entire mission and the redundancy may increase the fault tolerance of the colony. In general, the multi-robot systems have some obvious advantages over the individual robot, such as parallelism, robustness, scalability, and simpler programming for each one. However, at the same time, with the acquired multiple robot systems, the communication protocols, dynamic task allocation and cooperation between robots need to be developed on the multi-agent systems.

Extensive researches have been done for cooperation and communication in the multi-robot systems. Various coordination and cooperation approaches have been proposed for different application tasks, based on the assumption that that the robots can always communicate with each other [1]-[6]. All these research papers mainly focus on applying the advanced algorithms into the off-the-shelf robot systems. Since these expensive multi robot systems are not available in our laboratory, the students are required to build their own mobile robots using LEGOs with a more advanced on-board embedded microprocessor system instead of the simple RCX microprocessor box provided as one component of the LEGOs. The Handyboard developed by MIT was chosen as the main on-board processor. In addition, the students need to select appropriate on-board sensor systems for this specific target searching applications, such as a CMU Cam2 vision system developed by Carnegie Mellon University, IR sensor, bumper sensor, and SONAR sensors. These self-developed embedded systems for the robot system would consume less power, be less expensive, and be more efficient comparing to using a generic microprocessor system for the specific applications.

Inspired by these research papers in multi-robot systems and constrained by the resources available for this project, such as limited on-board processor power and limited sensor capabilities on the LEGO robots, a hierarchical heterogeneous multi-robot system is proposed and developed from scratch, which can retrieve randomly dispersed targets in an indoor office environment. This multi-robot system can serve as a base platform for future education as well as other research applications.

In general, this undergraduate research project fills a gap between the technical theories and real-world implementation in the student's education. By providing the tools and materials for students to work with complex electrics, control systems, mechanical engineering, computer systems and advanced algorithms, the students can explore and learn about key ideas in technology, engineering, and design through this research project. While traditional technical education strategies tend to promote individualism and competence among students, nowadays, engineering challenges in most areas, and especially robotics, require working with multidisciplinary teams in order to successfully integrate different areas of knowledge. Practical work on robotics at the university level can help engineering students develop the needed communication and working skills for team work.

In this paper, the design and implementation of the proposed multi-robot system are described first. Then, some experimental results are shown and discussed. Finally, the beneficial effects of this project on the engineering education for undergraduate students are discussed.

SYSTEM ARCHITECTURE

In this research project, the students have attempted to build a small set of robots capable of interacting with each other in a swarm manner as well as with a host PC in order to reach a certain objective, in our case, retrieve targets in an obstacle ridden area. Assume the targets are marked as some specific colors, where the color-based target detection algorithms can be applied in our project. Based on this project objective and the constraints of the limited resources and budgets, the system is designed as a hierarchical architecture. In the hierarchy, there are 3 tiers: the host PC, the master robot and the slave robot.

The host PC is in charge of localization, mapping and global path planning. Localization is accomplished through an overhead camera attached to the host PC. The host PC communicates with its direct subordinate, the Master.

The master robot is equipped with a Handyboard microprocessor system which can integrate and process the sensor information. It is also equipped with different sensors to determine its motion, avoid obstacles and detect targets. The master robot actively relays its obstacle data to the host, who in turn relays new heading data to the master. The master will follow these headings and begin its search pattern anew until an obstacle is found. The master also coordinates with its direct subordinate, the slave.

The slave is equipped with a RCX microprocessor as its processor and minimal sensors. Its primary purpose is to carry the load of the targets and provide the muscle that the fully-loaded masters are not capable of. The slave's communication and sensing is limited so it must remain within close proximity of a master robot at all times. With this hierarchy, we can create a steady flow of information with minimal communication required to be able to search an area.

DESIGN OF MASTER ROBOTS

In the above configuration of a heterogeneous multi-robot system, the master, as is shown in Figure 1, is built from scratch using an embedded microprocessor with various sensors, such as a camera, two bumpers, a pair of sonar, and a wireless card.

The base platform for the master has to be smarter and more powerful than the system built from the original LEGO components provided by the Mindstorm. In other words, it should be able to avoid obstacles, detect targets, and communicate with both the host PC and the slaves. To satisfy these requirements, a more advanced on-board processor system and some specific sensors are required.

