Real-Time Target Tracking with CPA Algorithm in Wireless Sensor Networks

Qing Yang, Alvin Lim, Kenan Casey and Raghu-kisore Neelisetti
Computer Science and Software Engineering, Auburn University
Email: {yangqin, limalvi, caseykl, neelira}@auburn.edu

Abstract—The original CPA (closest point of approach) algorithm can localize and track moving targets within a wireless sensor network that has a specific node configuration with respect to the target trajectory. As a target moves through a large network of randomly deployed sensors, the configuration of the nodes triggered along the target trajectory may not meet this requirement and will not localize and track the target correctly. To address this problem, we propose a modified CPA (ECPA) algorithm that can accurately compute the bearing of the target trajectory, the relative position between the sensors and the trajectory, and the velocity of the target. To validate ECPA, we designed and implemented the algorithm over a data-centric sensor network. This ECPA software also communicates over a collaborative mixed wireless sensor network with control software for controlling video sensor nodes that capture real-time images or video of the target at its predicted location. Our experimental results show that we can achieve our goals of detecting the target and predicting its location, velocity and direction of travel with reasonable accuracy. In addition, results from the target detection algorithm can be used to predict the future target location so that a camera can capture video of the moving target for identification purposes.

I. INTRODUCTION

With the development of technology in wireless communication and micro-electro-mechanical systems (MEMS), it has become feasible to deploy large scale wireless sensor networks composed of relatively inexpensive sensors capable of collecting, processing, storing and transferring information. Sensors can be widely deployed within a geographic area in order to gather physical data (e.g., sound and temperature) and to facilitate efficient and collaborative control of various natural and human generated events. One interesting application of wireless sensor networks (WSN) is target tracking in a hostile environment, where physical access is accompanied by some form of danger, e.g., battlefield. In this paper, we focus on vehicle localization and tracking in wireless sensor networks and use multi-hop transmission for communicating alerts generated by sensor nodes to special control nodes or base stations. At the base station, there will be a set of cameras to capture the image/video of detected targets for further analysis through image processing software or by human beings in order to reduce false alarms.

Localizing and tracking vehicles with wireless sensor networks is a challenging problem. First, the whole system must be energy efficient. The size and number of messages must be minimized, since network communication consumes significantly more power than the local computation. Second, due to the limited capacity of each sensor, tracking mobile target requires multiple nodes to collaboratively exchange information between each other. Third, target position, velocity and direction are required to control the camera. Since multi-hop communication from the active sensing region where target appears to the base station may incur considerable delay, the base station needs the target velocity information to predict the current target position at which the camera should point. Fourth, the localization algorithm must meet the real-time deadlines of the target tracking application. It must be as fast as (or faster than) the velocity of the vehicle.

To meet the design requirements, we designed and implemented an integrated system for target detection, tracking and image/video capture of moving targets using collaborative mixed wireless sensor nodes connected by directed diffusion [2] and 802.11. The CPA algorithm [1], [3] is adopted in our system because it can calculate target position, velocity and direction without complex signal processing and can handle targets with high velocity. In fact, based on the analysis in [1], higher speeds will generate lower localization errors. The original CPA algorithm, however, cannot be directly applied to wireless sensor networks, because specific deployment configurations of sensors are required [1], [3]. For instance, in [1] the target trajectory must intersect the convex hull of sensors. While in [3] the target trajectory must be located outside the convex hull of sensors. In a realistic system, the target will trigger a set of sensors that may not satisfy the configuration requirements. Therefore, neither of the CPA algorithms [1], [3] may be directly used. To solve this problem, we developed an enhanced CPA algorithm (ECPA) that first finds the estimated bearing of target trajectory and then computes the relative position information between sensors and the trajectory. In several field experiments, we successfully detected targets and predicted their location, velocity and direction of travel with reasonable accuracy.

The main contributions of this paper are as follows. First, we have developed the ECPA algorithm for target localization and tracking that is suitable for wireless sensor network composed of low-powered and inexpensive nodes. Second, we have designed and implemented a system that integrates and interoperates the ECPA algorithm for accurately computing target location, velocity and direction with control software for controlling camera sensor nodes that capture image or video of the target at its predicted location. The software
communicates over a data-centric sensor network software. Third, directed diffusion, 802.11 and TCP/IP were integrated together to facilitate the requirements for both ad hoc acoustic sensor networks and the control center for video capture. Fourth, we have evaluated the accuracy and effectiveness of target localization and tracking in field experiments.

