A Cost-Effective and Open Mobile Sensor Platform for Networked Surveillance

Gang Li, Jianhao Du, Chun Zhu and Weihua Sheng
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK, 74078, USA

ABSTRACT

In this paper, a compact, low-cost and open mobile sensor platform consisting of multiple ASCCbots for networked surveillance is presented. This platform is based on commercial off-the-shelf components and open source software. Compared to existing research platforms, our platform is reliable, reconfigurable and easy to duplicate. We develop novel algorithms for object detection on ASCCbot. Due to the distributed computing nature of the platform, we also conduct collaborative target localization, which is realized by fusing the data from omni-directional camera, laser range finder and sensor pose estimation. The performance of the mobile surveillance system is evaluated through experiments. The results from the experiments prove that the proposed platform is a promising tool for networked surveillance research and practice.

Keywords: Networked Surveillance, Collaborative Target Localization, Robot Operating System

1. INTRODUCTION

1.1 Motivation

Mobile sensor network has the ability to collect environmental information using a set of mobile sensor nodes.[1] With features such as wireless communication, distributed computing, and mobility, mobile sensor network has great potential in military surveillance. In order to support the research in networked surveillance, we propose the ASCCbot system, which is a cost-effective and open mobile sensor platform.

Simulation software, such as Microsoft Robotics Developer Studio,[2] Webots[3] and Player/Stage,[4] has been widely used in mobile robot research. But the robustness, efficiency and intelligence of the robotic algorithms can only be thoroughly tested by the realtime sensory data from real environment. Therefore, there is a need to employ a physical platform for mobile sensor network research.

In order to develop a networked mobile sensor platform, we propose that each mobile sensor node should meet the following criteria:

- Autonomy: Each node should have a reasonable amount of computational resource including CPU power and memory for its autonomous navigation, signal processing and collaboration with its neighbors;
- Reconfigurability: It should be reconfigurable so that it can fit in various research projects, thus calling for an open hardware and software architecture;
- Robustness: It should be built on a sturdy mobile platform to ensure repeated experiments;
- Easiness of duplication: It should be built mostly out of commercially-off-the-shelf (COTS) parts instead of customized hardware, which will allow other researchers to quickly duplicate the platform for their own research.

Further author information: (Send correspondence to Weihua Sheng)
Weihua Sheng E-mail: weihua.sheng@okstate.edu, Telephone: 1 405 744 7590
Gang Li E-mail: gang.li@okstate.edu
Jianhao Du E-mail:jianhao.du@okstate.edu
Chun Zhu E-mail:chunz@okstate.edu

Signal and Data Processing of Small Targets 2011, edited by Oliver E. Drummond, Proc. of SPIE Vol. 8137, 813713 · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.895520

In recent years, various existing mobile robot platforms have been developed by different research groups, but most of them can only satisfy a subset of the above criteria. Robomote is a tiny mobile sensor platform designed for large-scale ad-hoc sensor network research.[5] It is a low-cost compact platform, but it could not provide accurate mobility for localization and mapping applications. Pioneer 3DX[6] is customized with Laser Range Finder and cameras for mobile robotic research, but it is relatively expensive and has limited reconfigurability.

ASCCbot is created in the Advanced Sensing, Computation & Control Lab (ASCC) at Oklahoma State University. Compared with existing platforms, the proposed ASCCbot is a compact, intelligent mobile multi-robot platform which is relatively inexpensive (around $2500), purely open-source, extendable, duplicable and equipped with basic functionalities. The criteria listed above can be fully satisfied by the ASCCbot.

The hardware system of ASCCbot is composed of an iRobot Create,[7] an Atom processor-based computer, a Laser Range Finder(LRF), and an omnidirectional camera, which are all off-the-shelf components. The software system is built upon the ROS (robot operation system[8]), which functions like a real operation system including low-level device control, wireless communication, package management, etc. The networked surveillance application of ASCCbots is facilitated by the network and distributed computing functionalities of ROS.

The features of ASCCbot can be summarized as below:

- All components are off-the-shelf, inexpensive, easy to duplicate and extendable;
- It is developed based on ROS, which is open-source and friendly for wireless multi-agent application;
- Especially competent in networked surveillance application.