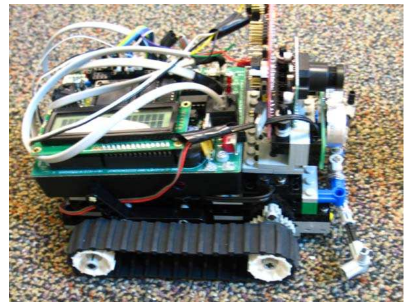

FIGURE 1
A MASTER ROBOT

The Handyboard developed by MIT is chosen as our main microprocessor board. This board is based on the Motorola 68HC11 microprocessor. The Handyboard has four PWM Motor outputs, one SPI port, one SCI port, eight analog inputs, ten digital inputs and an infrared I/O. The entire package runs on a self-contained, rechargeable 9.6V NiCd battery [7]-[8].

I. Self-learning Target Detection

To be able to detect the targets, the CMUCam 2 [9] developed by Carnegie Mellon University is applied in the master robot as the vision system. It is a small lightweight CMOS camera developed for low power applications. This camera system is connected to the Handyboard through the SPI port.

To simplify the target detection, a unique color as well as a flat finish is designed for the target marker, which allows for the least complications in background filtering and lighting, since a unique color field will persist through poor lighting conditions. As for the shape of target marker, a cylindrical design is adopted, which allows for quick recognition regardless of facing direction to the robots.

However, even with our controlled lab environment, the lighting condition may vary in terms of the time of day and weather conditions, which may lead to a high false detection rate. To make the master robot to adaptively respond to the different illumination conditions, a self-learning color-based target detection approach is proposed and developed. The basic idea of this approach is that the vision system has to "learn" the color it will track now rather than reconfiguring the color thresholds every time before the detection. The learning program averages the colors it detected and configures the color thresholds for tracking based on the current detected color. This self-learning approach reduces the setup time prior to testing different color configurations significantly and allowed the robot to adapt to uneven lighting situations.

II. Obstacle Avoidance

To be able to avoid obstacles automatically, an Ultrasonic Sensor is necessary. The UPDS Sensor from Mindsensors.com was originally designed for the LEGO Mindstorm system. The Mindstorm system differs from normal sensors in that it only has two lines, a signal/power line and a ground line. The signal/power line uses an H-bridge in the RCX controller that accounts for polarity changes as well as alternates between the power and the signal, storing the data in its memory. This is not the case in the Handyboard. In order to make this sensor work, a dual-transistor switch circuit was devised to alternate between the 9V power supply and the 5V signal line, with the Handyboard programmed to take measurements only when the 5V is active [12, 13]. To give the board control over the switching, another output has to be found, which can be configured as the digital input #9 of the board. Using this pin and the dual-transistor circuit, we are able to alternate power and signal to the Ultrasonic Sensor. Using RCX, the sonar was able to predict distance up to 40’, however, with Handyboard, this sensor can only detect the object in a range of 10’. Despite its reduced detection range, it can still fulfill its role of aiding in obstacle avoidance.

Primary method of obstacle avoidance is through a “crawler” program. The basic idea is to have the robot turn away from any obstacle it detects in front of it and move forward if nothing is detected. Through the testing, it is found that there is a large blind spot on the emitter-side of the sensor. One simple solution for this detection deficiency is to use bumper sensors to deal with the 45 degree collisions.

III. Localization Sensors

Since the visual tracking location approach, which will be discussed in the later section, can only estimate the position of the robot, to obtain the orientation information of the robot, the Devantech CMPS04 Digital Compass is adopted. This compass runs with a 5V power supply and communicates either by I2C protocol or PWM signal. While the Handyboard can be “bitbanged” to produce I2C through its SPI port [10, 11], a dedicated serial port for the wireless communication to the host PC is needed. Since two of the digital inputs are connected to the TIC line of the 6811 processor, the time difference between the rising edge and falling edge of the PWM signal can be detected. Using this method, the directional data to an accuracy of one degree can be obtained.

IV. Wireless Communication Sensors

For the host-master communication, the RF transceivers from Abacom Technologies [14] are adopted. These interface via RS232 protocol, which is not supported in the Handyboard [15]. The only serial port left on the board is the SPI port. Since the MAX3110E UART is specifically designed for the SPI interface and has a build-in MAX232 to convert TTL voltage to RS232, this UART can be used as the interface for RF transceivers. For the master-slave communication, an IR sensor provided from Mindstrom is installed on the Handyboard through its IR port. The host-master and master-slave communication protocols will be described in the following sections.

