Experiments in Instrumenting Wireless Sensor Networks for Real-Time Surveillance

Phoebus Chen[*], Songhwai Oh[*], Michael Manzo[*], Bruno Sinopoli[*], Cory Sharp[†], Kamin Whitehouse[*], Gilman Tolle[*‡], Jaein Jeong[*], Prabal Dutta[*], Jonathan Hui[*‡], Shawn Shaffert[*], Sukun Kim[*], Jay Taneja[*], Bonnie Zhu[*], Tanya Roosta[*], Mike Howard[*], David Culler[*‡], and Shankar Sastry [*]
{phoebusc,sho,sinopoli,kamin,get,jaein,prabal,jwhui,sms,binetude,taneja, bonniez,roosta,mhoward,culler,sastry}@eecs.berkeley.edu, mike@manzo.org, cory@moteiv.com

[*] Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, California 94720

[†] Moteiv Corporation
Berkeley, California 94704

[‡] Arched Rock Corporation
Berkeley, California 94704

***Abstract*— On August 30, 2005, we successfully demonstrated a large-scale, real-time, surveillance and control application on a wireless sensor network. The task was to track multiple human targets walking through a 5041 square meter sensor field and dispatch simulated pursuers to capture them.**

We employed a multi-target tracking algorithm that was a combination of a multi-sensor fusion algorithm for fusing binary detections and a Markov chain Monte Carlo data association (MCMCDA) algorithm that can initiate and terminates tracks autonomously and is robust to a high level of false alarms and missing measurements, a common problem in sensor networks. The tracks were used by a multi-agent coordination and control algorithm to capture the evaders.

We were able to demonstrate successful pursuit of two crossing targets and successful tracking of three targets moving through a 144 node sensor field. To the authors' best knowledge, this experiment is the largest demonstration to date of a real-time tracking and control system on a wireless sensor network that does not use classification information.

I. Overview and Setup

At the DARPA Network Embedded Systems Technology (NEST) final experiment on August 30, 2005, we demonstrated the feasibility of building a real-time surveillance and control system on a wireless sensor network. The system employed the tracking algorithm described in our companion paper [1] which uses multi-sensor fusion to combine binary detections and Markov chain Monte Carlo data association (MCMCDA) to track human targets walking through a sensor field. Then, it dispatched simulated pursuers following the robust minimum time-to-capture coordination and control algorithms described in [2] to capture the targets.

The experiment was performed on a large-scale, long-term, outdoor sensor network testbed deployed over a short grass field at UC Berkeley's Richmond Field Station (see Figure 1, right). A total of 557 Trio [3] sensor nodes were deployed at the NEST final experiment for demonstrating new sensor network hardware and software services, and 144 of those nodes were allotted for the multi-target tracking and pursuit experiment. The 144 nodes used for the tracking and pursuit experiment were deployed at approximately 5 meter spacing in a 12×12 grid. Each node was elevated by a tripod to prevent obstruction of the passive infrared (PIR) sensors by grass and uneven terrain.

Fig. 1. Hardware for the sensor nodes. (left) Trio sensor node on a tripod. On top is the microphone, buzzer, solar panel, and user and reset buttons. On the sides are the windows for the passive infrared sensors. (right) A live picture from the 2 target pursuit-evasion game experiment. The targets are circled in yellow.

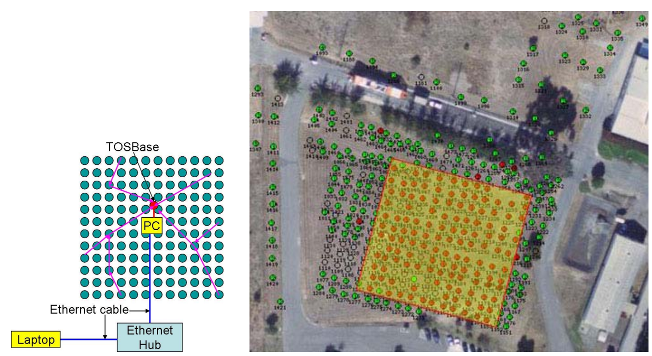

Fig. 2. (left) 144-node sensor network deployment setup for experiment. (right) Sensor network deployment on a map.