II. RELATED WORK

There are many research efforts on target detection and tracking in wireless sensor networks that describe several aspects of collaborative signal processing [4], [5], [6], target tracking with camera sensors [7] and real-time application for field biologists to discover the presence of individuals [8]. Recently, a set of approaches [9], [10], [11] were proposed to solve the target localization and tracking problem with proximity binary sensors which report only 1 bit information to indicate if a target appears. Though the information transmitted in networks was reduced, the localization error was increased. As proven in [10], the achievable spatial resolution $\Delta$ in localizing a target trajectory is of the order of $1/\rho R$, where $R$ is the sensing radius and $\rho$ is the sensor density per unit area. Suppose there are 25 sensors, with sensing radius of 20m, deployed in a $100 \times 100m^2$ area, then the lower bound of localization error will be 20m.

In the area of the acoustic sensor networks, there are many solutions for target localization and tracking which can be divided into three basic categories: differential signal amplitude [12], direction of arrival (DOA) [8] and time difference of arrival (TDOA) [13], [14]. The main problem with differential signal amplitude method is that distance calculation based on received signal strength is a very error-prone procedure. Because the accuracy of the RSSI (received signal strength indication) range measurements is highly sensitive to multi-path, fading, non-line of sight scenarios, and other sources of interference, this method may result in large errors. These errors can propagate through all subsequent triangulation computations, leading to large localization error. The DOA approach requires a microphone sub-array on each sensor, which will increase not only the cost of deployment but also the signal processing overhead on sensors. TDOA methods [13], [14] make use of the relative time differences among sensors. To obtain the relative time difference, each sensor needs to first get the dominant frequency of acoustic spectrum, and then broadcast this information [13], [14]. This may involve collaborative signal processing, such as FFT, which will increase the computational overhead. On the other hand, the audio data exchanged among sensors requires more network bandwidth, a very precious resource in wireless sensor network.

We chose the CPA target tracking algorithm [1] to estimate target position, velocity and moving direction. Intuitively, the CPA time is the instance when the target was at the closest point to sensor node. Unlike TDOA algorithms, the CPA algorithm only requires the CPA time be stored and exchanged among sensors. Since the CPA time may be stored as two bytes (one for the integer and the other for the decimal), message exchange overhead is low. As previously described, the original CPA algorithm [1], [3] cannot be directly used for wireless sensor networks because of the specific sensor deployment requirements. ECPA algorithm, however, does not have such network configuration requirements and it can overcome the localization error caused by the uncertainty of position and CPA time. Therefore, to the best of our knowledge, we are the first to apply an enhanced CPA algorithm in a practical wireless sensor network, composed of acoustic sensors, to localize and track moving targets.

III. TARGET LOCALIZATION ALGORITHM

A. Original CPA Algorithm

The CPA (closest point of approach) algorithm was originally designed for localizing low-flying aircraft by means of acoustic sensors [1]. The problem is first formulated in the three dimensional case and then specialized to the case when the target and sensors are all in one plane. Similar approach was proposed in [3], which requires three non-collinear sensors deployed on the same side of target trajectory before the trajectory direction can be calculated by solving a single linear equation with the sine and cosine of the angle formed by the target trajectory and a reference axis. Then, the source speed can be computed by a single linear equation. One assumption of the CPA algorithm is that the target moves with a constant velocity (i.e. linear path) while passing through the set of nodes (either three [3] or four [1] sensors). This assumption is necessary because of the limited detection range of low-powered sensor nodes and the high speed of the target. For example, suppose there are 25 sensors uniformly distributed in a $100 \times 100m^2$ area, the sensing range is 20m, and target speed is 30 miles per hour. Then within 5 seconds, the target will trigger on average 6 sensors around the trajectory, which meets the requirement of the CPA algorithm. We argue that within a short time interval, e.g. 5 seconds, the target speed can be considered constant.