DESIGN OF SLAVE ROBOTS

The basic configuration of the slave robots involves a minimalist chassis designed to keep weight down as much as possible without sacrificing too much durability. The RCX microcontroller is mounted on the same level as and in line with the electric motors used for movement. The infrared port on the RCX faces away from the motors, which are connected through a simple, medium-torque gear train to a pair of treads on the sides of the robot. Front and back are defined according to the role of the robot and its program. For instance, in a pair of complementary programs wherein one robot receives commands from another, the IR port of the receiver is on the front, while the reverse is true of the sender. The three slave robots are shown in Figure 2.

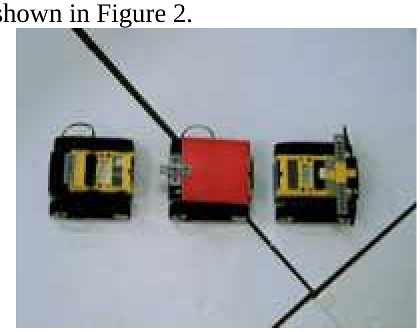

FIGURE 2
THREE SLAVE ROBOTS IN A LINE, WHERE THE MIDDLE SLAVE ROBOT HOLDS A RED COLOR TAG FOR THE OVERHEAD VISION LOCALIZATION SYSTEM

I. Communication Platform

The RCX microcontroller has an integrated infrared emitter/detector. This device is intended to be used as the IR communication for slave-slave. The basic communication format between RCXs is a simple 8 bit (0-255) number that is emitted as an IR “mail.” This signal can then be picked up by other RCXs within range where it is stored in a special register.

II. Slave Control Module

The slave configuration is a standard chassis with forward-facing IR sensor and rear-facing magnets. This is designed to follow a master robot to the target, maneuver around it to retrieve the target, and then follow the master back to wherever the target needs to be. The reason that the slave has to follow the master around is because it has no sensing capability of its own to navigate. The robot is designed as an extension of the master robots, to give them the carrying capacity needed to move the target object.

III. Remote IR Controller

This is an advanced incarnation of the master simulator. The RCX is programmed to be controlled via the triple touch sensor array with no pretense of autonomous behavior. The mater becomes either a mobile or immobile remote control for the slave(s). Commands are continuously transmitted from the control RCX to the subordinate device. This program is used for technology demonstration, mobility testing, and mobile target simulation. This last use bears mentioning in particular.

WIRELESS COMMUNICATION BETWEEN THE MASTER ROBOTS AND THE HOST COMPUTER

In order to communicate commands and sensor information between the host computer and master robots, the RF transceivers from Abacom Technologies [14] are used. The host computer accesses the card via the communication port and the wireless card itself is virtually transparent. Because all members of the system must exhibit a large degree of autonomy, there is no reason to constantly be polling data between members, an event based protocol is proposed. A simplified version of the protocol is displayed in Figure 3, where the central computer represents the host computer and the chief robot represents the master robot.

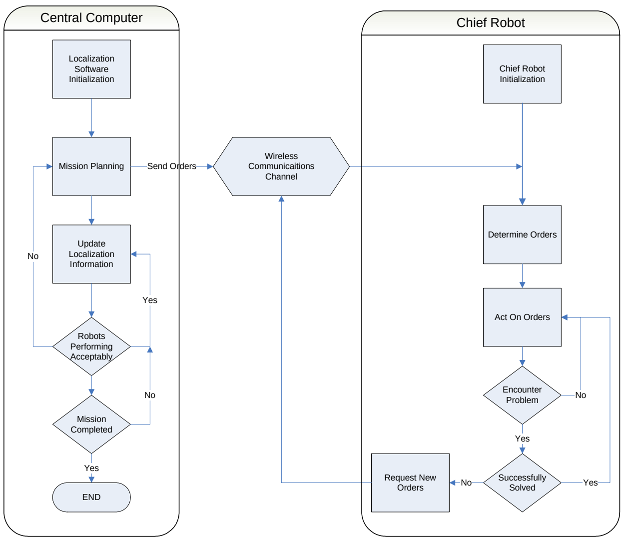

FIGURE 3
PROTOCOL OF WIRELESS COMMUNICATION BETWEEN THE HOST COMPUTER AND THE MASTER ROBOTS

In reality, the protocol also must coordinate the handling of many events and their expression between the operatives. The message formats for these communications that take place over the wireless communications channel are standardized among all operatives of the system.