1.2 Related Work

We realize that in recent years there exist many multi-robot or multi-sensor platforms developed in various research projects. However, these platforms do not meet most of the criteria above. Some of the platforms are developed based on small, tightly resource-constrained simple robots which lack sufficient computational power and sensing capability to conduct accurate navigation or implement desired signal processing and collaboration algorithms. These platforms include the MicaBot,[9] CotsBot,[10] Robomote[5] robots, and the commercial Khepera robot and its siblings. Some of the platforms are designed from the scratch, which makes it hard to quickly duplicate due to their customized hardware design. These platforms include the COMET testbed developed by Cruz et al.,[11] the MARS testbed developed by Grocholsky et al.,[12] the Scarab robots[13] developed by Michael et al., and the MVWT-II Hovercraft testbed developed by Jin et al.[14] Some of the platforms are developed based on full size commercial robots such as the Pioneer robots, which lack the reconfigurability and may incur significant cost in setting up large sensor networks. These platforms include MIT's multi-vehicle platform[15] and the multiple heterogeneous robot testbed developed under the DARPA SDR program.[16] Additionally, many testbeds[17,18] have to be constrained in a very small space, which is not sufficient to capture the characteristics of large real world environments.

The paper is organized as follows. Section 2 describes hardware setup, software configuration and networking between mobile sensor nodes. Section 3 presents the object detection function of ASCCbot and its evaluation through experiments. Section 4 provides performance evaluation for networked surveillance applications through collaborative target localization experiments. Section 5 concludes the paper and discusses our future work.

2. HARDWARE AND SOFTWARE DESIGN

2.1 Hardware Setup

The ASCCbot is built on an iRobot Create with an Atom processor-based computer called FitPC2,[19] a Hokuyo LRF[20] and a Q24 panoramic camera[21] or a webcam. The iRobot Create is a platform that is designed for robotics development and possesses a serial port through which sensor data can be read and motor commands can be issued using the iRobot Roomba Open Interface protocol. The FitPC2 is a small, light, mini computer. The Laser range finder (LRF) is a USB-powered device which uses a laser beam to determine the distance to an object. The Hokuyo LRF URG-04LX has a measuring range between 20 mm and 4094 mm, a scanning range of

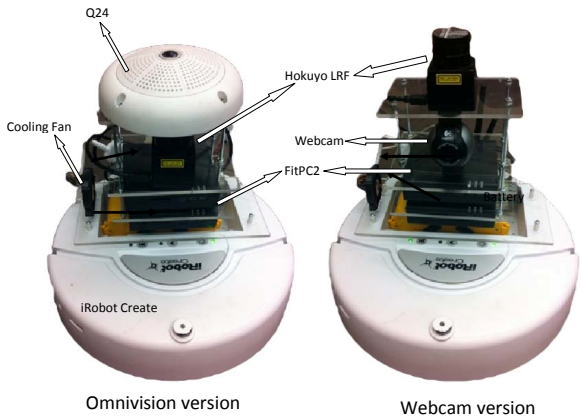

Figure 1. Two versions of ASCCbot.

Figure 2. A mobile sensor network.

240°, a scanning rate of 100 ms/scan, a distance accuracy of ±3% and an angle resolution of 0.36°. The fish-eye camera (Q24) is capable of providing different views simultaneously including a panoramic view so that it can cover the surrounding area of the mobile platform. The camera provides a highest resolution at 3 Megapixels and color images scalable from 160×120 to 2048×1536, and it uses an Ethernet-based interface. The features of the camera (including resolutions, frame rates, etc.) can be easily adjusted by sending a web request. Moreover, the zooming and panning of the camera lenses can be done by a virtual PTZ function. The camera itself is a web server so that the stream of live images can be obtained by setting up a socket connection. Besides the aforementioned components, external batteries are used to power FitPC2 and Q24 while USB-powered mini fan is used to cool the FitPC2.