The gateway to the sensor network was a mote connected to a personal computer, marked by *TOSBase* in Figure 2 (left). For the purposes of displaying the application to an audience sitting outside the sensor field, the personal computer routed the data packets back to a laptop near the audience via Ethernet. The laptop then timestamped the returning packet, ran the tracking and coordination algorithms, and displayed the results to a screen. We used simulated pursuers in the experiment because it was difficult to navigate a ground robot in the field of tripods.

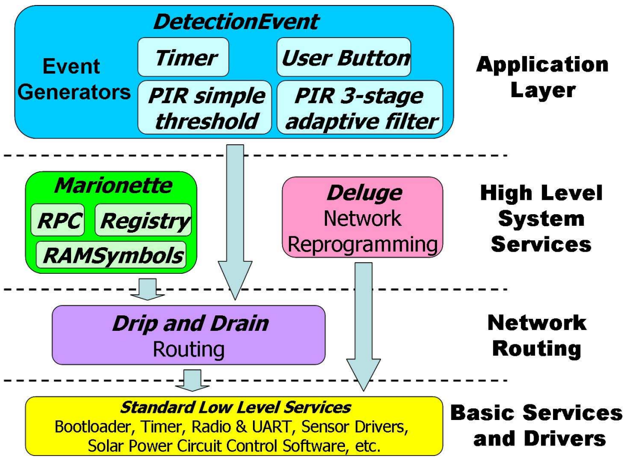

Fig. 3. Software services on the sensor network platform. The core network management services are *Deluge* for network reprogramming [7] and *Marionette* for fast reconfiguration of parameters on the nodes [8]. The *DetectionEvent* application relies on the *Drip* and *Drain* routing layer for dissemination of commands and collection of data [9]. For more details on the software architecture used in the outdoor testbed, see [3], [8].

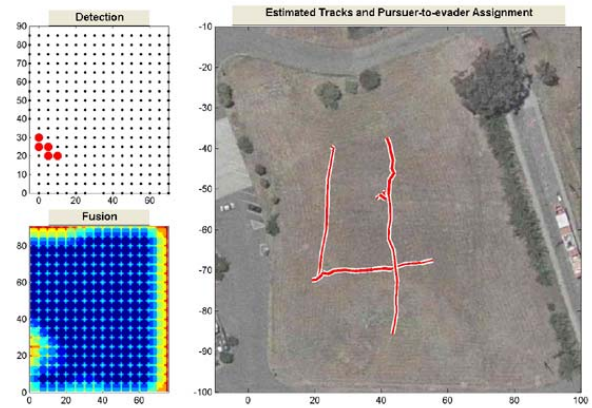

Fig. 4. Estimated tracks from an experiment with three people walking in the field. (upper left) Detection panel. Sensors are marked by small dots and detections are shown in large disks. (lower left) Fusion panel shows the fused likelihood (see [1]). (right) Estimated Tracks and Pursuer-to-evader Assignment panel shows the tracks estimated by MCMCDA, estimated evader positions (stars) and pursuer positions (squares). (This figure is best viewed in color.)

II. Platform

A. Hardware

A new sensor network hardware platform called the *Trio* mote was designed by Dutta *et al.* [3] for the outdoor testbed. The Trio mote, pictured in Figure 1 (left) is a combination of the designs of the *Telos B* mote [4], *eXtreme Scaling Mote* (XSM) sensor board [5], and *Prometheus* solar charging board [6], with improvements.

The Telos B mote is the latest in a line of wireless sensor network platforms developed by UC Berkeley for the NEST project. It features an 8MHz Texas Instruments MSP430 microcontroller with 10kB of RAM and 48kB of Program Flash and a 250kbps, 2.4GHz, IEEE 802.15.4 standard compliant, Chipcon CC2420 Radio. The Telos B mote provides lower power operation than previous motes (5.1 μA sleep, 19 mA on) and a radio range of up to 125 meters, making it the ideal platform for large-scale, long-term deployments.