VISION-BASED LOCALIZATION APPROACH

An algorithm for visual position tracking of cooperative robots within the operational environment is proposed in this paper using a static surveillance camera system hanging on the ceiling of the laboratory. This camera system is connected to the host computer through an Ethernet cable. The vision-based localization system is implemented in a host computer, where the information is stored in a rectangular grid relative to the camera's field of view. Figure 4 is a snapshot of a grid map for the localization system. In the image, the red blocks represent obstacles while the green block on the left side represents the current position of the robot, and the yellow grids show the motion path of the robot. The robots' position is estimated by setting thresholds in HSV color space. The obstacle detection is very rudimentary as most of the handling of obstacles is implemented by the master robot.

To be able to handle the localization of the multiple robots simultaneously, a segmentation algorithm proposed by Cormac Herley [16] is applied in our project by using the different color cardboard square attached on top of each individual robot. An example of this image segmentation is shown in Figure 5.

The localization software can determine and store the positions of any operatives on the field, as well as identify the obstacle positions. The resolution of the grid is flexible enough under different illumination conditions. The localization program also serves as the foundation for the path planning and will display a graphic representation of the computed path for debugging the algorithm.

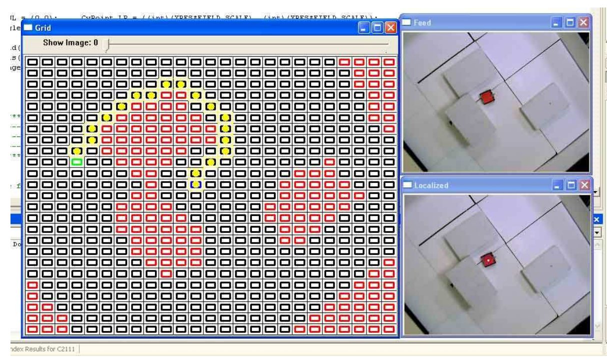

FIGURE 4
A SNAPSHOT OF VISION-BASED GRID MAP FOR LOCALIZATION

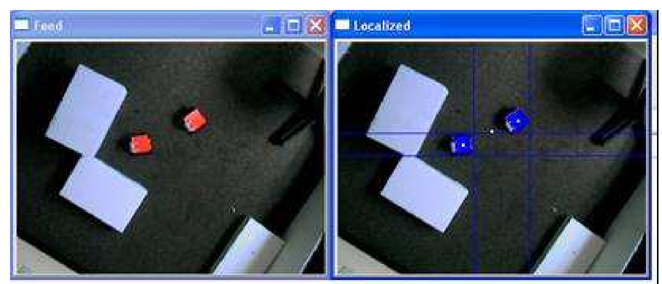

FIGURE 5
AN EXAMPLE OF IMAGE SEGMENTATION

PATH PLANNING

For a mobile robot, in order to navigate in a partially known environment to a given destination position, a path planning is necessary. D* algorithm is applied as the path planner for the mobile robots in this project. It is a dynamic version of the A* algorithm which is *dynamic* in the sense that arc cost parameters can change during the problem solving process [17], and it is more efficient for real world implementation. The base of the D* algorithm came from [17]. In development, the code utilized a 2D environment to mimic a simple map with some rectangular obstacles.

Primarily, the algorithm allocates a grid of node which is maintained in an array. Robot position and the goal location are initially inputted by the user via grid coordinates. A set of rectangular obstacles are also allowed to be set by inputting the grid coordinates of the top-left and bottom-right corners which the robot treat as unknown.

Initially the robot attempts to create a path based on the partially known environment until the next node is acknowledged or when unknown obstacles create a need to re-plan. A heuristic function is defined using the Euclidian distance from the goal position to the robot. Given the visual representation of directional lines which simulate the expansion of the nodes in the search and the direction of the back pointers in the path, the search can be conducted. The robot follows its path and replans until the goal is reached or a path doesn't exist. Figure 6 is one example of the path planning results, where the structure indicates the node info, the state changes, costs, and node positions during the iterations in search.

The path planner has to work with the vision-based localization algorithm. Path planner maintains information on each node it has examined and the predicted optimal path from any node to the goal. A combination of path distance along with exponential function would have allowed proper navigation while accounting for error.