As shown in Fig.1, $v$ and $r$ denote target velocity and closest point of approach to the origin of the coordinate system, respectively. Then the target path is given by:

$$p(t) = r + v(t - \tau), \quad r^T v = 0 \quad (1)$$
where $T$ denotes the transpose of a certain vector and $\tau$ the instance at which the target moves through $r$. Let $r_k$ denote the position vector of node $k$ and $t_k$ the measured CPA time. We define the CPA time as the time when target reaches the closest point of approach, and measure CPA time as the instance at which the receiving amplitude is the highest at the sensor. Therefore, the difference between those two will be time of sound propagation from the closest point of approach to the sensor. Then we can define $t_k$ as:

$$|p(t') - r_k| = \min |p(t') - r_k| = c(t_k - t') \tag{2}$$

where $| \cdot |$ is the Euclidian norm and $c$ the velocity of sound. Using Equation (1) to solve the minimal $t$ and inserting the result into the Equation (2) yields:

$$r - r_k + (v^T r_k)v/v^2 = c(t_k - \tau - v^T r_k/v^2) \tag{3}$$

where $v = |v|$. The left side of Equation (3) can be rewritten as:

$$r - r_k \cdot r + (v^T r_k) v \cdot r/v^2 \tag{4}$$

In the two dimensional case, $r$ and $v$ are orthogonal, so Equation (4) can be simplified as:

$$|r - (r_k T r)/r|/|r|^2 = |r| \cdot |1 - r_k T r|/|r|^2 \tag{5}$$

Thus, Equation (3) becomes:

$$r|r_k T r/r^2 - 1| = c(t_k - \tau - v^T r_k/v^2) \tag{6}$$

For convenience, we place the origin of the coordinate system onto one of the sensors, e.g. sensor0 ($r_0 = 0$). Subtracting $r = c(t - \tau)$ from Equation (6) yields:

$$r(|r_k T r/r^2 - 1| - 1) = (c/v^2)r_k T v = c(t_k - t_0) = d_k \tag{7}$$

Let $r_k = (x_k, y_k)^T$ and let $\phi$ and $M$ be defined by:

$$r = r(\cos \phi, \sin \phi)^T, (c/v^2)v = M^{-1}(\sin \phi, \cos \phi)^T \tag{8}$$

Then, the Mach number $M$ is positive or negative depending on whether the target crosses the CPA from right to left or left to right, as observed from the origin point. If the target trajectory intersects the line $r_0 r_k$, which also means $r_k T r/r^2 = 1 \geq 0$, then Equation (7) becomes:

$$x_k \cos \phi + y_k \sin \phi + (-x_k \sin \phi + y_k \cos \phi)/M = 2r = d_k \tag{9}$$

Otherwise, while the trajectory does not intersect line $r_0 r_k$ ($r_k T r/r^2 = 1 < 0$), we have:

$$-x_k \cos \phi - y_k \sin \phi + (-x_k \sin \phi + y_k \cos \phi)/M = d_k \tag{10}$$

Now, to solve for the target motion parameters $r$, $v$ and $\tau$, three more sensors are needed. Thus, four sensors (including the one at origin of coordinate system) should be enough to solve the problem and compute the target motion parameters. However, the method in [1] will fail if the trajectory generates an even decomposition of the sensor field, in which two sensors are on one side of the trajectory and others are on the other side, because the resulting equation of $M$ will be of fourth degree and must be solved numerically.

To solve the localization problem, the trajectory must unevenly divided the four sensors into two groups: three on one side and one the other side. To ensure this situation occurs, we need to collect information from five sensors. The next section will discuss this process in detail. Now suppose the origin of the coordinate system is at $n_0$ which is the lone node, then we have the following equation:

$$x_k \cos \phi + y_k \sin \phi + (-x_k \sin \phi + y_k \cos \phi)/M - 2r = d_k \tag{11}$$

where $k = 1, 2, 3$. By subtracting any two of the above equations, we will obtain two formulas. For example:

$$(x_1 - x_3) \cos \phi + (y_1 - y_3) \sin \phi + [(x_3 - x_1) \sin \phi + (y_1 - y_3) \cos \phi)]/M = d_1 - d_3 \tag{12}$$

$$(x_2 - x_3) \cos \phi + (y_2 - y_3) \sin \phi + [(x_3 - x_2) \sin \phi + (y_2 - y_3) \cos \phi)]/M = d_2 - d_3 \tag{13}$$