The Trio sensor board includes a microphone, a piezoelectric buzzer, x-y axis magnetometers, and four passive infrared (PIR) motion sensors. For the multi-target tracking application, we found that the PIR sensors were the most effective at sensing human subjects moving through the sensor field. The magnetometer sensor had limited range even detecting targets with rare earth magnets and the acoustic sensor required complex signal processing to pick out the various acoustic signatures of a target from background noise. The PIR sensors provided an effective range of approximately 8 meters, with sensitivity varying depending on weather conditions and time of day. The variability in the signal strength of the PIR sensor reading prohibited easy extraction of ranging information from the sensor, so we used the PIR sensors as binary detectors.

B. Software

The software running on the sensor nodes are written in NesC [10] and run on TinyOS [11], an event-driven operating system developed for wireless embedded sensor platforms. The core sensor node application is the *DetectionEvent* module, a multi-moded event generator for target detection and testing node availability. The sensor node application relies on a composition of various TinyOS subsystems and services that facilitate management and interaction with the network (see Figure 3).

The *DetectionEvent* module provides four modes of event generation from the node – events generated periodically by a timer, events generated by pressing a button on the mote, events generated by the raw PIR sensor value crossing a threshold, and events generated by a three-stage filtering, adaptive threshold, and windowing detection algorithm for PIR sensor readings developed by the University of Virginia (UVa) [12]. The timer generated events aided visualizing which nodes in the network were alive after parsing and displaying the responses on the base station. The three-stage PIR detection filter code by UVa was used during the development cycle. While it had potential to be more robust to different environmental conditions, we reverted to the simple threshold PIR detector during the day of the demo because the simple threshold detector was easy to tune and performed well.

The multi-sensor fusion, MCMCDA, assignment, and pursuit algorithms are written in MATLAB and C++ and run on the base station. Data was timestamped at the base station before being used by the multi-sensor fusion algorithm.

III. Live Demonstration

The multi-target tracking algorithm was demonstrated on one, two, and three human targets, with targets entering the field at different times. In all three experiments, the tracking algorithm correctly predicted the number of targets and produced correct tracks.

Figure 4 shows the multi-target tracking results with three people walking through the field. The algorithm rejected false alarms (see Figure 5), compensated for missing detections, and dynamically corrected previous track hypotheses as it received more sensor readings. Furthermore, the algorithm correctly disambiguated crossing targets in the two and three target experiments without classification labels on the targets, using the dynamic models and target trajectories before crossing to compute the tracks.

In the last demonstration, two simulated pursuers were dispatched to chase two crossing human targets. The pursuer-

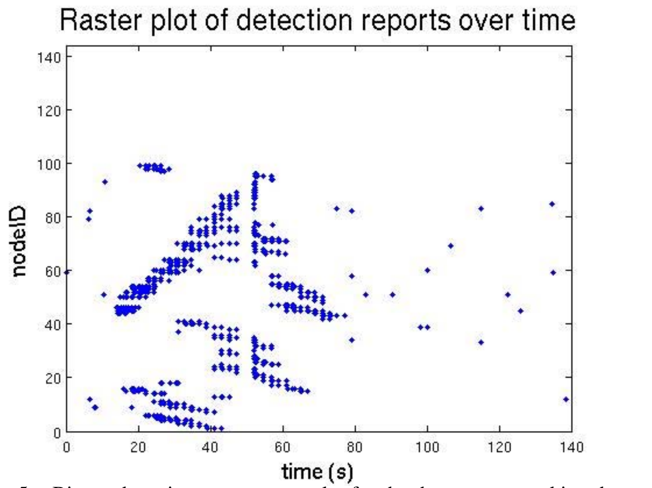

Fig. 5. Binary detection report raster plot for the three-target tracking demo. The targets entered and left the field between time 10 and time 80. Notice the abundance of false alarms beyond those times. (This figure is best viewed in color.)