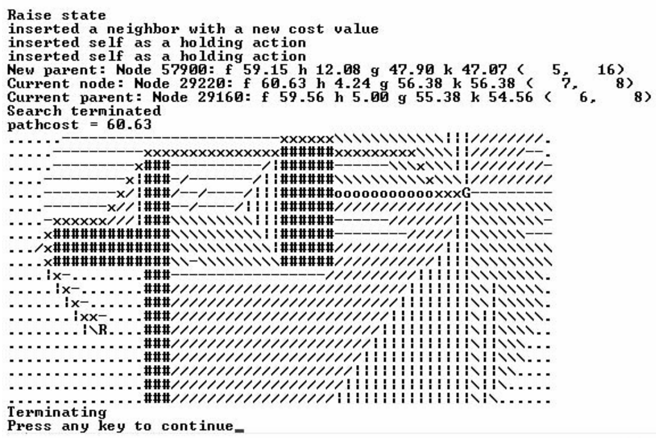

FIGURE 6
ONE EXAMPLE OF PATH PLANNING RESULTS

EXPERIMENTAL RESULTS

The extensive experiments have been conducted on the multiple LEGO Mindstorm robots in an indoor office environment. The system includes one host computer, and one overhead-hanging CCD camera which connected to the host computer, one master robot and three slave robots. Although in the proposed system architecture, two masters and two slaves are desirable. Due to the limited time and resources for this summer project, only one master robot is built up. The area used for experiments is a 2m x 2m space surround with white cardboard walls and some box obstacles are randomly distributed inside the space. There is one target randomly located inside this area. Figure 7 shows the team of robots inside this working environment, where blue cylinder is the target and white boxes are obstacles.

The master robot is equipped with a microprocessor system RCX box, a Handyboard, a CMU Cam2, a wireless card RTcom-RS232-HS, an Ultrasonic distance sensor, a digital compass, an IR sensor, a light sensor, a tough sensor, motors, and wheels. The slave robot is equipped with a RCX box, an IR sensor, a light sensor, a touch sensor, motors and wheels.

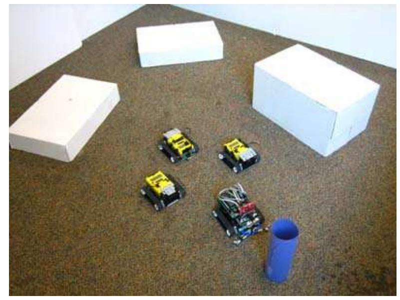

FIGURE 7
THE SEARCHING TASK IMPLEMENTATION BY A HETEROGENEOUS MULTI-ROBOT SYSTEMS

I. Master Robot Target Detection and Obstacle Avoidance

In the above described environment, the master robot needs to be tested for target detection. Before the detection procedure starts, the master robot is placed in front of the blue cylinder target it will track, and starts the self-learning program to configure the color threshold under the current lighting condition. Then, the target is placed somewhere in the searching area, and the master robot starts from any initial position looking for the target. Initially the master robot moves randomly since it does not know where the target is. Once it detects the target through the vision system, it will move toward the target and stop at the place with distance of 0.3 meters from the target. To evaluate the obstacle avoidance capability, one big white box is place between the robot and the target, and let the searching procedure start again.

30 runs have been conducted with different combination of the target locations and initial positions of the robot. The average searching time is 1.5 minutes with variance 0.6 minutes. The detection rate is about 87.45%.

II. IR Communication Test

To ensure that the slave robots could coordinate their efforts, the range and flexibility of the IR link need to be tested, which can be accomplished in two basic ways.

In direct line of sight testing, it was found that the RCX emitter can successfully communicate over a distance of approximately 5 meters or more. Reliable signals can be transmitted over 2-3 meters even under some adverse situations. In fact, continuous signal integrity can be maintained with some effort outside on a sunny day, in the face of solar infrared interference.

In addition to direct communication, some explorations of the reflective capability of the RCX IR emitter are also investigated. The "arena" used for much of the hardware testing is made with white cardboard walls and white box obstacles. In this environment, fairly reliable communication at oblique or even opposed emitter-receiver angles is observed. In fact, in one experiment, an RCX unit was placed inside one of the obstacles, and yet still intermittently received signals.

One pair of the programs used, while not specifically designed to test communications, involves literally continuous communication between two RCX units, and this program set has been successfully employed in several environments, such as in the laboratory, out of doors on grass, out of doors on cement, and during a presentation of the work in adverse optical conditions. In general, the IR emitter and receiver on the RCX units are found to be a fine and robust base for a communication scheme.