Combining these two with Equation (11) when $k = 3$, yields:

$$\xi_k \cos \phi + \eta_k \sin \phi + (-\xi_k \sin \phi + \eta_k \cos \phi)/M = \delta_k \tag{14}$$

$$x_3 \cos \phi + y_3 \sin \phi + (-x_3 \sin \phi + y_3 \cos \phi)/M = d_3 \tag{15}$$

where $k = 1, 2, \xi_k = (x_k - x_3)$, $\eta_k = (y_k - y_3)$ and $\delta_k = (d_k - d_3)$. By multiplying Equation (14), relative to specific $k$ value ($k = 1, 2$), with $\xi_{3-k}$ and subtracting them, we obtain:

$$\delta_1 \xi_2 - \delta_2 \xi_1 = (-\eta_1 \xi_2 + \eta_2 \eta_1) \sin \phi + (\eta_1 \xi_2 - \eta_2 \xi_1) \cos \phi/M \tag{16}$$

Multiplying Equation (14), relative to specific $k$ value ($k = 1, 2$), with $\eta_{3-k}$ and subtracting them again gives:

$$\delta_1 \eta_2 - \delta_2 \eta_1 = (-\eta_1 \xi_2 + \eta_2 \xi_1) \cos \phi + (\eta_1 \xi_2 - \eta_2 \eta_1) \sin \phi/M \tag{17}$$

Adding the square value of Equation (16) and (17), yields:

$$1 + \frac{1}{M^2} = \frac{\delta_1 \xi_2 - \delta_2 \xi_1)^2 + (\delta_1 \eta_2 - \delta_2 \eta_1)^2}{(\xi_1 \eta_2 - \xi_2 \eta_1)^2} \tag{18}$$

Now since all the variables $\delta_1, \xi_1, \eta_1, \delta_2, \xi_2, \eta_2$ are known, then two possible $M$ can be computed. As we mentioned above, the Mach number $M$ is positive or negative depending on whether the target crosses the origin from right to left, or vice versa. Therefore, by checking the position of nodes with the earliest and latest measured CPA times, the target direction can be determined easily.

Till now, Equation (14) can be rewritten as:

$$(-\xi_k + \eta_k/M) \cos \phi + (-\eta_k - \xi_k/M) \sin \phi - \delta_k = 0 (k = 1, 2) \tag{19}$$

Given the computed $M$ and $\cos \phi = \pm \sqrt{1 - \sin^2 \phi}$, four possible value of $\phi$ will be generated by solving the previous equation. According to the original CPA algorithm, by inserting the $M$ and each $\phi$ into Equation (11), $r$ can be obtained. If $r < 0$ or the computed trajectory does not match the assumed sensor field decomposition, the value of $\phi$ will be rejected; otherwise, it is correct.
The algorithm is applicable only if there are at least three nodes in detail how ECPA solves those problems. Since the CPA localization error caused by the uncertainty of position and between line \( n_k \) get two possible results for this trajectory since we need to know how it divides the hull of sensors. Let \( n_1 \) and \( n_2 \) be the two nodes with the smallest received signal amplitude, which also means they are at the farthest position from the trajectory. Then we have:

\[
d_i = \frac{|k' \cdot x_{n_i} - y_{n_i} + b|}{\sqrt{k'^2 + 1}} (i = 1, 2)
\]