The mechanical design of the platform endows ASCCbot with stability, and extendability. On top of the iRobot, a plexiglass stand is designed to support the control, communication and sensing devices (see Figure 1). The final design of the plexiglass stand has four layers. Devices on the stand are FitPC2, Hokuyo laser range finder (LRF), Mobotix Q24 360° fisheye IP camera or a webcam, and a USB-powered cooling fan. There are two layout versions for the stand. The first and second layers are reserved for FitPC2s for both versions, in case we need two of them. But for the third and fourth layer, Omnivision version is to put LRF on the third layer and Q24 on the fourth layer; Webcam version is to put other component like webcam on the third layer and LRF on the fourth layer. The plexiglass stand design fits the iRobot create on the bottom very well, it is very stable for setting all the components on top of it, and more importantly it can be conveniently extended by adding layers.

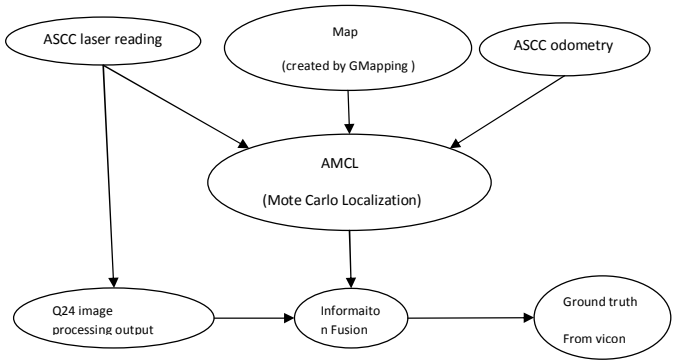

Figure 3. ROS nodes structure for the proposed platform.

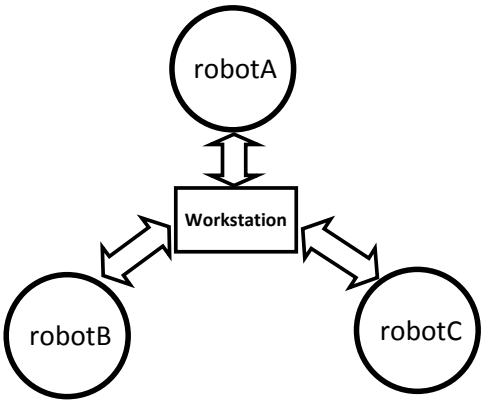

Figure 4. Centralized network setup

2.2 Software Configuration

Robot Operating System (ROS) is an open-source, meta-operating system for robots. It provides services similar to real operation systems, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. ROS has two basic parts, one is the core part of ROS which functions as an "Operation System", the other part is the packages contributed by the whole ROS community. The proposed ASCCbot platform is utilizing the contributed ROS packages and also will contribute its source code to the community later.

In ROS, a program can be divided into different nodes which can be distributed to different computing devices in the same network. Nodes are separate processes which can receive messages from and publish messages to any other nodes. The driver of the hardware units or a data processing method can be made as one node, as long as all the nodes share one "ROS master".

2.3 Networking between Mobile Sensor Nodes

The inherited distributed computing feature of ROS makes the networking of mobile sensor nodes straightforward. Any ROS node can run in one computer or be distributed to multiple computers, and *roscore* can also be put on any computer within the network. For each single computer, ROS_MASTER_URI and $ROS\_IP/ROS\_HOSTNAME$ are used to define the *roscore* IP and its own IP. Once the IP addresses are set up, all the computers can communicate with each other. If multiple computers want to run the same node, *namespace* need to be used. In our case, *namespace* $robotA/, robotB/, robotC/$ are used to specify different mobile sensors.

There are two types of setup for the network. One is distributed setup (see Figure 5) where all the nodes are equal. The other one is centralized setup (see Figure 4) where there is a laptop or other portable computer ("workstation" in Figure 4) which is responsible for the networking.

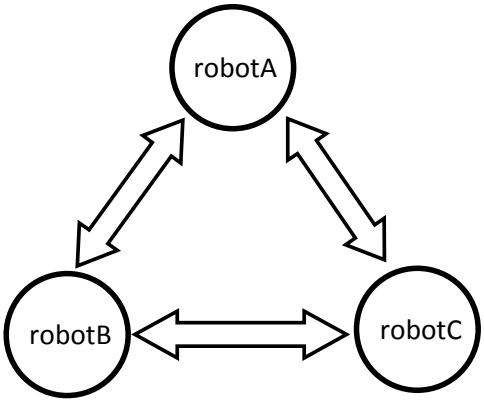

Figure 5. Distributed network setup

Figure 6. Detection using color gegmentation.