III. Slave Following and Target Grabbing

A vital part of the project is the slave robots' ability to follow the master and grab the target to a preset destination. In general, this is a matter of successfully timing and coordinating discrete movement behaviors. The slave robots lack any sensing capability yet have a need to maneuver around the masters to perform retrieval and grabbing. Getting a hold on the target is not ideal for cluttered environments, because the process involves dragging the object. Figure 8(a) shows a slave robot is following a master robot for the target, and Figure 8(b) shows a slave robot is holding the target and dragging it to a preset destination. A mechanism to lift the object would add complexity, but may be worth the investment in the long run, as it could actually turn out to simplify the programming.

(A) (B)

FIGURE 8

(A) A SLAVE IS FOLLOWING A MASTER FOR A TARGET; (B) A SLAVE ROBOT IS GRABBING A TARGET AND MOVING IT TO THE DESTINATION POSITION.

IV. Cooperative Searching and Target Transportation

The vision-based localization algorithm and the D* path planning algorithm are implemented in the experiment. The localization results are satisfied, which can obtain the accuracy of up to 0.2cm. One host computer and four mobile robots, one master and three slave robots, are conducted in a searching task for one target in the scene shown in Figure 8. 35 runs have been conducted, and the average searching time plus target transportation time is 4.39 minutes with variance of 1.2 minutes.

CONCLUSION

A searching task has been implemented by a heterogeneous multi-robot system using LEGO Mindstorm system in an indoor office environment. Several coordination and communication issues between the robots have been addressed in this paper. One major accomplishment of this project is building up a fully autonomous master robot which is capable of detecting and tracking target, avoiding obstacles, and communicating with the host computer as well as the slave robots. Another major contribution is the combination of the vision-based localization and path planning algorithms.

The combination of simplicity and complexity is the key to this robotics system. Instead of having a single, large complex robot to handle all these tasks, we have several robots that specialize in different aspects of the search process .

This multi-robot system has been used for one of the robotics course for both single robot design and the multi-robot cooperation with two main objectives: as a test bed platform for the students in the course so that they can test their designs of electronic boards or software algorithms. This system can also serve as the platform for future undergraduate summer research projects, where more advanced algorithms and strategies can be explored.

REFERENCES

[1] H. Hu, I. Kelly, D. Keating, D. Vinagre, "Coordination of multiple mobile robots via communication", *Proceedings of SPIE'98 Mobile Robots XIII Conference*, Nov. 1998, pp. 94-103.

[2] R. R. Murphy, "Marsupial and shape-shifting robots for urban search and rescue", *IEEE Intelligent Systems*, 15(2), 2000, pp.14-19.

[3] T. Balch and R.C. Arkin, "Communication in reactive multiagent robotic systems", *Autonomous Robots,* 1(1), 1994, pp. 27-52.

[4] B. S. Pimentel and M. Campos, "Cooperative Communication in Ad Hoc Networked Mobile Robots", *Proceedings of 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, October 2003, pp. 2876-2881.

[5] K. E. Bavarri0Sernebtm, C.J.J. Paredis, P.K.Khosla, "A Beacon System for the Localization of Distributed Robotic Teams", in *Proceedings of the International Conference on Field and Service Robotics*, Pittsburgh, PA, 1999.

[6] T. Tooth-Fejel, "LEGO™s to the Stars: Active MesoStructures, Kinetic Cellular Automata, and Parallel Nanomachines for Space Applications", *1996 International Space Development Conference*.

[7] The Official Handyboard support site: http://www.handyboard.com

[8] The Official Handyboard discussion group: http://groups.yahoo.com/handyboard

[9] Carnegie Mellon University's CMUCam Resource page : http://www.cs.cmu.edu/~cmucam2/

[10] Maxim's site for the MAX3110E SPIUART:http://www.maxim-ic.com/quick_view2.cfm?pdf_num=2052

[11] The Seattle Robotics Society Homepage: http://www.seattlerobotics.org/

[12] HandyBoard Hackers' Resource:http://funkmunch.net/hc11-ide/

[13] Motorola's main site, useful for 68hc11 architecture information: http://www.motorola.com

[14] ABACOM's transceiver page : http://www.abacom-tech.com/transceivers.htm

[15] Fred Martin's homepage; developer of the Handyboard: http://www.cs.uml.edu/~fredm/

[16] C. Herley, "Recursive Method to Detect and Segment Multiple Rectangular Objects in Scanned Images", MSR-TR-2004-01. 2005. ftp://ftp.research.microsoft.com/pub/tr/tr-2004-01.pdf

[17] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments," *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1994.