Since the distance \( d_i \) is roughly proportional to the square root of the inversed receiving signal amplitude, we can solve Equations (22) and get two possible \( b \) values which give two possible locations of the trajectory. Since the two node \( n_1 \) and \( n_2 \) can either be on the same side of trajectory or on the different sides, we have two possible network configuration as shown in Fig. 3(a) and Fig. 3(b). In this figure, the two possible trajectories \( l \) (the correct one) and \( l' \) (the wrong one) are shown as solid and dashed lines, respectively. In Fig. 3(a) where \( n_1 \) and \( n_2 \) are on the same side of the trajectory, if \( l' \) is considered as the expected trajectory, it will contradict the assumption that both \( n_1 \) and \( n_2 \) are at the farthest position. In Fig. 3(b) where \( n_1 \) and \( n_2 \) on the different sides, line \( l' \) denotes the trajectory being outside the convex hull of sensors that also contradicts the assumption that the trajectory intersects the convex hull. Eliminating the inadmissible solutions, we eventually can derive a formula for estimating target trajectory in the coordinate system. This formula may not be accurate enough to meet our target localization and tracking goals, but it is sufficient for determining the practical sensor deployment that meets the requirements of the CPA algorithm. Based on the estimated trajectory, four sensors which are divided into two groups (one is on one side and the other three are on the other side) by the trajectory will be selected. Then from Equation (18), the Mach number \( M \) can be computed and inserted into Equation (19). Because \( \cos \phi = \pm \sqrt{1 - \sin^2 \phi} \), four possible value of \( \phi \) will be generated. However, because the uncertainty of nodes position and measured CPA time will cause \( r \) to be incorrectly computed, the original CPA algorithm will fail to compute the target location. As it will be shown in Section VI, the failure rate of the original CPA algorithm is very high (about 90%). Location of nodes obtained from GPS may be affected by the ionosphere error, satellite clock error, orbit error, troposphere error and multipath error [15]. The measurement error of a C/A code receiver with either standard correlator or narrow correlator will range from 0.1 to 3 meters [15]. In addition, the measured CPA time will be affected by the environment noise and the error of time synchronization between sensors.

Through mathematical analysis, we found that the error of the computed \( \phi \) is highly dependent on the network configurations. For some node deployments, the error of the computed \( \phi \) given by CPA will be very large; in this case, the estimated \( \phi \) serves as a reference point and the computed \( \phi \) that is closest to the estimated \( \phi \) will be considered the result. Further analysis and simulations confirm that the selected \( \phi \) value is very accurate. Then the final target trajectory location is computed based on the newly computed slope and the received signal strength at nodes, as previously stated. Due to the space limit, we omit the complete analysis in this paper and refer the reader to our future publications for detailed description.
In this section, we describe the framework of our target detection, localization and tracking system and the details of the architecture, hardware information, mixed networks internetworking and target detection method.

A. System Architecture

Our target detection and tracking system is composed of six components that are inter-networked together using either IEEE 802.11 or IEEE 802.3, and directed diffusion [16]. We adopt this architecture and hardware to show that the ECPA algorithm works well in practice, although there are more efficient system implementations which are outside the scope of this paper. Each of the six components represents a role or a function which is performed by one or multiple computers. Fig. 4 illustrates the architecture of the system. The six components are stated as follow:

1) Sensor: Sensor nodes monitor for targets, detect the measured CPA time and report results to the cluster head through the directed diffusion network. Initially, sensors are in target monitoring mode where they continuously monitor for acoustic events. Once an acoustic threshold is exceeded, the sensor node switches into target detection mode. During the detection period, the CPA time is obtained as the instance when maximum amplitude of sound occurs.

2) Cluster Head: The cluster head node receives the CPA time data from sensors over the directed diffusion sensor network as the input to determine the position, direction and velocity of the target by running the ECPA algorithm. This data, along with the CPA time of the cluster head, is transmitted to the gateway node.

3) Gateway: The gateway node internetworks the diffusion network to the IP network. It forwards target motion information from the cluster head to the camera control node. The gateway represents the camera controller to the diffusion network and its primary task is to receive packets destined for the camera and convert them from diffusion packets to IP packets. This involves subscribing to target tracking data on the diffusion network and sending IP packets to the camera control node on the IP network.

4) Camera Control: The camera control node receives packets from the gateway, and then predicts target position based on the target velocity and the time delay for package transmission from the sensors to the camera control node. It then pans and zooms the camera to point at the predicted location. The camera controller resides on a wired Ethernet network and listens for packets from the gateway on port 8899. When a packet is received, the camera controller issues movement commands to the camera over the serial port.

5) Video Capture: The video capture node is responsible for interfacing with the video output of the camera. The output may be captured as a video or individual frame which may be saved as images.

6) System Control: The system control node manages the execution of the remote nodes. This includes starting, monitoring and stopping processes. We developed two methods for managing sensor networks: ISEE software (Section IV.D) and a direct method using SSH. For simplicity, we used the direct method. The system control computer uses SSH to remotely login and issue commands on the sensor, cluster, and gateway nodes. We developed scripts to handle common tasks of the sensor system (e.g. starting and stopping processes). Since any command may be given over SSH, the system control node has complete control of all the nodes in the diffusion network and even the gateway and camera control node.