3. OBJECT DETECTION

3.1 Object Detection Algorithms

The specific task of object detection is to recognize certain object and to locate it when the autonomous robot is exploring in an unknown environment. Despite a large amount of work done on object recognition in computer vision area, few have investigated methods for object recognition on a mobile robot platform and evaluated them in real-time environment. Compared to normal surveillance system, our camera is mounted on a moving platform so the light condition is changing and the images contain a significant amount of noise.

On our platform, we design two different levels of vision processing algorithms to detect objects based on vision processing loads. The low-level task is to detect the orange traffic cones simply using color segmentation which is fast and with low computational cost, while the high-level task is to recognize the OSU logo by taking advantage of SURF (Speeded Up Robust Features)[22] features which is a bit time-consuming but effective.

Algorithm I: Cone detection

Color segmentation technique is used to detect the traffic cones. The purpose of color segmentation is to find contiguous regions in which individual pixels share common characteristic. So it is a simple and fast way to detect the traffic cone due to its distinctive color. Since RGB color space is very sensitive to lighting changes, the captured image is converted from RGB color space to HSV color space first. HSV space decouples the intensity component from color carrying information. Then color segmentation in HSV color space can be achieved by setting a threshold for each channel. Binary image is obtained by filtering HSV channels. After applying the Gaussian filter and the morphology method (dilate and erode) to reduce the noise, the cone region is detected just as shown in Figure 6.

Algorithm II: Logo recognition

Sometimes the object we want to detect may not be as easy as the traffic cone, for example, logos or trade markers, so local features are used to provide detail information about the object. Here we use SURF features

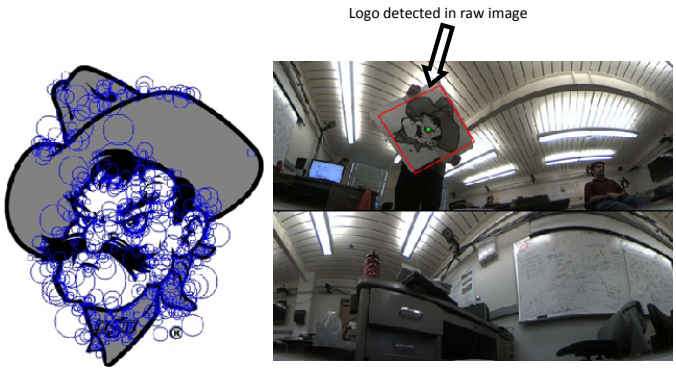

Figure 7. OSU logo recognition using SURF features.

Figure 8. Object detection performance for cone detection(left column) and logo detection(right column) with different rotation speed ((a) Low speed (b) Medium speed (c) High speed).

to detect the logo "Pete" of Oklahoma State University. Firstly SURF features are extracted from each new image frame. Then feature matching is accomplished by computing the correspondences between image features and the object template in our database. Based on the matched features the object is detected by an affine transformation with the object stored in the database just as shown in Figure 7.

3.2 Object Detection Experiments

Moving object detection for mobile robot has been studied by many researchers[23].[24] In the object following scenario, the rotation of robot platform affects the object detection performance most. In order to achieve an acceptable object detection accuracy, rotation rate has to be limited. Experiments have been designed to evaluate the relation between rotation rate and objection detection accuracy as below.

With an image resolution of 1024×768 pixels the processing speed is around 4fps for cone detection and 2fps for logo recognition which ensures the real-time processing. The object (cone or logo) is deployed 1 meter away from the mobile robot which is rotating at different levels of speed (0.26 rad/s, 0.52 rad/s and 0.87 rad/s). For each speed we ran our program for 5 minutes. The experiment results are shown in Table 1. The images become more blurred when the speed of rotation increases, which affects the accuracy of object detection (Figure 8). From Table 1, it is shown that for the low-level task the detection rate drops only a little with changing of speed which means that color features are quite robust. For the high-level task the detection rate drops significantly with blurred image which means the rotation rate should not be too fast if local features are used.