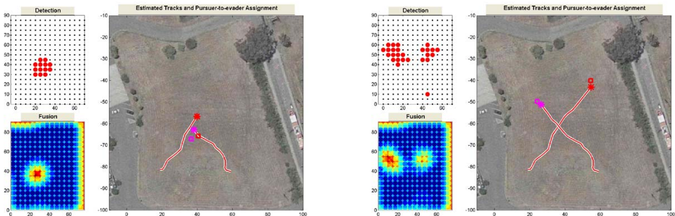

Fig. 6. Estimated tracks of evaders and pursuer positions (left) before and (right) after crossing from the pursuit evasion game experiment. (This figure is best viewed in color.)

to-target assignment and the robust minimum time-to-capture control law were computed in real-time, in tandem with the real-time tracking of the targets. The simulated pursuers captured the human targets, as shown in Figure 6. A live picture of this experiment is shown on the right of Figure 1.

IV. Discussion

We demonstrated a large-scale, wireless, robust surveillance and control system operating on a sensor network. Further experiments need to be done to carefully measure how much noise and data loss such a control system can tolerate.

Some preliminary results on the behavior of the sensor network when tracking multiple targets can be gleaned from Figure 5. The targets trigger up to approximately 10 reports per node because the sinusoidal signal from the PIR sensors trigger the detection threshold multiple times and the targets were traveling at low speeds (closely spaced reports from a sensor were binned together as one report and filtered by the tracking algorithm, and hence were not a problem). The spike in traffic shortly after time 50 was approximately when two of the targets crossed. More careful experiments need to be run to determine whether this was a result of network congestion and routing delay or whether the base station laptop was overloaded and timestamped the data incorrectly. Table I gives a sense of the amount of network traffic generated during the multi-target tracking experiments. Throughout the demonstrations, the routing tree formed by the Drip and Drain routing layers had a depth of less than 3 hops and gave complete communication coverage over the surveillance region.

Experiment	Duration (sec)	Number of Reporting Nodes	Number of Reports
1 target	53.5	44	293
2 target	138.5	99	992
3 target	116.4	87	1001
2 target pursuit	136.5	91	819

TABLE I

MULTI-TARGET TRACKING TRAFFIC STATISTICS FOR THE NEST FINAL EXPERIMENT DEMONSTRATIONS.

It is also worth noting that of the 144 nodes used in the experiment, 6 were not functioning on the day of the demo. This small node failure rate is expected for a large-scale, outdoor, distributed system. Nevertheless, the tracking algorithm was robust to these gaps in sensing coverage.

Acknowledgment

The authors would like to thank Hoam Chung, Eric Fraser, Xiaofan Jiang, August Joki, Philip Levis, Joseph Polastre, Travis Pynn, Peter Ray, David Shim, and Robert Szewczyk for their contributions.

References

[1] S. Oh, P. Chen, M. Manzo, and S. Sastry, "Instrumenting wireless sensor networks for real-time surveillance," in *Proc. of the International Conference on Robotics and Automation*, May 2006.

[2] L. Schenato, S. Oh, P. Bose, and S. Sastry, "Swarm coordination for pursuit evasion games using sensor networks," in *Proc. of the International Conference on Robotics and Automation*, 2005.

[3] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: Enabling sustainable and scalable outdoor wireless sensor network deployments," in *Proc. of the Fifth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, 2006.

[4] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. of the Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, April 2005.

[5] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proc. of the Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, April 2005.

[6] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proc. of the Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, April 2005.

[7] J. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems*, 2004.

[8] K. Whitehouse, G. Tolle, J. Taneja, C. Sharp, S. Kim, J. Jeong, J. Hui, P. Dutta, and D. Culler, "Marionette: Providing an interactive environment for wireless debugging and development," in *Proc. of the Fifth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, 2006.

[9] G. Tolle, "A network management system for wireless sensor networks," Master's thesis, Univ. of California, Berkeley, 2005.

[10] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesc language: a holistic approach to networked embedded systems," in *ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI'03)*, vol. 38, USA, June 9-11 2003, pp. 1–11.

[11] "TinyOS," http://www.tinyos.net/.

[12] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, T. He, A. Tirumala, Q. Cao, J. Stankovic, T. Abdelzaher, and B. Krogh, "Lightweight detection and classification for wireless sensor networks in realistic environments," in *SenSys*, November 2005.