B. Hardware

Each of the nodes is an x86-based laptop computer. The sensor nodes including the cluster head are laptops with 800MHz CPU and 82801CA/CAM AC’97 Audio Controllers. The gateway node, camera control, and system control are laptops with 1.2GHz CPU and 512M memory. The video capture computer is a Pentium IV based laptop (1.2GHz CPU) with Windows XP operating system. Each computer is also equipped with an IEEE 802.11b card for wireless connectivity. Video capture is performed using a Pinnacle 500 USB video converter which converts the camera’s RCA output to digital video and interfaces with the computer using USB 2.0. The camera is a Sony EVID30, and the interface to the controller is RS232C, 9600 bps, serial port.

C. Collaborative Processing with Directed Diffusion

Directed diffusion is useful for disseminating sensor data to the sink nodes because of its energy efficiency and scalability. However, in these experimental studies, we have used diffusion to support collaborative processing among distributed sensors.

Direct diffusion uses a publish/subscribe communication model whereby a sink node requests data by sending interests for a named data. As the interest is flooded through the network, each intermediate node establishes a gradient with its neighbors and enables data that match the interest to be “drawn” toward the sink. Sensor nodes with data that matches the interest will forward an “exploratory data” that is propagated by intermediate nodes through established gradients to the sink. The sink sends a reinforcement message to the node that first forwarded the new data to it.

Clusterheads broadcast interests for target CPA detection data and sensor nodes that detected the target will send back to the cluster the detection and CPA data that matches the
interest. Similarly, camera or video sensor nodes broadcast the interest for target tracking data and the cluster head that computed these data correctly sends the target location, velocity and directions information to the camera nodes.

D. ISEE Sensor Network Control

As sensor networks become larger and more complex with sophisticated collaborative processing, developing these sensor networks requires a simple execution and monitoring environment for repeatable experimentation that allows for easy transfer to in-situ real world environments. We have developed ISEE [17], an Interactive Sensor Network Execution Environment that allows for control and access to simulated, emulated, and real sensor networks. The control and access environment provides remote execution, logging, interaction, and analysis facilities independent of the implementation of the sensor network. ISEE use a simple graphical interface for controlling and accessing sensor networks as shown in Fig. 5. This framework allows for extensibility, scenario creation, and experiment repeatability. This provides visibility and repeatability to sensor network experimentation that may not be otherwise available. The combination of our framework with a distributed service framework allows for reactive and language independent user and developer interaction to a sensor networks. Such an environment is necessary as simulation has a tendency to warp reality when simulating sensor networks that are very sensitive to the environment and resource constraints. At the most, it requires fully immersive runtime and development environment. Large-scale sensor networks allow for the visualization of real world environments. Such a runtime environment will transparently support any testing environment – simulated, emulated, or real.

E. Internetworking Mixed Networks

The target tracking system is networked using several different network technologies. The sensors and cluster head communicate through directed diffusion running over an 802.11 network. The camera control and system control machines are connected with an 802.3 (Ethernet) network running IP. In order to internetwork the two networks we developed an internetworking software which executes on the gateway node, which is configured with both an 802.11 interface and an Ethernet interface. The internetworking software converts diffusion packets into IP packets and vice versa. To accomplish this task attribute-value pairs from the diffusion network must be mapped to an IP address corresponding to the machine on the IP network which is interested in the data from the diffusion network. The current application only requires one-way communication (from diffusion to IP) since the target tracking application does not need to communicate in the opposite direction. The camera software generates subscription (interest) for target detection data through the IP network which is translated by the internetworking software and flooded into the directed diffusion network. When target detection from a cluster head arrives at the internetworking software in the gateway node, it unpacks the data in a packet received from the cluster head and encapsulates it in an IP packet. This packet is addressed to the IP address of the camera control computer which has previously registered with the gateway to receive target tracking packets. The newly created IP packet is then sent to the camera controller on a pre-specified port.

F. Target Detection

A typical recorded sound (5 seconds long) of moving vehicle with sample rate 4kHz is shown in Fig. 6. The CPA time can be obtained as the instance when the highest amplitude occurs which is indicated by a vertical line in Fig. 6. Obviously, longer recording times will cause higher detection delays, so the recording time should be as short as possible.