4. COLLABORATIVE TARGET LOCALIZATION

Target localization is one important step for surveillance. In order to evaluate the networked surveillance potential of ASCCbot, a collaborative target localization experiment is designed and carried out. The idea is that each single ASCCbot can detect the target and give an estimate of the target location. Then different target location

Table 1. Detection rate with different rotation speed

Detection Accuracy / Rotation Speed / Detection Mode	Low speed	Medium speed	High speed
Cone Detection	99.09%	97.41%	97.05%
Logo Recognition	91.89%	64.15%	25.86%

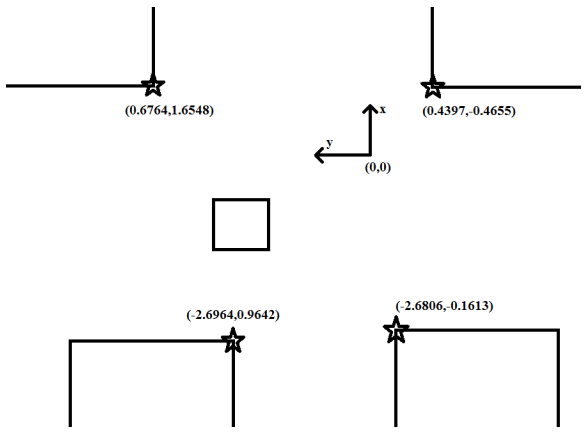

Figure 9. Landmarks in Vicon coordinate

estimates from multiple ASCCbots will be fused by considering the uncertainty of each estimate. At last, the location estimate after fusion will be compared with the ground truth provided by the Vicon motion capture system.[25]

In order to compare the object location estimate from mobile sensor nodes with object location ground truth from the Vicon motion capture system, a calibration process needs to be carried out to overlap the coordinates of the Vicon system and the motion sensor node. Four desk corners are used as landmarks in both coordinates. Figure 9 and Figure 10 show the map from the Vicon system and the mobile sensor node, respectively. The average differences along x axis and y axis are found to be 0.15 meters and 0.38 meters.

After the calibration process, the collaborative target localization experiment can proceed. First, the map of an office room is created by running Simultaneous Localization and Mapping (SLAM)[26] on one of the ASCCbots. The starting pose of the ASCCbot will determine the origin and the orientation of the map. Then the Vicon motion capture system will also be calibrated with the same origin position and orientation. By doing this, the Vicon frame and the office map frame will be the same (see Figure 11). Three ASCCbots will be placed inside

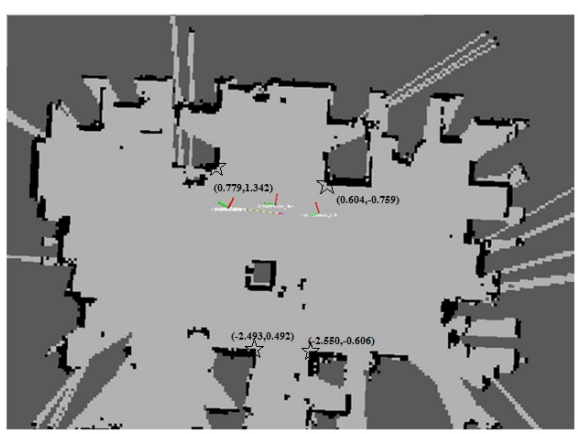

Figure 10. Landmarks in ASCCbot coordinate

Table 2. Target location estimates from ASCCbots. The unit of x and y coordinate are meters while the unit for uncertainty is meter square

	X estimation			Y estimation			Uncertainty					
	Set1	Set2	Set3	Set1	Set2	Set3	Set1 x	Set1 y	Set2 x	Set2 y	Set3 x	Set3 y
RobotA	0.77325	1.67626	1.39862	0.22630	0.70589	1.2412	0.10967	0.000935	0.01248	0.002561	0.01476	0.00780
RobotB	0.78929	1.52608	1.3822	0.22650	0.6998	1.2602	0.0060329	0.002237	0.012039	0.0012959	0.017079	0.002451
RobotC	0.81590	1.51796	1.4641	0.26530	0.7162	1.286	0.008578	0.0015116	0.009264	0.000376	0.0225	0.003677

Table 3. Comparison between the estimate after fusion and the ground truth. (unit is meter)