To reduce the impact of environmental noise, we designed a simple yet effective algorithm to detect the measured CPA time. Through the \texttt{dsp} (digital signal processing) interface, we first recorded a period of data (e.g. 1 sec.) from the sound card, and then filtered out the spurious spike samples in it. As shown in Fig. 6, the amplitude of recorded sound will keep increasing until it reaches the peak (shown as a vertical line), and then the amplitude decreases. Therefore, spurious spikes caused by noise should be ignored. After filtering, the sample will be divided into a number of time slices, such as 8 slices per second (0.125 second per slice). A target will be positively detected if and only if the following rules are satisfied: (1) the average amplitude level of the sound is larger than a threshold (e.g., ±20db), (2) the zero crossing rate of recording data is above a certain level. This rule eliminates the sound of the wind, since the wind has a low frequency compared to the sound of moving vehicles.

G. Time Synchronization

Time synchronization among sensor nodes was achieved by NTP (Network Time Protocol), which synchronizes the clocks
of sensors over packet-switched, variable-latency wireless networks. NTP typically provides accuracies of less than a few milliseconds on wireless networks [18], which are accurate enough for ECPA to successfully localize the target. Time synchronization of sensors is required because of the nature of the CPA localization algorithm, but this synchronization need only be done locally among the sensors. Therefore, the correctness of absolute time is not necessary in our system. In the experiment, we set the cluster head as the local time server and all sensors fetch the reference time from it. We noticed that after synchronization, the sensors did not need to be adjusted for 3-4 hours in the experiment; therefore, the time synchronization process can be done at a relatively low frequency.

V. EXPERIMENTAL RESULTS

To evaluate the performance of our systems, we set up the target detection and tracking sensor networks at two field locations: a vacant parking lot at Auburn University, as shown in Fig. 7 and at AU’s National Center for Asphalt Technologies (NCAT) test tracks. We experimented with several wireless ad-hoc sensor network configurations: (1) a basic sensor network and camera network configuration, (2) a network with 3 additional wireless network hops from the cluster head to the camera control node and (3) a network supporting more directions (orientations) for the target to move.

Our results show that we can achieve our goals of detecting the target and predicting its location, velocity and direction of travel with reasonable accuracy. The results from the algorithm that computes the target location and velocity are shown below. Our second set of results shows that the results of target detection algorithm can be used by a camera to take pictures or video of the target for identification purposes.

A. Results of Computing Target Location and Velocity

ECPA computes the target location, velocity and direction of travel based on the CPA time reported from the five sensors. TABLE I summarizes the results of the eight runs that we conducted with two target trajectories. The results show that the target location and velocity can be accurately computed. The average position error is only 1.8 meters, the average velocity error is only 3 mph and the average trajectory error is 5 degrees. For example in the first run, the predicted target location is (3608558, 641360), only 1 m from the actual location (3608558, 641359). The computed speed is 28 mph whereas the actual speed is 30 mph, and the direction of travel is 90 degrees compared to the actual direction of 87 degrees.

Fig. 8 shows the predicted position and actual path of the vehicle for the eight runs in our experiments. Note that the computed target speed and trajectory almost exactly match the actual values. The shorter dashed lines show the results from two runs in the AC direction and two runs in the CA directions (1st-4th runs in TABLE I). These trajectories are very close to the actual one shown by the solid AC path. The longer dash lines show the results of two runs in the BD direction and two runs in the DB direction (5th-8th runs in TABLE I). These trajectories (with one exception) are very close to the actual trajectory.

Compared to the results published in [6], ECPA gives more accurate results. For example, the average localization error of ECPA is 1.78m; while the root mean square errors of Extended Kalman Filter (EKF), Lateral Inhibition (LAT) and EKF & LAT are 8.877m, 9.362m and 11.306m, respectively.

B. Video Capture of the Target

The camera first points in an initial position (in our case, west). The target then moves through the sensor field and quickly passes through the camera’s field of view. The net-
worked sensors then detect the target and calculate the target location and velocity. The predicted position of target is calculated based on the velocity and transmission delay from sensors to the camera. The camera then pans and zooms towards the predicted position to capture the target on video. The images from the video capture can be used to identify the target more accurately. Fig. 9 show the videos clips of this sequence of events where Fig. 9(a) shows the target first appearing in the sensor network field, and Fig. 9(b) shows target at the predicted position.

VI. SIMULATION AND RESULTS

We make use of the extensions made to ns-2.27 by the Naval research laboratory [19], which provides the simulation with various physical phenomena such as acoustic, seismic and chemical agents. The presence of physical phenomena in ns-2 is modeled with broadcast packets which are sent over a designated channel called the "phenom" channel. In the real world, detecting acoustic events is made more difficult by the environmental noise and sensitivity issues of the microphones. We assume that the acoustic packets experience a loss profile similar to 802.11 data packets, so the noise and packet losses are simulated by the extensions provided in [20]. In the simulation, we use the same node deployment as was used in the field experiment. As expected, the simulation results with target velocities of 30 mph match the field test very well.

A. Successful Localization Rate

Due to the error of measured CPA time, the original CPA algorithm always failed while selecting the proper $\phi$ value. In most cases, the CPA algorithm can find $r > 0$, but after inserting this $r$ into the CPA algorithm, the resulting trajectory is outside the convex hull composed of sensors. Thus, the localization process failed. As shown in Fig. 10, the successful detection rate of CPA is as low as 10%. Furthermore, when considering the error of sensors location, the situation becomes even worse. However, ECPA first chooses the $\phi$ which is closest to the estimated one, and then computes the trajectory location within the convex hull of sensors, thereby achieving a localization success rate of 100%.

B. Impact of Velocity

As stated in [1], the relative error on the target distance estimate is roughly proportional to the square root of the inverse Mach number. This means the localization error of ECPA should also decrease as target velocity increases. This property is demonstrated in Fig. 11, where the average velocity error decreased from 1.3 m/s to 0.2 m/s, average direction error decreased from 5.5 to 0.5 degree and the average location error decreased from 1.35 to 0.7 meters. Note that when the target speed is 13 m/s (30mph), the error of estimated location, velocity and direction will be 1 m, 1.7 mph and 3.5 degree, respectively. Indeed, these values are very similar to the errors experienced in the field (TABLE I).

C. Impact of Sensors Location Errors

Errors in sensor location range from less than one meter to a few meters, depending on the localization technology. To understand the effect of this error, we simulated sensors location error as a uniform distributed function upon (0, $Max$), where $Max$ increases from 0.1 to 5 meters. As shown in Fig. 12, when the location error of sensors increased, the target localization errors will also increase. Interestingly, ECPA is highly tolerant of the sensors location error because the estimated target information is quite close to the actual values even with a larger sensor location error. For example, the average error of target location increase only 0.7 meters as sensors location error increased from 0 to 5 meters. Moreover, the error of estimated velocity and direction are still within the acceptable range (0.8 m/s and 6 degrees).

VII. CONCLUSION

The original CPA algorithm fails after calculating the velocity since the uncertainty of sensors’ position makes it infeasible to obtain a positive value of $r$ (the distance from a target to the reference nodes). Our enhanced CPA scheme first estimates the trajectory and then obtains the sensors’ deployment information to improve the estimate. Through the field experiments, ECPA has been shown to be an efficient and practical localization and tracking method for moving vehicles within wireless sensor networks.

For rapid deployment in the field, we will design new compact sensor fusion nodes which based on COTS components (such as PC104), which are small, inexpensive and can execute collaborative algorithms for reliably detecting and tracking targets. In addition, we will extend the network with multiple clusters to enable us to track targets moving along non-linear paths. Finally, we will experiment with arbitrary deployment of sensors in random configurations and develop methods for automatically forming clusters in these topologies.

Our prototype target tracking system demonstrates the feasibility of detection, tracking and image/video capture of moving
targets using collaborative, wireless, acoustic sensor nodes. By using the ECPA localization algorithm, target position, velocity and direction can be accurately computed and transmitted to the video camera. In addition, we successfully integrated the directed diffusion, 802.11 and TCP/IP network protocol together to facilitate the requirements for both ad hoc acoustic sensor networks and the control center for video capture. This research is supported in part by the U.S. Army Night Vision Electronic Sensors Directorate (NVESD) under Prime Contract Number DAAB07-03-D-C213-005, Subcontract No. SUB1170933RB.

REFERENCES