	X estimation after fusion	X from vicon	Difference X	Y estimation after fusion	Y from vicon	Difference Y
Set 1	0.79946	0.7621	0.03736	0.25819	0.2256	0.03259
Set 2	1.56716	1.596	-0.02884	0.71187	0.7413	-0.02943
Set 3	1.41002	1.455	-0.04498	1.26587	1.266	-0.0013

the office room with AMCL node running, which will publish robot position estimates through Monte Carlo localization algorithm (see Figure 12). Meanwhile, the Q24 camera will detect specific target and output the angle information with respect to the ASCCbot. With the help of the angle, the laser can read the distance at that angle, which yields the distance estimates of targets to the ASCCbots. After the robot pose, target angle and target distance estimates are all acquired, the target location estimate can be computed according to (1) and (2). Since the pose estimates from AMCL has uncertainty, the fusion of the three target location estimates can be fused based on those uncertainties. Ground truth from the Vicon system will then be compared with the fused estimation result.

$$X_{estimate} = x_{robot} + distance \times \cos\left(angle_{robot} + angle_{target}\right) \tag{1}$$

$$Y_{estimate} = y_{robot} + distance \times \sin\left(angle_{robot} + angle_{target}\right) \tag{2}$$

The fusion strategy is summarized as the equation below:

$$W_a = \frac{1/U_a}{1/U_a + 1/U_b + 1/U_c} \tag{3}$$

$$W_b = \frac{1/U_b}{1/U_a + 1/U_b + 1/U_c} \tag{4}$$

$$W_c = \frac{1/U_c}{1/U_a + 1/U_b + 1/U_c} \tag{5}$$

$$X_{fusion} = X_a \cdot W_a + X_b \cdot W_b + X_c \cdot W_c \tag{6}$$

$$Y_{fusion} = Y_a \cdot W_a + Y_b \cdot W_b + Y_c \cdot W_c \tag{7}$$

where, W stands for the weight, while U indicates the uncertainty.

In Figure 13, it shows one set of the collaborative target localization results. The green cylinder indicates the target location estimate after fusion while the yellow star shows the ground truth. The experiment results can be found in Table 2 and 3.

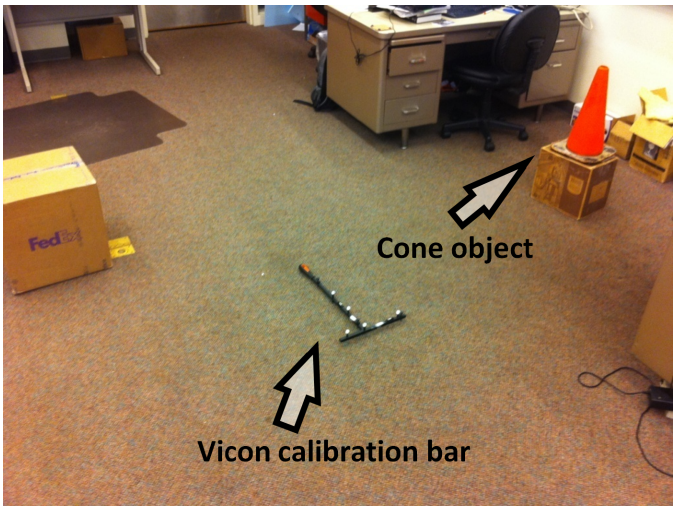

Figure 11. Vicon calibration.

Figure 12. Collaborative target localization with three ASCCbots.

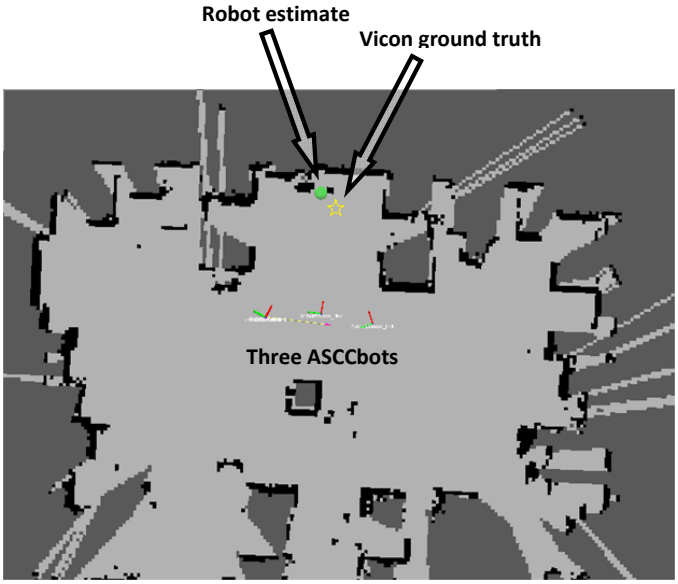

Figure 13. Collaborative target localization result.

5. CONCLUSIONS AND FUTURE WORK

This paper proposes a cost-effective, compact, intelligent mobile sensor platform consisting of ASCCbots. The hardware and software design for the ASCCbot is presented. Object detection and collaborative target localization are evaluated as two functions through experiments. The results of the experiments prove that the proposed mobile sensor platform is a useful platform for networked surveillance research.

In the future, new strategies need to be adopted to improve the performance of the object detection and target localization. New functionalities need to be implemented in order to realize realtime networked surveillance applications.

ACKNOWLEDGMENTS

This project is supported by the DoD ARO DURIP grant 55628-CS-RIP and the DEPSCoR grant W911NF-10-1-0015.

REFERENCES

1. T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, and L. Gu, "Energy-efficient surveillance system using wireless sensor networks," in *In Mobisys*, pp. 270–283, ACM Press, 2004.
2. Microsoft, "http://www.microsoft.com/robotics/," 2011.
3. Webots, "http://www.cyberbotics.com/," 2011.
4. Player, "http://playerstage.sourceforge.net/," 2011.
5. G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks," 2002.
6. P. 3DX, "http://www.mobilerobots.com/researchrobots/," 2010.
7. iRobot Create, "http://store.irobot.com/shop/index.jsp?categoryid=3311368," 2011.
8. R. wiki, "http://www.ros.org/wiki/," 2010.
9. M. McMickell and B. Goodwine, "Micabot: A robotic platform for large-scale distributed robotics," in *IEEE International Conference on Robotics & Automation*, 2003.
10. S. Bergbreiter and K. Pister, "Cotsbots: An off-the-shelf platform for distributed robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1632–1637, 2003.

11. D. Cruz, J. Mcclintock, B. Perteet, O. A. A. Orqueda, Y. Cao, and R. Fierro, "A multivehicle platform for research in networked embedded systems," 2007.
12. B. P. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robotics & Automation Magazine* **13**, pp. 16–25, Mar. 2006.
13. N. Michael, J. Fink, and V. Kumar, "Experimental Testbed for Large Multi-Robot Teams: Verification and Validation," *IEEE Robotics and Automation Magazine* **15**(1), p. 61, 2008.
14. Z. Jin, S. Waydo, E. B. Wildanger, M. Lammers, H. Scholze, P. Foley, D. Held, and R. M. Murray, "Mvwt-ii: The second generation caltech multi-vehicle wireless testbed," 2004.
15. E. King, Y. Kuwata, M. Alighanbari, L. Bertuccelli, and J. P. How, "Coordination and control experiments on a multi-vehicle testbed," in *American Control Conference (ACC)*, pp. 5315–5320, (Boston, MA), 30 June-2 July 2004.
16. A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH* **25**(5/6), p. 431, 2006.
17. V.Vladimerou., "Multi-vehicle systems control over networks," *IEEE Control Systems Magazine* **26**(3), pp. 56–69, 2006.
18. D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, and J. Lepreau, "Mobile emulab: A robotic wireless and sensor network testbed," in *IEEE INFOCOM*, 2006.
19. fit-PC2, "http://www.fit-pc.com/web/," 2011.
20. Hokuyo laser, "http://www.hokuyo-aut.jp/," 2011.
21. Q24, "http://www.mobotix.com/," 2011.
22. T. T. Herbert Bay, Andreas Ess and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vison and Image Understanding* **110**(3), pp. 346–359, 2008.
23. B. Jung and G. S. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in *Conference on Intelligent Autonomous Systems*, 2004.
24. S. Feyrer and A. Zell, "Detection, tracking, and pursuit of humans with an autonomous mobile robot," in *Intelligent Robots and Systems*, 1999.
25. vicon motion capture systems, "http://www.vicon.com/."
26. M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation* **17**, pp. 229–241, 2001.